Bulletin of Electrical Engineering and Informatics
Vol. 13, No. 5, October 2024, pp. 3255~3267
ISSN: 2302-9285, DOI: 10.11591/eei.v13i5.8218

Image classification of fabric defects using ResNet50 deep transfer learning in FastAI

Erwin Sitompul[1], Vincent Leonhart Setiawan[1], Hendra Jaya Tarigan[2], Mia Galina[1]
[1]Study Program of Electrical Engineering, Faculty of Engineering, President University, Cikarang, Indonesia
[2]Department of Engineering, Computer Science, and Physics, School of Science and Mathematics, Mississippi College, Mississippi, USA

Article Info

Article history:

Received Jan 22, 2024
Revised Feb 15, 2024
Accepted Feb 24, 2024

Keywords:

Deep learning
Fabric defect
FastAI
ResNet50
Transfer learning

ABSTRACT

One of the most common issues in manufacturing is the inability to persistently maintain good quality, which can lead to product defects and customer complaints. In this research, the novel implementation of deep learning for fabric defect classification in FastAI was proposed. The residual network structure of ResNet50 was trained through transfer learning to classify the data set that contained five classes of fabric images: good, burned, frayed, ripped, and stained. A novel approach to constructing the data set was undertaken by compiling randomly downloaded fabric images within the aforementioned five classes with a broad variety from the internet. The effect of the two splitting methods in dividing the data into training and validation data was investigated. Random splitting divides the data into random class proportions, while stratified splitting maintains the original class proportions. Models were tested offline with unseen data and reached a mean accuracy of 92.5% for the 2-class model and 70.3% for the 5-class model. Based on the attained accuracy and precision, no splitting method was superior to the other. The feasibility of the system's online implementation was evaluated by integrating a smartphone camera to capture and classify fabric samples, with a mean accuracy of 75.6% for the 5-class model.

Corresponding Author:

Erwin Sitompul
Study Program of Electrical Engineering, Faculty of Engineering, President University
Jababeka Education Park, Jl. Ki Hajar Dewantara, Cikarang 17550, Indonesia
Email: sitompul@president.ac.id

1. INTRODUCTION

Operation of production lines in manufacturing industries requires proper quality control as a means of evaluating, maintaining, and improving product quality based on a certain standard. A quality control process benefits from the developments in automation technology in terms of improvements in production efficiency and product reliability [1], [2]. In the long run, the utilization of humans in the entire inspection process in a large manufacturing process is prone to inconsistency and inefficiency [3], [4]. On the other hand, automatic quality control offers popular solutions to the challenges, opportunities, and crucial aspects faced by industries, making it the focus of current research [5], [6]. Automatic quality control offers stable performance and can be suited to certain quality control methods applied in specific industries.

One main challenge in automatic fabric quality control is the variety in motives and patterns. A unique pattern can easily create ambiguity and at a glance can be mistaken as a rip or a fray. In addition, defects with the same terminology, *i.e.*, burned spot or stain, may look very different from one to another in size, shape, and color. Thus, creating a deep learning model that retains artificial intelligence (AI) to perform the correct fabric defect classification with a strict quality threshold needs to accommodate a vast library of knowledge.

Some machine learning libraries such as Keras, TensorFlow, and OpenCV are capable of performing image recognition or image classification tasks. These libraries combine the use of convolutional neural networks (CNN) with various algorithms for optimization, data pre-processing, and data post-processing [7]. The research history of the library's utilization in solving industrial problems proves that the aforementioned learning libraries are able to create accurate and robust models [8]. More recent libraries such as FastAI and PyTorch offer a combination of high-level abstraction through simplified and intuitive application programming interface (API) and low-level details of data processing, data augmentation, and model architecture when more customization and control are needed [6], [9]. FastAI also provides easy access to the state-of-the-art training techniques and experimentation tools, making it suitable for research purposes [10].

The application of ordinary CNN with a novel pairwise-potential activation layer (PPAL) to classify 2 classes of micro-scale fabric defects has been proposed [11]. Liu *et al*. [12] proposed defect classification in macro-scale detection based on the following classes: frayed, scratched, and missing yarn. Each member of the data set can be classified into more than one defect type. The model itself was built by using transfer learning in a 16-layer-deep CNN with several modifications. Further micro-scale classification of eight defect types in woven fabric has been presented [5], [13]. The stages included pre-processing, background separation, morphological processing, and classification with support vector machine (SVM). Computer-vision using RPLidar, which focuses on the health service assistance [14]. Jun *et al.* [15] proposed the use of CNN in fabric defect detection with a two-step strategy. Several defect classes were applied, however only one fabric type and color were utilized throughout the experiment. Finally, Ramakrishnan *et al*. [16] presented the use of appropriate CNN with an active contour feature for the specific type of fabric defect. Fabrics such as silk, jute, diamond pattern, and flower pattern, each requires one separate model.

In this paper, a machine-learning-based automatic fabric defect classification system using a CNN with ResNet50 architecture is proposed. The main contribution of the study is twofold: i) the use of FastAI in fabric defect classification utilizing ResNet50, with one model for the whole data set with various fabric types and ii) the presentation of a new approach in constructing the data set which consists of random images with strong differences to increase the generality.

The system's area of application is a novelty in the FastAI library. The pre-trained model provided by the library allows transfer learning to obtain a more efficient training phase, because the a-priori knowledge was further transferred and improved according to the specific task at hand [3], [14]. The proposed system is aimed to yield higher efficiency and endurance in fabric defect classification through image recognition than human eyes. The performance of the system is examined in the classification of five fabric classes: good, burned, frayed, ripped, and stained.

An additional novelty of the proposed system is the construction of the data set. In this case, the majority of the data set was constructed by fabric images, which were randomly downloaded from the internet, complemented by fabric images self-taken by the authors. This approach was meant to close the technical gaps associated with the lack of application generality in classification using AI due to the use of images of which mostly have similar attributes in the data set. The original images were used without any data pre-processing and multiplication, in terms of orientation or fragmentation. Therefore, the data set used by the proposed system contained an enormous amount of data variation. Care was taken to ensure that the data used in the training phase, the validation phase, and the testing phase did not intersect with each other, whether entirely or partially.

This paper is organized as follows. First, deep learning with CNN is presented briefly. The optimization algorithm, learning rate, and performance measure are presented. Afterward, the implementation platforms and the stages of research are delivered, with an emphasis on data set preparation and model building and training. Then, the results of model implementations are presented. In an offline implementation, the system is tested in order to classify unseen fabric images. The online implementation is conducted to test the real-time operability of the system in a hardware configuration consisting of a laptop and a smartphone camera. This undertaking shows the versatility and portability of the system, which is a starting point for the system to develop further and become a turnkey solution. Discussion and conclusion sections are presented at the end of the paper.

2. METHOD

2.1. Machine learning and deep learning

Machine learning is a type of AI that focuses on building a system that uses certain algorithms to learn from a given set of data. Machine learning imitates the way how humans learn and make decisions on something, which is characterized by a learning process to obtain gradual improvement of accuracy over time. A machine learning algorithm can analyze and find patterns in a given data. The algorithm can also classify labeled or unlabeled data in a supervised learning or unsupervised learning condition [10], [17]. Provided that certain data belongs to a specific class, the ability to perform the classification emerges following the training process, where the data knowledge is embedded into the system [18].

Deep learning is a subset of machine learning, which is made of a multi-layered neural network exposed to a vast amount of data [19]. Through the learning algorithms, the network can perform elaborate tasks such as speech and image recognition. While being able to act as a universal function approximator using a single hidden layer only, a neural network with additional hidden layers and auxiliary connections among neurons will be able to conduct more complex tasks through refined accuracy and optimization [18], [20], [21].

2.2. Convolutional neural networks

A CNN is a highly connected neural network with the smallest unit in the form of a residual block, as shown in Figure 1 [22]. The residual blocks use the rectified linear unit (ReLU) activation function, as shown in the figure. The replicated blocks can be arranged in series with an arbitrary number of layers to create a deep neural network. The skip connection solves the problem of vanishing or exploding gradient, where the increased number of layers causes an increase in training and testing errors. If the performance of the overall network is affected by the difficulty of training the weights of a certain layer, then the skip connection will override the corresponding layer and continue to the next layer.

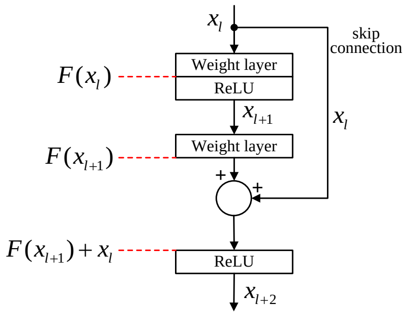


Figure 1. The architecture of CNN's residual block

The basic architecture of a CNN can be seen in Figure 2, which incorporates three processes: feature extraction, classification, and probability distribution. The input image is processed through a convolutional layer and a pooling layer [23]. The output of the pooling layer serves as the input to a fully connected neural network, which uses the ReLU activation function. The output of this network is then further processed by an activation function to determine the probability distribution of the classification.

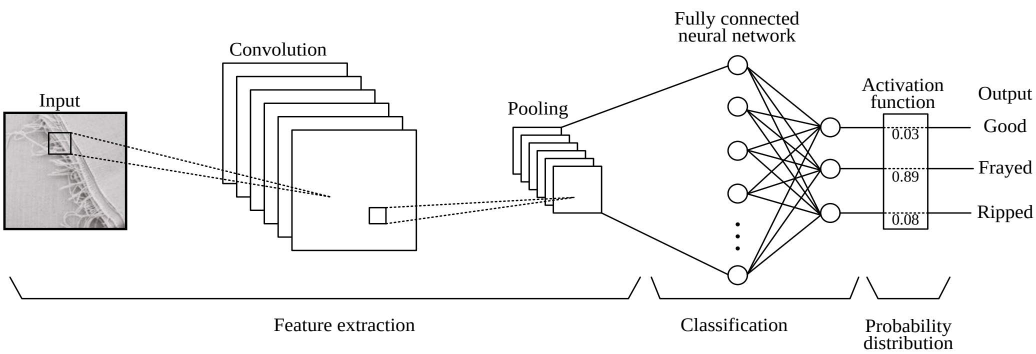


Figure 2. The basic architecture of a CNN

One type of the CNN models in FastAI is ResNet50, which is given in a pre-trained condition using the ImageNet data set [20], [24]. As the name implies, ResNet50 is a residual network that encompasses

50 layers with a sequence of one max pooling layer, 48 convolutional layers, and one average pooling layer [22]. A fraction of pre-trained parameters can be kept unchanged or frozen for some training epochs, to allow gradual adaptation of the model parameters to the new classification task [25].

In this research, the inputs to the model consisted of a batch size of 64 images per epoch. The data was processed within 3 RGB layers, where the dimensions of all images were made uniform: 150×150 pixels. Thus, the tensor size is [3,150,150]. The ResNet50 architecture took 7×7 kernels in each convolutional layer, with a stride of 2. The maximum pooling layer worked with 3×3 kernels, of which the number of strides was 2.

2.3. Optimization algorithm and learning rate

Adaptive moment estimation (Adam) [26] was chosen as the algorithm to optimize the parameters of the CNN in this research. Adam is an adaptive and efficient optimization algorithm that dynamically updates the learning rate for each parameter using low memory requirements [27]. The parameter update in Adam is similar to the gradient descent method with momentum and additional root mean square (RMS) propagation. Adam uses the exponential moving average of the gradient and the square of the gradient. The parameter update is governed by (1) to (5):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (1)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2)$$

$$\widehat{m}_t = \left(\frac{m_t}{1-\beta_1^t}\right) \quad (3)$$

$$\hat{v}_t = \left(\frac{v_t}{1-\beta_2^t}\right) \quad (4)$$

$$\theta_{t+1} = \theta_t - \eta \frac{1}{\sqrt{\hat{v}_t + \varepsilon}} \widehat{m}_t \quad (5)$$

where θ_t is the parameter vector at the t^{th} epoch and η is the learning rate. Furthermore, m_t is the dynamic average of the gradient, v_t is the dynamic average of the square of the gradient, with $\widehat{m}_t$ and $\hat{v}_t$ as their correction biases, respectively. $\beta_1, \beta_2, \beta_1^t, and\ \beta_2^t$ represent the exponential decay rates; and ε is a small constant, which is included to avoid a division by zero.

A learning rate is an important feature that if carefully chosen can lead to efficient training in a short amount of time. In this research, a cyclical learning rate (CLR) approach [28] was implemented to improve the accuracy. Here, a cycle is defined as the training fragment consisting of two steps, of which one step corresponds to an increasing learning rate and the other step represents a decreasing learning rate. Each step consists of a number of iterations. The maximum, minimum, and change of rate of the learning rate in a cycle are first set. Then, based on a heuristic decision, a policy is introduced to decrease the learning rate for several orders of magnitudes, which are less than the initial value in the corresponding cycle [29].

2.4. Performance measure

The performance of the proposed system that was used to classify the fabric conditions was evaluated by using a confusion matrix. Such a matrix contains a tabular summary of the number of correct and incorrect predictions made by a classifier, as shown in Table 1. As was mentioned before, the data set contains fabric images in five classes: good, burned, frayed, ripped, and stained. If an image of a certain class is assigned to the correct class, the classification is called true positive (TP). Otherwise, if an image is assigned to the wrong class, then the classification is called false positive (FP). If an image is predicted to be the one that belongs to a certain class while it actually does not, the classification is called a false negative (FN). Finally, if an image that does not belong to a class is classified as one that does not belong to that class, the classification is called a true negative (TN).

Table 1. The confusion matrix

Actual class	Predicted class	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Two measures of classification performance were chosen: accuracy and precision. Accuracy compares the number of true classifications attempts to all classification attempts. Precision measures the capability of the model to identify TPs among all positive classifications. The formulas to calculate the accuracy and the precision are given in (6) and (7):

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{6}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{7}$$

The precision strives to make the model more cautious while performing the classification, by not neglecting the FPs. If the distribution among the classes in the data set is uneven, a successful classification of the majority class may disguise a failed classification of the minority class. In (7) suggests that the TP of the majority class will overwhelm the FP of the minority class. In this case, further measures need to be included to correctly assess the classification performance of the system. In this research, however, the data set was arranged to be balanced, so that the accuracy and precision will be adequate to represent the performance of the model.

2.5. Implementation platforms

Two implementation platforms were used in this research: FastAI and OpenCV. FastAI is a deep learning library built on top of PyTorch, which is an open-source machine learning framework based on Python programming. FastAI was developed to simplify the process of building neural networks [24]. In addition to providing high-level components that increase productivity by cutting lines of codes usually required in building machine learning systems, FastAI also provides low-level components that can be customized and combined to create new neural network architecture to suit flexible user requirements [30].

OpenCV is a library of programming functions, which is used mainly for real-time computer vision applications that include subdomains such as image recognition, event detection, and object recognition. OpenCV was written natively in C++ but can be integrated into other languages. In this research, OpenCV was used to capture the video frames delivered by a smartphone and preprocess the frames into a format ready to be processed by the model under FastAI.

2.6. Stages of research

The stages of AI implementation in fabric defect classification consisted of data set construction, model building and training, and model implementation, in the form of offline and online tests. All stages involved the use of the FastAI platform, while the OpenCV platform was used in online testing only. Two models were built for 2-class and 5-class classifications. The 2-class model indicates if a fabric is good or stained, while the 5-class model indicates if a fabric is good, stained, ripped, frayed, or burned. Both models were included in offline tests, however, only the 5-class model would be included in online tests.

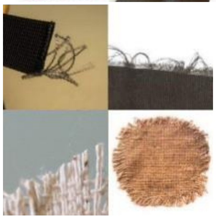
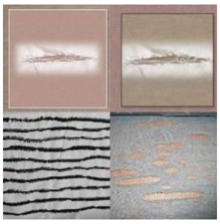
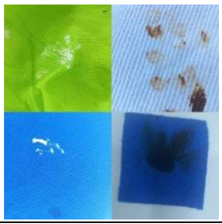
2.6.1. Data set construction

The data set was extracted from two sources: random fabric images from the internet and fabric images from the author’s self-documentation. The random fabric images consisted of a compilation of images downloaded from the internet by using bing image downloader. A sample of the data set is shown in Table 2.

The large variation of the fabrics in terms of shape, pattern, and color can be seen in Figure 2. In addition, the defects also vary in shape, size, and color. Each fabric class consisted of 170 distinct images, where 130 images were used for training and validation, while 40 images were used for testing. All images were resized to 150×150 pixels by using the padding resize technique.

The creation of the random data set is the novelty and the strength of this research. In most of the implementations of deep learning for classification tasks, the data set used for training, validation, and testing consists of images with large similarities in colors, textures, and brightness. On the contrary, the ability of the deep learning system to adapt to the great data variety and to extract information about the fabric condition was extremely put to the test in this research. In addition, the data set was split into training, validation, and testing data sets, which were mutually exclusive [31]. The data set was reasoned to contain adequate variation and unbiased information about fabric conditions. On the other hand, since the data set in this research was constructed without any reserve in variation, a moderate performance in terms of accuracy and precision would be expected [5]. Nevertheless, the authors were convinced that if the model can perform in learning and classification using this data set, then the model should be able to learn and classify any less random, pre-conditioned, and pre-processed data set with significantly better performance.

Table 2. The data set sample

Class	Characteristics	Images	
		Random download	Self-documentation
Burned	Destroyed and damaged by heat or fire		
Frayed	Unraveled or worn at the edge		
Good	Free from defects		
Ripped	Badly torn, fibers pulled apart or separated		
Stained	Colored patches or dirty marks		

2.6.2. Model building and training

Model building is defined as the creation of the model in the implementation platform of FastAI with the required architecture and optimal parameters. After a model is built, the parameters of the model are to be determined through model training. The model building was initiated by importing the FastAI library and the data set by using a set of specific commands. Training and validation were conducted simultaneously to avoid overfitting.

Out of 650 images of all classes that were used for training and validation, 80% were used to construct the training data and 20% were used to form the validation data. Two methods of splitting the data into the training data and the testing data were examined: random splitting and stratified splitting. Random splitting divides the data into training and testing data in random class proportions. On the other hand, stratified splitting splits the data according to the original class proportions [32]. In general, stratified splitting is more suitable for unbalanced class data proportion. If the class data proportions are balanced, random splitting can be used to increase the generality and find better results [31].

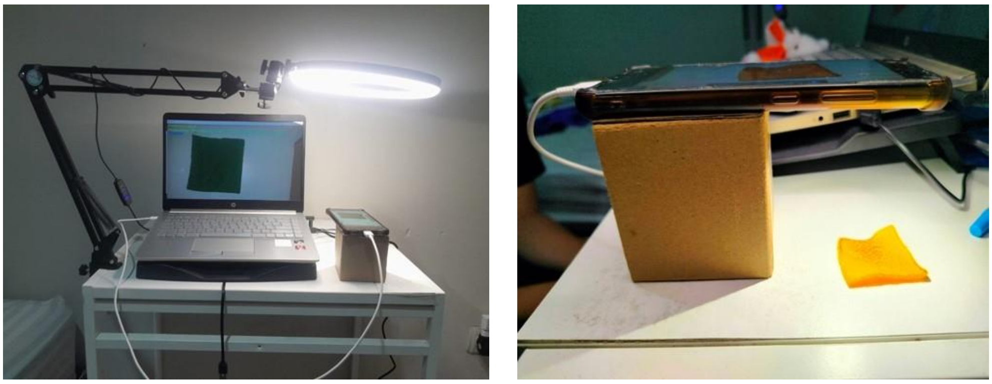
The deep learning process was initiated by an API called learners. Several features such as the optimization function and the performance metrics (such as error rate and accuracy) could be initiated and defined in this object. Early stopping callbacks such as the minimum delta and the patience were also implemented to avoid overfitting in the training process [33].

Thereafter, the pre-trained ResNet50 model was loaded as the starting point for the training process, which is also referred to as transfer learning. The pre-trained model was to be trained further, utilizing the aforementioned training and validation data. The fine-tuning process conducted by the authors provided the best learning rate training scheme of 10 unfrozen epochs followed by 5 frozen epochs. The models with the best performance were saved for further implementation in offline and online tests.

2.6.3. Model implementation

In the model implementation, the model itself undertook offline and online tests. The offline testing was conducted to evaluate the ability of the proposed system to classify the unseen data. This step was fully conducted in a FastAI.

The online test was conducted to investigate the feasibility of implementing the proposed system in a real-time automated fabric defect classification. This step was conducted in FastAI and OpenCV. The hardware setup for the online test is depicted in Figure 3. The required hardware includes a laptop, a smartphone, and additional lighting, as shown in Figure 3(a). Fabric samples were positioned in front of a smartphone camera within the autofocus range of the camera, as shown in Figure 3(b).



(a) (b)

Figure 3. The hardware setup of the online test: (a) front view and (b) side view

The ResNet50 model with the best parameters was used to classify images captured by a Samsung A32 camera. The built-in camera of the smartphone comprises a lens with an equivalent focal length of 25 cm and a 64 MP S5KGW3 sensor. The video mode operated in a 1080p 24 fps setting. The smartphone camera is superior compared to a webcam because it has a higher resolution and built-in auto-focus feature. In the online test, the video image was taken for 10 seconds, amassing 240 frames for each fabric sample to be classified by the model. The classification results were averaged and the highest class percentage was delivered as the sample's final classification result.

An IP Webcam application was used to turn the smartphone into a local host with a unique IP address and make its camera accessible from any device with internet access. A laptop was used to combine the operation of the OpenCV in capturing the video and converting it into frames and the FastAI in executing the classification using the proposed ResNet50 model. The lighting incorporated a 9 W lamp, with a 26 cm beam diameter and 6500 K color temperature.

3. RESULTS AND DISCUSSION

3.1. Result of model building and training

The deep neural network ResNet50 was applied based on its original architecture, with a base learning rate of 0.001 and a minimum delta of 0.01. The patience, which is the number of epochs the model will be further trained even if the training results were not getting better, was set to 3. Minimum delta is the minimum decrease of error rate that must be fulfilled before the training is continued to the next epoch.

The best 2-class and 5-class models with the lowest error rate and highest accuracy are presented in Table 3. The number of images used in the training and validation process for the 2-class model was 260. In the case of the 5-class model, 650 images were utilized the splitting type was varied in each model, between random splitting and stratified splitting. Error rate and accuracy are complementary to attain a value of 1 (equivalent to 100%).

Table 3. The best results of the model training

Model type	Number of images in training and validation data set	Splitting type	Error rate	Accuracy
2-class	260	Random	0.0576	0.9423
2-class	260	Stratified	0.0384	0.9615
5-class	650	Random	0.3077	0.6923
5-class	650	Stratified	0.2462	0.7538

3.2. Results of model test

The results of the model training were further tested by using the unseen testing data set. The summary of the model testing arrangement is presented in Table 4. The offline test was conducted on the 2-class and the 5-class models, each with a random and stratified splitting type. Forty images (20 images from each class) were used as the test instrument for the 2-class model. The 5-class model was tested using 200 images (40 images from each class). All images were unseen during the training and the validation. An online test was conducted on the 5-class model only. Forty-five real fabric samples (9 samples from each class) were used to test the model.

Table 4. The model test arrangement

Testing type	Model type	Splitting type	Test instrument
Offline	2-class	Random Stratified	40 images
Offline	5-class	Random Stratified	200 images
Online	5-class	Random Stratified	45 samples

The summary of the offline test results for the 2-class model is shown in Table 5 (random splitting) and Table 6 (stratified splitting). For each test image, the actual class was compared with the predicted class and the result was entered into the corresponding table cell. The accuracy was calculated across all classes, whereas the precision was calculated for each class, and then the precision values were averaged out.

Table 5. Random splitting offline test result for the 2-class model

Actual class	Predicted class		Precision (%)
	Good	Stained	
Good	20	0	100
Stained	0	20	100
Average precision			100
Accuracy			100

Table 6. Stratified splitting offline test result for the 2-class model

Actual class	Predicted class		Precision (%)
	Good	Stained	
Good	19	1	79.2
Stained	5	15	93.8
Average precision			86.5
Accuracy			85.0

The summary of the offline test results for the 5-class model is shown in Table 7 (random splitting) and Table 8 (stratified splitting). For the stratified splitting, the precision values among the classes varied significantly, where good class attained 49.3% (the lowest) and stained class attained 100% (the highest). A low precision associated with the good class indicated a large number of FP classifications. As shown in good column in Table 8, 67 images were classified as good although the true positive was only 40, among which only 33 were correctly classified. The high precision of stained class suggests an excellent result, where 30 stained images were correctly classified with no FP classifications. However, the remaining 10 stained images were falsely classified into the other classes, as can be seen in the last row (denoted stained) in Table 8. This resulted in the lower precision for the classes denoted burned, good, and ripped.

In the online test, the 5-class model was tested to classify 45 fabric samples, 9 samples from each class. Figure 4 shows the screenshots of the webcam interface as the proposed system was tested with the lighting setup as described in the previous section. Figure 4(a) depicts a successful classification of a stained orange fabric sample. The fabric with the same material (flannel) with a different color (green) and a different defect (frayed) was also successfully classified, as shown in Figure 4(b). The five class probabilities were shown on the screen and the prediction was taken based on the highest acquired probability. The class order from left to right is burned, frayed, good, ripped, and stained. Figure 5 shows the screen of the laptop when the system was verified to successfully classify unseen fabric samples under indirect daylight, which is different from the standard lighting setup previously described. Figure 5(a) shows how an unseen patterned handkerchief in good condition could be classified accurately by the proposed model. Another test with an unseen sample is shown in Figure 5(b), where the model was able to classify a burned medical mask fabric. The proposed system was proven to be operable in a different condition provided that the lighting is adequate.

Table 7. Random splitting offline test results for the 5-class model

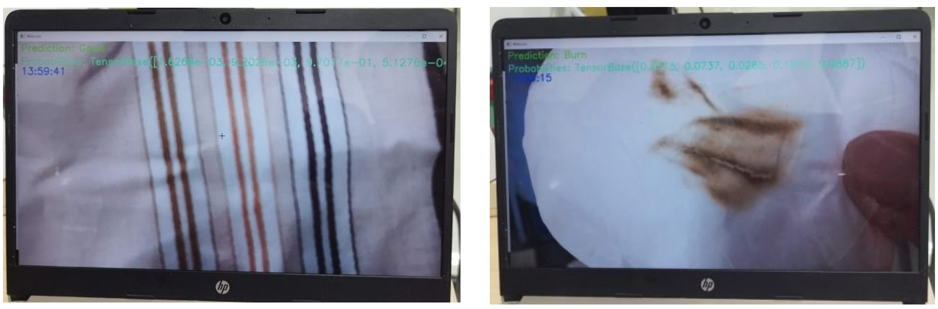
Actual class	Predicted class					Precision
	Burned	Frayed	Good	Ripped	Stained	(%)
Burned	28	3	3	5	1	63.6
Frayed	2	32	2	2	2	84.2
Good	2	1	29	8	0	67.4
Ripped	7	2	7	24	0	55.8
Stained	5	0	2	4	29	90.6
Average precision						72.3
Accuracy						71.0

Table 8. Stratified splitting offline test results for the 5-class model

Actual class	Predicted class					Precision
	Burned	Frayed	Good	Ripped	Stained	(%)
Burned	23	1	16	0	0	62.2
Frayed	2	28	6	4	0	90.3
Good	3	0	33	4	0	49.3
Ripped	3	2	10	25	0	71.4
Stained	6	0	2	2	30	100.0
Average precision						74.6
Accuracy						69.5

(a) (b)

Figure 4. Online test for the 5-class model with a lighting setup: (a) stained orange fabric and (b) frayed green fabric



(a) (b)

Figure 5. Online test for the 5-class model under indirect exposure to daylight: (a) good patterned fabric and (b) burned white fabric

The summary for the online test testing results for the 5-class model is shown in Tables 9 and 10. Each class had 9 distinct fabric samples. As shown in both tables, only burned class in random splitting achieved 100% precision with all 9 samples correctly classified. The respective burned and frayed classes associated with the stratified splitting achieved 100% precision but only with 7 and 6 correct classifications.

Table 9. Random splitting online test results for the 5-class model

Actual class	Predicted class					Precision (%)
	Burned	Frayed	Good	Ripped	Stained	
Burned	9	0	0	0	0	100.0
Frayed	0	7	1	0	1	77.7
Good	0	1	6	1	1	75.0
Ripped	0	1	1	6	1	66.7
Stained	0	0	0	2	7	70.0
Average precision						77.9
Accuracy						77.8

Table 10. Startified splitting online test results of the 5-class model

Actual class	Predicted class					Precision (%)
	Burned	Frayed	Good	Ripped	Stained	
Burned	7	0	2	0	0	100.0
Frayed	0	6	3	0	0	100.0
Good	0	0	7	2	0	53.9
Ripped	0	0	1	7	1	58.3
Stained	0	0	0	3	6	85.7
Average precision						79.6
Accuracy						73.3

3.3. Discussion

The use of random splitting delivered better results in the 2-class model offline test, with an accuracy and average precision of 100%. On the other hand, different splitting methods produced mixed results for the offline 5-class model, where the model with random splitting achieved a slightly better accuracy (71.0%) while the model with stratified splitting reached a slightly better average precision (74.6%). The mixed results were also obtained in the online test for the 5-class model, where random splitting delivered a better accuracy (77.8%) and stratified splitting gave a better average precision (79.6%). Based on this outcome, it can be inferred that between the two splitting methods, none was clearly better than the other. The user is urged to try both methods before making a choice.

Table 11 summarizes the mean of the results with different splitting methods. Each model has one mean value for accuracy and average precision. The offline 2-class model achieved the highest accuracy (92.5%) compared to 70.3% accuracy of the offline 5-class model. Classifying a fabric condition only between good or stained was known to be a much easier task than classifying 5 different classes. In this case, the use of a data set with a broad variety also increased the challenge in training the 5-class model.

Table 11. Summary of the testing results

Testing type	Model type	Mean of accuracy (%)	Mean of average precision (%)
Offline	2-class	92.5	93.3
Offline	5-class	70.3	73.5
Online	5-class	75.6	78.8

In the online implementation, the accuracy of the 5-class model increased to 75.6%, which could be the result of random optimization results or due to a favorable lighting condition during the testing. The latter might occur when the lighting intensity of the online test captured the images that matched those of the training data set. The online implementation was easily conducted without any technical difficulty. The acquisition of the fabric frames at a rate of 10 seconds/sample can be further optimized by finding a lower number of frames required for the classification task. Additionally, the implementation of the online system in a real fabric production line will require the setup of multiple cameras in multiple positions to capture the same spot as the fabric sheet moves along the line.

A comparison with other research results was now undertaken. The proposed 2-class model obtained 92.5% accuracy with a data set of 260 images. This accuracy is lower than the value obtained by the models proposed by other researchers with the same task of classifying a data set with 2 classes. Liu *et al*. [12] reported a 98.1% accuracy with a data set of 8,000 images, while Ouyang *et al*. [11] obtained 96.7% accuracy with a data set of 896 images. In the 5-class model, the 70.3% accuracy with 650 images in this research is to be directly compared with 96.4% accuracy with 2,246 images reported by Barman *et al*. [6] and 89.0±2.0% accuracy with undisclosed images as reported by Sedikki [34]. The proposed model in this research used one model for fabric classification with various colors, types, and defect appearances. With a much easier classification task using one fabric type with one color, Jun *et al*. [15] reported an accuracy of 97%. However, ResNet50 could also reach a very high accuracy (96%) with the same task. Ramakrishnan *et al*. [16] achieved an accuracy ranging from 79% to 87%. However, this is achieved with the cost of managing multiple models.

The low accuracy obtained by the proposed models was due to the large differences among the images in the data set. The images were neither preprocessed, augmented, nor multiplied. There are trade-offs between the intention of increasing the generalization of the model and the lower performance. One possible solution to improve the model performance while maintaining robustness is increasing the size of the data set [35]. This major finding implies that future attempts to classify a data set strong variation must involve a considerable number of image data in order to improve the model's accuracy and precision.

4. CONCLUSION

The implementation of AI in fabric defect classification using deep learning is proposed in this paper. A CNN with RestNet50 architecture was developed in FastAI for this purpose, which was the first effort ever made in such a particular deep-learning implementation platform. A novel approach in constructing the data set with a strong variation was done using randomly downloaded fabric images from the internet without relying on data preprocessing and multiplication.

The offline 2-class model classified good and stained fabric images with a mean accuracy of 92.5%. The offline 5-class model achieved the classification of good, burned, frayed, ripped, and stained fabric images with a mean accuracy of 70.3%. The online 5-class model was able to classify fabric samples with a mean accuracy of 75.6%. The online fabric defect classification system worked using a smartphone camera and a laptop. Each static sample was observed for 10 seconds before the classification was decided. The main contribution of this research was the effort to obtain a model with excellent generalization due to the new approach in data set construction. However, the model needs to be improved in terms of accuracy in order to reach a value above 90%. A future development will focus on preparing the proposed system for a real industry implementation, where the classification accuracy must be increased by expanding the training data, adjusting the CNN architecture, and utilizing a stronger parameter learning algorithm. In addition, the processing time and the hardware scheme need to be further optimized to fulfill the requirement of dynamic sample acquisition.

REFERENCES

[1] E. Hossain, M. F. Hossain and M. A. Rahaman, "A Color and Texture Based Approach for the Detection and Classification of Plant Leaf Disease Using KNN Classifier," *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Cox'sBazar, Bangladesh, 2019, pp. 1-6, doi: 10.1109/ECACE.2019.8679247.

[2] A. Chakraborty, D. Kumer and K. Deeba, "Plant Leaf Disease Recognition Using Fastai Image Classification," *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2021, pp. 1624-1630, doi: 10.1109/ICCMC51019.2021.9418042.

[3] M. Schrimpf *et al*.," Brain-Score: Which Artificial Neural Network for object recognition is most brain-like?," *bioRxiv*, 2018, doi: 10.1101/407007.

[4] X. Zhou *et al*., "Automated Visual Inspection of Glass Bottle Bottom with Saliency Detection and Template Matching," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 11, pp. 4253–4267, Nov. 2019, doi: 10.1109/TIM.2018.2886977.

[5] C. F. J. Kuo, W. R. Wang, and J. Barman, "Automated Optical Inspection for Defect Identification and Classification in Actual Woven Fabric Production Lines," *Sensors*, vol. 22, no. 19, Oct. 2022, doi: 10.3390/s22197246.

[6] J. Barman, H. C. Wu, and C. F. J. Kuo, "Development of a real-time home textile fabric defect inspection machine system for the textile industry," *Textile Research Journal*, vol. 92, no. 23–24, pp. 4778–4788, 2022, doi: 10.1177/00405175221111477.

[7] F. Çerezci *et al*., "Online Metallic Surface Defect Detection Using Deep Learning," *Emerging Materials Research*, vol. 9, no. 4, pp. 1266–1273, doi: 10.21203/rs.3.rs-41274/v1.

[8] J. Zhao, B. Zhu, M. Peng, and L. Li, "Mobile phone screen surface scratch detection based on optimized YOLOv5 model (OYm)," *IET Image Process*, vol. 17, no. 5, pp. 1364–1374, Apr. 2022, doi: 10.1049/ipr2.12718.

[9] R. Nithya, B. Santhi, R. Manikandan, M. Rahimi, and A. H. Gandomi, "Computer Vision System for Mango Fruit Defect Detection Using Deep Convolutional Neural Network," *Foods*, vol. 11, no. 21, p. 3483, 2022, doi: 10.3390/foods11213483.

[10] J. Howard and S. Gugger, "Fastai: A Layered API for Deep Learning," *Information*, vol. 11, no. 2, p. 108, Feb. 2020, doi: 10.3390/info11020108.

[11] W. Ouyang, B. Xu, J. Hou, and X. Yuan, "Fabric Defect Detection Using Activation Layer Embedded Convolutional Neural Network," *IEEE Access*, vol. 7, pp. 70130–70140, 2019, doi: 10.1109/ACCESS.2019.2913620.

[12] Z. Liu, C. Zhang, C. Li, S. Ding, Y. Dong, and Y. Huang, "Fabric defect recognition using optimized neural networks," *Journal of Engineered Fibers and Fabrics*, vol. 14, Dec. 2019, doi: 10.1177/1558925019897396.

[13] C.-H. Shih, C.-J. Lin, and C.-L. Lee, “Integrated image sensor and Deep Learning Network for fabric pilling classification,” *Sensors and Materials*, vol. 34, no. 1, p. 93, 2022, doi: 10.18494/sam3548.

[14] Z. J. Maknuny, S. F. Ramadhan, A. Turnip, and E. Sitompul, “RPLiDAR-Based Mapping in Development of a Health Service Assisting Robot in COVID-19 Pandemic,” in *Proc. International Conference on Sustainable Engineering and Creative Computing (ICSECC)*, Cikarang, Indonesia, 2022, pp. 72–72, doi: 10.33021/icsecc.v1i1.4171

[15] X. Jun *et al.*, “Fabric defect detection based on a deep convolutional neural network using a two-stage strategy,” *Textile Research Journal*, vol. 91, no. 1–2, pp. 130–142, Jun. 2020, doi:10.1177/0040517520935984

[16] K. Ramakrishnan, P. G. Jayakumar, P. Saravanan and P. Sivakumar, "A Novel Fabric Defect Detection Network in textile fabrics based on DLT," *2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF)*, Chennai, India, 2023, pp. 1-8, doi: 10.1109/ICECONF57129.2023.10083970.

[17] M. Vashisht and B. Kumar, "A Survey Paper on Object Detection Methods in Image Processing," *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, Gunupur, India, 2020, pp. 1-4, doi: 10.1109/ICCSEA49143.2020.9132871.

[18] S. Kim, H. Wimmer and J. Kim, "Analysis of Deep Learning Libraries: Keras, PyTorch, and MXnet," *2022 IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications (SERA)*, Las Vegas, NV, USA, 2022, pp. 54-62, doi: 10.1109/SERA54885.2022.9806734.

[19] L. Alzubaidi *et al.*, “Review of Deep Learning: Concepts, CNN Architectures, challenges, applications, Future Directions,” *Journal of Big Data*, vol. 8, no. 1, pp. 1–74, 2021, doi:10.1186/s40537-021-00444-8.

[20] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

[21] B. A. See and J. T. Francis, “High Classification Accuracy of Touch Locations from S1 LFPs Using CNNs and Fastai,” in *Proc. 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2022, pp. 342–345, doi: 10.1109/EMBC48229.2022.9871856.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *arXiv,* Dec. 2015, doi 10.48550/arXiv.1512.03385.

[23] I. Tabian, H. Fu, and Z. S. Khodaei, “A convolutional neural network for impact detection and characterization of complex composite structures,” *Sensors*, vol. 19, no. 22, p. 4933, 2019, doi:10.3390/s19224933.

[24] S. P. Praveen, P. N. Srinivasu, J. Shafi, M. Wozniak, and M. F. Ijaz, “ResNet-32 and FastAI for diagnoses of ductal carcinoma from 2D tissue slides,” *Scientific Reports*, vol. 12, no. 1, p. 20804, Dec. 2022, doi: 10.1038/S41598-022-25089-2.

[25] A. A. Asiri *et al.*, “Brain tumor detection and classification using fine-tuned CNN with RESNET50 and U-Net Model: A Study on TCGA-LGG and TCIA dataset for MRI applications,” *Life*, vol. 13, no. 7, p. 1449, 2023, doi:10.3390/life13071449.

[26] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv,* Jan 2017, doi: 10.48550/arXiv.1412.6980.

[27] M. Reyad, A. M. Sarhan, and M. Arafa, “A modified Adam Algorithm for deep neural network optimization,” *Neural Computing and Applications*, vol. 35, no. 23, pp. 17095–17112, 2023, doi:10.1007/s00521-023-08568-z.

[28] L. N. Smith and N. Topin, “Super-Convergence: Very Fast Training of Residual Networks Using Large Learning Rates,” *arXiv*, May 2018, doi: 10.48550/arXiv.1708.07120.

[29] A. Vulli, P. N. Srinivasu, M. S. K. Sashank, J. Shafi, J. Choi, and M. F. Ijaz, “Fine-Tuned DenseNet-169 for Breast Cancer Metastasis Prediction Using FastAI and 1-Cycle Policy,” *Sensors (Basel)*, vol. 22, no. 8, pp. 1–25, Apr. 2022, doi: 10.3390/S22082988.

[30] P. Alaguvathana *et al*., “Brain Disease Classification & Brain Tumor Estimation using CNN,” *Journal of Pharmaceutical Negative Results*, vol. 13, no. SO3, pp. 1584–1588, 2022, doi: 10.47750/pnr.2022.13.S03.244.

[31] R. Pramoditha, “Random vs Stratified Splits. Which one should you use when splitting your data?” [Online]. Available: https://medium.com/data-science-365/random-vs-stratified-splits-5d3d528d445b. (Accessed: Nov. 23, 2023).

[32] M. E. Ulum, J. Purnomo, E. Syahrul, and E. Wahyuningsih, “Implementation of machine learning using FASTAI for image classification on the Automatic Waste Sorter prototype,” *International Journal of Computer Applications*, vol. 184, no. 7, pp. 1–8, 2022, doi:10.5120/ijca2022922026.

[33] M. D. Tandjung, J. C.-M. Wu, J.-C. Wang, and Y.-H. Li, “An implementation of FASTAI tabular learner model for Parkinson’s disease identification,” in *2021 9th International Conference on Orange Technology (ICOT)*, 2021, pp. 1–5, doi: 10.1109/icot54518.2021.9680650.

[34] K. Sedikki *et al*., “Cumulative learning enables convolutional neural network representations for small mass spectrometry data classification,” *Nature Communications*, vol. 11, 5595, 2020, doi: 10.1038/s41467-020-19354-z.

[35] V. H. Phung and E. J. Rhee, “A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets,” *Applied Sciences*, vol. 9, no. 21, p. 4500, 2019, doi: 10.3390/app9214500.

BIOGRAPHIES OF AUTHORS

Erwin Sitompul is an Associate Professor at the Study Program of Electrical Engineering, Faculty of Engineering, President University, Indonesia. He has been the faculty member since 2007. He graduated from the Department of Physics Engineering, Bandung Institute of Technology, Indonesia, in 1998. Two years later he received the Master's degree in Electrical Engineering at the University of Kaiserslautern, Germany. In 2005 he obtained the doctorate in Automatic Control of the Technical University of Kaiserslautern, Germany. His research interests include AI, neural networks, fuzzy logic, robotics, and genetic algorithms. He can be contacted at email: sitompul@president.ac.id.

Vincent Leonhart Setiawan received a Bachelor degree in Electrical Engineering from Faculty of Engineering, President University, Indonesia, in 2024. His research interest includes the field of automation and control, such as machine learning, programmable logic controller (PLC), and SCADA. He can be contacted at email: vincentleonhart2@gmail.com.

Hendra Jaya Tarigan is an Assistant Professor of Electrical Engineering at Mississippi College (MC), USA. He is an IEEE member. He joined MC in August 2021. Prior to joining MC, he was teaching at President University (PU) in Cikarang, Indonesia (2019-2021), Howard Payne University (HPU), Texas, USA (2016-2019) and PU (2003-2009). He earned a B.S. degree in Engineering Physics in 1988 and M.S. degree in Physics in 1992, both at the University of Nevada, Reno (UNR), USA. Further, he earned an M.S. degree in Applied Physics in 2012, M.S. degree in Electrical Engineering in 2013 and Ph.D. degree in Electrical Engineering in 2016, all three at Texas Tech University (TTU), USA. His research interests are nanophotonics and sensors. He can be contacted at email: htarigan@mc.edu.

Mia Galina graduated from Brawijaya University with a Bachelor's degree in Electrical Engineering in 2001 and a Master's degree from the Universitas Indonesia in 2015. She is a lecturer in the Study Program of Electrical Engineering at President University and is pursuing a Doctorate in Electrical Engineering at Universitas Indonesia. Prior to joining President University, she worked for 15 years for a telecommunications company and as a consultant. Her research interests are IoT, sensors, and mobile wireless communications. She can be contacted at email: miagalina@president.ac.id.