International Journal of Fashion Design, Technology and Education

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tfdt20

Automatic defect detection for fabric printing using a deep convolutional neural network

Samit Chakraborty, Marguerite Moore & Lisa Parrillo-Chapman

INTERNATIONAL JOURNAL OF FASHION DESIGN, TECHNOLOGY AND EDUCATION
https://doi.org/10.1080/17543266.2021.1925355

Automatic defect detection for fabric printing using a deep convolutional neural network

Samit Chakraborty, Marguerite Moore and Lisa Parrillo-Chapman

Department of Textile and Apparel, Technology and Management, North Carolina State University, Raleigh, NC, USA

ABSTRACT
Defect detection is a crucial step in textile and apparel quality control. An efficient defect detection system can ensure the overall quality of the processes and products that are acceptable to consumers. Existing techniques for real-time defect detection tend to vary according to unique manufacturing processes, focal defects and computational algorithms. Although the need is high, research related to automatic printed fabric defect detection processes is not prevalent in academic literatures. This research proposes a novel methodology that demonstrates the application of convolutional neural network (CNN) to classify printing defects based on the fabric images collected from industries. The research also integrated visual geometric group (VGG), DenseNet, Inception and Xception deep learning networks to compare model performance. The results exhibit that the VGG-based models perform better compared to a simple CNN model, suggesting promise for automatic defect detection (ADD) of printed fabrics that can improve profitability in fashion supply chains.

ARTICLE HISTORY
Received 8 March 2021
Accepted 28 April 2021

KEYWORDS
Automatic defect detection; printed fabric; image classification; convolutional neural network; model performance

1. Introduction

Companies within the textile and apparel industry primarily initiate a quality inspection of raw materials and fabrics through human visual inspection, where deviations of textile characteristics from a predetermined standard are categorised as defective. Quality inspectors visually examine fabric rolls, which are loaded onto inspection machines unaided by technology (Wong & Jiang, 2018). However, dependence on manual inspection introduces challenges for effective quality control that arise from human fatigue, lack of reliability among inspectors, and the potential for inattentiveness. Therefore, it is highly desirable to develop computer vision-based technique to automate the process of fabric inspection (Eldessouki, 2018). Automatic defect detection (ADD) executed through computer vision can overcome limitations inherent in conventional human inspection systems through automatic identification of variations from a pre-defined visual standard (Essid, Laga, Samir, & Pan, 2018).

This research focused on rotary screen print fabric defect detection because textile printing is an expensive operation (Yumpu, 2015). Rotary screen printing involves the process of creating coloured patterns on fabric using dyes, thickeners, and various chemicals (Miles, 1994). Colour spots and misprint are two of the most common defects that occur during rotary screen printing (Cotton Inc., 2020; Yumpu, 2015). Colour spots appear when the colourant deposits in the wrong place on the fabric or when there is an absence of a particular print colour in a specific place on the fabric (Stitchdiary, 2017). Misprints occur when the fabric is crimped or creased, and the folded areas of the fabric resist the colourant, resulting in a broken or missing print area (Sharan, 2011).

Complex print patterns are particularly important to consider for ADD given their potential for fabric wastage. ADD using deep learning algorithms such as CNN can reduce the probability of waste and supply chain disruption in time-sensitive fashion supply chains. A good portion of existing literature focused on weaving or knitting faults (i.e. needle mark, hole, yarn breakage) rather than actual printing faults (e.g. misprints and colour spots) (Kang et al., 2015; Kuo, Hsu, Chen, & Chiu, 2012; Tiwari & Harma, 2015). Additional related research identified problems associated with design or pattern displacement (Jing & Ren, 2020; Li, Cui, & Xie, 2015) and displacement of print colour from an existing standard (Jahangir Alam, Hu, & Roy, 2021; Pan, Gao, Qian, & Zhang, 2010; Tavanai et al., 2006). Due to the pattern complexity and number of colours inherent in textile print designs, it is

CONTACT Samit Chakraborty schakr22@ncsu.edu Wilson College of Textiles, North Carolina State University, 1020 Main Campus Drive, Raleigh, NC 27606, USA
This article has been republished with minor changes. These changes do not impact the academic content of the article.

challenging to identify or detect faults. Therefore, this study developed a CNN-based image classification method to address real-time ADD for printed fabrics. Two research objectives (i.e. RO1 and RO2) are stated to pursue the overall research purpose:

RO1: To compile a database of print fabric images that represents two persistent defects (i.e. spots, misprints) as well as non-defective images based on rotary screen printed fabric to train and test a CNN-based ADD model for real-time deployment.

RO2: To develop, train and test a convolutional neural network (CNN) model for automatic (i.e. real-time) defect detection of printed fabric based on the image dataset established in RO1.

1.1. Relevance of automatic textile defect detection for the fashion industry

Improving the overall sustainability of textile products is an important objective of the contemporary fashion world driven by sustainability and profitability goals. The fashion industry is increasingly held accountable for their unsustainable working environments (Kabir, Chakraborty, Hoque, & Mathur, 2019). Additionally, consumers' increased environmental consciousness forces fabric producers to conform to sustainable practices in manufacturing (Chakraborty, 2016). Application of ADD offers a potential technological advance for reducing fabric defects. Since fabric printing is one of the critical operations involved in textile manufacturing, where waste is prevalent, (Kabir, Chakraborty, Hoque, & Mathur, 2019), the application of an efficient and effective ADD system is particularly important in reducing waste and preventing costly supply chain disruption (Chakraborty, Moore, & Chapman, 2021a). Fashion supply chains are widely recognised for their sensitivity to demand changes (i.e. fads, short product life cycles) and can therefore benefit greatly from preventing delivery delays in dynamic markets (Čiarnienė & Vienažindienė, 2014).

2. Literature review

2.1. Computer vision in automatic textile defect detection

Automatic defect detection involves real-time identification and classification of product defects based on image acquisition, image processing, feature extraction and image classification using Computer vision (CV) (Rahaman & Hossain, 2009). CV uses real-world images to train algorithms and machine learning models to predict outcomes based on the visual data (Vernon, 1991; Wen & Wong, 2018). The CV technique is based on two functions: image processing and pattern recognition (Hinton, Osindero, & Teh, 2006; Kavukcuoglu et al., 2010). Image processing is generated by detailed visual information extracted from images followed by pattern recognition which analyses this information to determine visual meaning to predict a target outcome (Hinton et al., 2006; Vernon, 1991). Feature extraction, image augmentation and image classification are the most important component of compute vision. Feature extraction is a useful process to reduce the number of resources without losing essential information (DeepAI, 2020). Image augmentation is the processing of image manipulation such as image rotation at different angles, horizontal and vertical flipping, upside down, noise removal, blurring and cropping of special areas (Eldessouki, 2018; Goodfellow, Bengio, & Courville, 2016). Image classification is the process of accepting given input images and then classifying them under given classes or labels or annotations (e.g. defective or non-defective fabrics) based on extracted features (Balaji & Lavanya, 2019; Dsouza, 2020). Section 2.2 presents different CV-based models and algorithms used in textile yarn and fabric defect detection.

2.2. State-of-the-art research on automatic textile defect detection methods

2.2.1. Automatic defect detection in textile yarn

Fabijańska and Jackowska-Strumiłło (2012) applied a mathematical morphology-based image-processing algorithm to identify yarn hairiness based on 2500 images for two types of yarn. Ghosh, Hasnat, Halder, and Das (2014) used a probabilistic neural network (PNN) model to classify yarn neps using the features obtained from 300 images. Ghaderpanah, Mokhtari, and Latifi (2015) proposed a direct tracking algorithm using 150 images to evaluate false textured yarn. The correlation between yarn orientation angle and crimp contraction was analysed to determine structural deviation from a given standard. Wang, Xu, and Gao (2018) used the Otsu thresholding algorithm to extract the features of protruding fibres from the images and subsequently segment the yarn based on the presence of hairiness using 120 sample images The yarn hairiness index metric is commonly used to detect yarn hairiness defects.

2.2.2. Automatic knit fabric defect detection

Habib and Rokonuzzaman (2010) developed an artificial neural network (ANN) model to detect knit fabric defects such as colour variation, holes, missing yarn and spots using 100 fabric images. Tabassian, Ghaderi, and Ebrahimpour (2011) also adopted a learning-based approach to classify defects in weft knitted fabric. Li, Ai, and Sun (2013) applied a wavelet transformation algorithm to identify broken yarns in warp knitted fabrics using hundreds of images. Mottalib, Habib, Rokonuzzaman, and Ahmed (2015) proposed a Bayesian classifier to detect defects in weft knitted fabric such as coloured yarn, fabric holes, missing yarn and spots using 1024 fabric images. Xia, Jiang, and Ma (2016) used Contourlet transform based spectral approach to detect defects in warp knitted fabrics such as broken yarn, width barrier and spots. Bandara et al. (2018) adopted a morphological transformation based algorithm for feature extraction of the defects and a convolutional neural network (CNN) to classify the defects. Hanbay, Talu, Özgüven, and Öztürk (2019) developed a real-time defect detection system to detect defects in weft knitted fabric including needle mark, stains, holes and press-off using 13,820 images.

2.2.3. Automatic defect detection in woven fabric

Schneider and Aach (2012) proposed a defect detection system using Gabor wavelet-based algorithmic function to classify defects in woven fabric such as double yarns, end missing, pick missing and thick and thin places in the yarn based on the 74 fabrics images collected from the industry. Schneider, Holtermann, and Merhof (2014) also developed a real-time defect detection prototype using template matching technique to classify woven fabric defects such as double yarns, yarn missing and problems with yarn thickness based on the 4054 images collected during fabric production. Tong, Wong, and Kwong (2017) applied k-means clustering along with Principal Component Analysis (PCA) on 204 images to classify defects such as double yarn, missing yarn and knotted yarn. Mei, Wang, and Wen (2018) applied a convolutional Autoencoder network to identify spot, fabric hole, fly yarn and perforated or damaged fabric surface. Guan, Zhong, Rui, Zheng, and Wu (2019) also applied convolutional neural networks to extract features of fabric defects including yarn missing, fabric hole and double yarns based on 12,000 gray scale images.

2.2.4. Automatic defect detection in printed fabric

Pan et al. (2010) led an experiment on colour displacement identification in the printed fabric using normalised cross-correlation techniques in order to differentiate non-defective and defective images. Tavanai et al. (2006) also applied a normalised cross-correlation technique to identify colour and pattern displacement in the printed fabric in order to differentiate non-defective and defective images. Kuo et al. (2012) used fuzzy logic control algorithms to detect defects in the printed fabric including oil spots, broken warp and weft and cracks based on 150 images, with only cracks belonging to printing defects. Tiwari and Harma (2015) proposed a mathematical morphology-based a defect detection system to detect defects associated with warp and weft yarn direction. Kang et al. (2015) developed defect detection model using the Gabor filtering approach for image processing and distance matching template to identify defects such as broken warp yarn, fabric holes, yarn knots and thick and thin places in the yarn based on the 130 images. Jing and Ren (2020) used template-matching functions to detect pattern displacement, oil spots and fly yarn in the printed fabric based on 160 images.

The existing literature exhibits limited research that focuses on automatic defect detection for printed fabric. Additionally, existing studies tend to use a non-defective image counterpart associated with the actual defective image to compare designs and explore the model accuracy, rather than developing a model capable of detecting an array of defects among unexamined images. Further, existing work to date has not applied deep learning-based CNN to identify print fabric defects, which is increasingly more popular and effective compared to image classification models (Anh & Giao, 2019; Gao, Zhou, Wong, & Gao, 2019; Ouyang, Xu, Hou, & Yuan, 2019). This research aims to fulfill this gap by developing a deep CNN model to identify real-world printing defects using the images collected directly from the rotary screen printing industry.

3. Research method

The process flow of this research starts with a research gap identification as discussed in the previous sections. The process steps include database development and model development followed by dataset splitting, image augmentation and hyperparameter selection. The model is then run on the developed database for image classification, which includes identification non-defective and defective fabrics with misprints and colour spots. The process flow diagram of this research is presented in Figure 1.

3.1. Data collection and processing

As a first step, a library was established that consisted of 400 images of rotary screen-printed fabric, collected from eight knit apparel production facilities in Bangladesh. The printed knit fabric was comprised of floral and dot designs. Factory contacts were specifically asked to provide images with colour spots and misprints

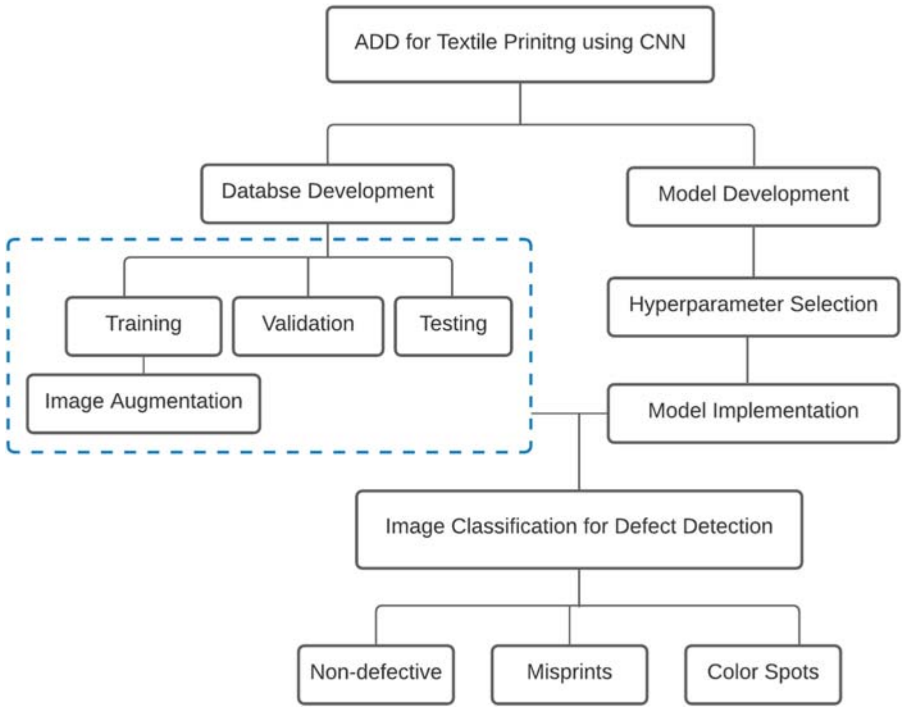


Figure 1. Process flow diagram of the proposed research.

(i.e. defective) as well as non-defective images. While the database contains some defective images with corresponding non-defective counterparts (i.e. the same print with and without defects), not all defects have a non-defective surrogate. Initially, the total dataset was divided into subsets: 50% (training, $n = 200$): 25% (validation, $n = 100$) and 25% (testing, $n = 100$). The subsets were randomly selected while controlling for the proportion of defective vs. non-defective images. The training dataset contained the largest proportion on non-defective images (i.e. 120 non-defective, 50 defective colour spots and 30 misprints), while the validation and testing datasets contained relatively larger proportions of defective images (i.e. 35 non-defective, 45 defective colour spots and 20 misprints). The proportions of non-defective images to defective images were identical for the validation and testing datasets. Image augmentation (i.e. by a factor of four) was carried out to expand the training dataset from the initial 200 images to a total of 800 images. Augmentation was implemented using three operations: the original dataset was rotated, flipped upside down and applied random noise to generate another 800 images. Image rotation is conducted by rotating images left or right to an angle between 1° and 359°. Image flipping is done by turning over images across horizontal and vertical axis. Random noise is applied to images by injecting matrix of random values generally drawn from Gaussian distribution (Shorten & Khoshgoftaar, 2019). Image augmentation was carried out using OpenCV library. Hence, image augmentation facilitated the expansion of the training dataset to using the following labels & codes: non-defective (0), colour spot (1) and misprint (2). Figure 2 shows print fabric images with (a) colour spot and (b) misprint.

3.2. Model development

3.2.1. Convolutional neural network (CNN)

Convolutional neural network (CNN) is a type of deep learning network; commonly used for feature extraction, object identification, image classification and image segmentation (Gao et al., 2019; Goodfellow et al., 2016). Figure 3 shows a CNN architecture comprising of input, output layers and multiple hidden layers comprising neurons or nodes.

Here in Figure 3, the input layer holds image data. The activation in this layer is equal to input value, as expressed in Equation (1). The hidden convolutional layers can be of different numbers starting from first hidden layer ($h^{(1)}$) to last fully connected layer ($h^{(k)}$). The first convolution layer can be expressed as Equation (2), where $h^{(1)}$ denotes first hidden layer, $f^{(1)}$ is activation function applied on input x (Gao et al., 2019; Missinglink, 2020). Moreover, $w^{(1)}$ is the weight on $h^{(1)}$ and $b^{(1)}$ is the bias applied on neuron of $h^{(1)}$ layer (Goodfellow et al., 2016; LeCun, 2020). The fully

(a)

(b)

Figure 2. Print fabric images with (a) colour spot and (b) misprint.

connected input layer ($h^{(k)}$) takes the output from adjacent previous layers and flattens them into a single vector of values representing a probability that a specific feature belongs to a class. The last fully connected layer can be expressed as Equation (3), where $h^{(k)}$ denotes the last layer or output layer that receives output information from its previous layer ($h^{(k-1)}$), $f^{(k)}$ is the activation function applied on $h^{(k)}$. The output layer displays final probabilities or class scores or arbitrary real-valued numbers for image classification. The final outcome can be expressed as Equation (4), where $\hat{y}$ represents a predicted value or probability of a class or label (Gao et al., 2019; Goodfellow et al., 2016;

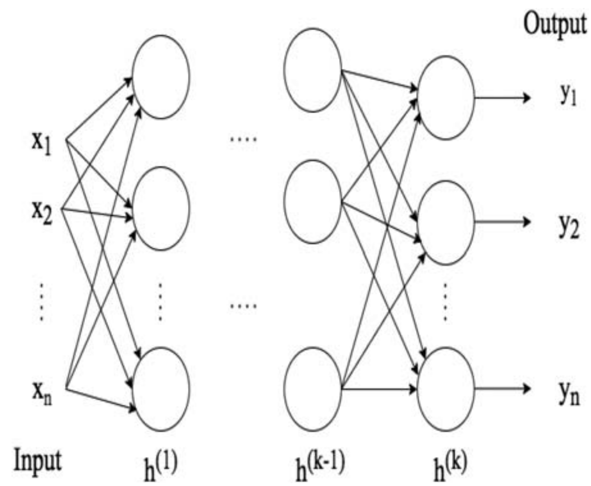


Figure 3. A general structure of fully connected convolutional neural network.

Mlnotebook, 2017; Missinglink, 2020; Stanford, 2020).

$$x_i = f_i^1 \tag{1}$$

$$h^{(1)} = f^{(1)}(w^{(1)} * x + b^{(1)}) \tag{2}$$

$$h^{(k)} = f^{(k)}(w^{(k)} * h^{(k-1)} + b^{(k)}) \tag{3}$$

$$\hat{y} = \frac{exp(h^{(k)})}{\sum exp(h^{(k)})} \tag{4}$$

3.2.2. Architecture of the proposed convolutional neural network (CNN)

To address RO2, a multi-layer convolution neural network has been employed to extract features and identify defective and non-defective fabric images. A convolutional neural network (CNN) model runs based on hyper parameters such as learning rate, batch size, hidden layers, dropout probability, regularisation and activation function. Books and scholarly articles on deep learning, image classification and automatic fabric defect detection using neural networks were studied, and empirical research was adopted to select the

Table 1. Hyperparameters and their values selected for model training.

Hyperparameters	Tested range
Learning rate	0.0001, 0.0003, 0.001, 0.003
Batch size	16, 32, 64, 128
Number of Hidden neurons	32, 64, 128, 256
Dropout probability	0.1, 0.2, 0.3, 0.4
Regularisation	0.0001, 0.0003, 0.001, 0.003
Activation function	Relu, Tanh, Sigmoid, Softplus

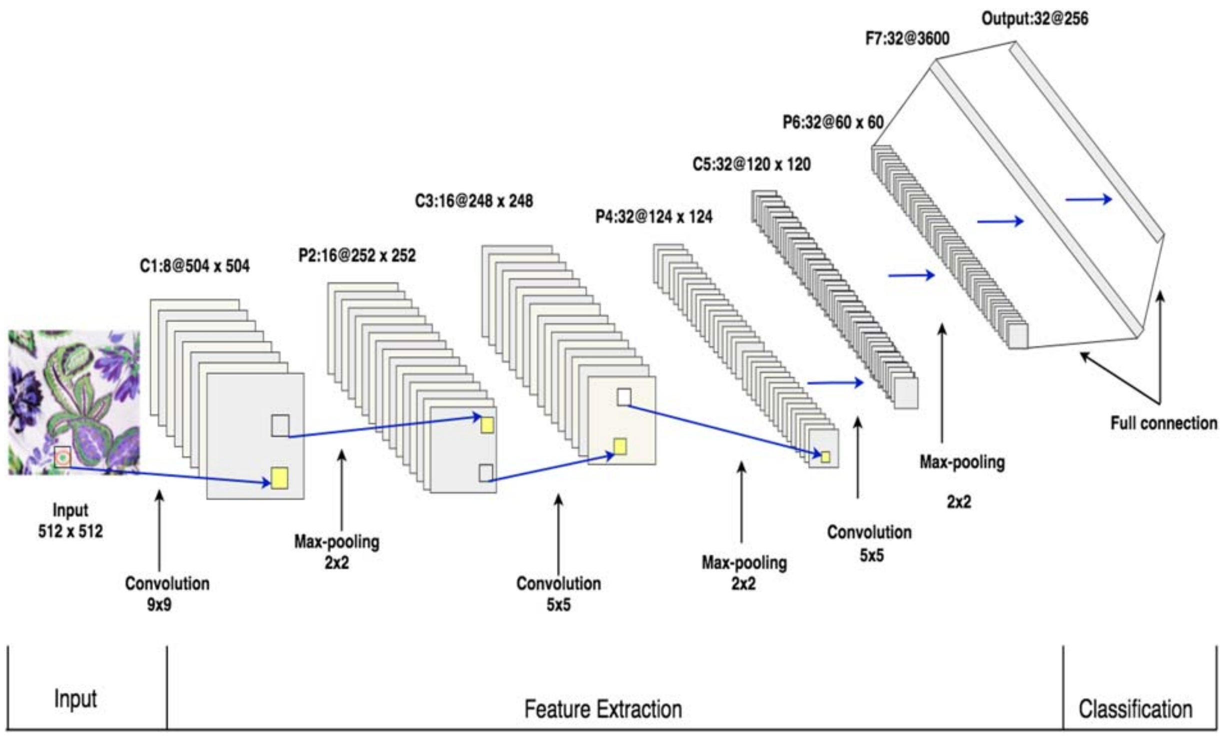


Figure 4. Proposed convolutional neural network (CNN) architecture. Each plane represents a feature map in the feature extraction stage. C and P represent convolutional layer and max-pooling layer respectively. C1: 8@504 × 504 denotes that it is the first convolutional layer comprising of eight feature maps with size of 504 × 504. The letters and numbers mentioned in the other layers also have similar connotation, whereas F7: 3600 means that this seventh layer containing 3600 neurons was connected to the previous pooling layer.

hyperparameters and their initial range of values (Albawi, Mohammed, & Al-Zawi, 2017; Basirat & Roth, 2018; Devarakonda, Naumov, & Garland, 2018; Goodfellow et al., 2016; Ide & Kurita, 2017; Luo, Wang, Shao, & Peng, 2019; Park & Kwak, 2017; Takase, Oyama, & Kurihara, 2018). Hyperparameter tuning is a computationally expensive as well as exhaustive process (Goodfellow et al., 2016; Ide & Kurita, 2017; Luo et al.,

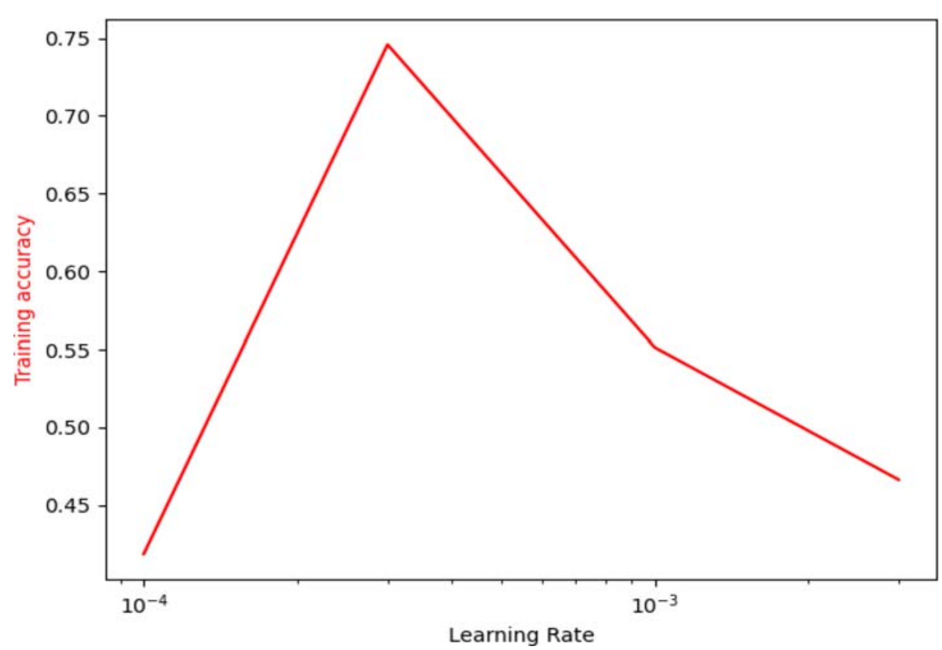


Figure 5. Training accuracy for different learning rates.

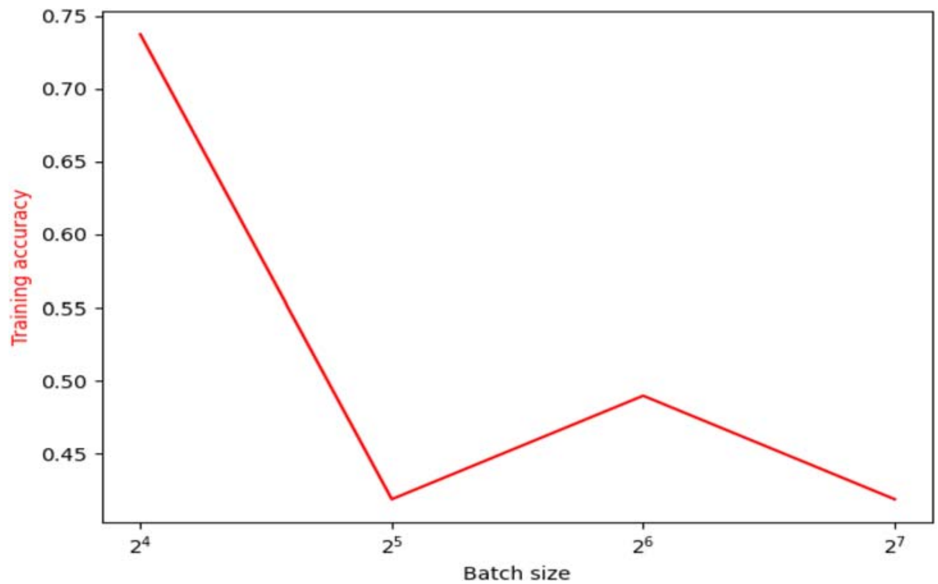


Figure 6. Training accuracy for different batch sizes.

2019). Therefore, geometric progression of hyperparameter values was chosen. Only for the dropout probability arithmetic progression was chose (Table 1), because a probability value of more than 0.5 may drop too many connections without improving the network (Goodfellow et al., 2016; Park & Kwak, 2017). A simple iteration of 20 epochs was utilised sequentially on individual parameters to search for a value that yields the highest validation accuracy instead of running an exhaustive search. Although the model performed until to 40 epochs, the performance did not improve much. Additionally, it took a long time for completing each epoch, which is computationally expensive and a slow process. Therefore, it will not be justified for real-time deployment. Hence, it was decided to keep 20 epochs to run the final model. The study employed the Adam optimiser for the loss function, which is considered as the best optimiser for image classification using CNN.

Python 3 programming language was used to develop the coding script for dataset introduction and model development (Chakraborty, Moore, & Parrillo-Chapman, 2021b). The proposed convolutional Neural Network (CNN) architecture was developed considering

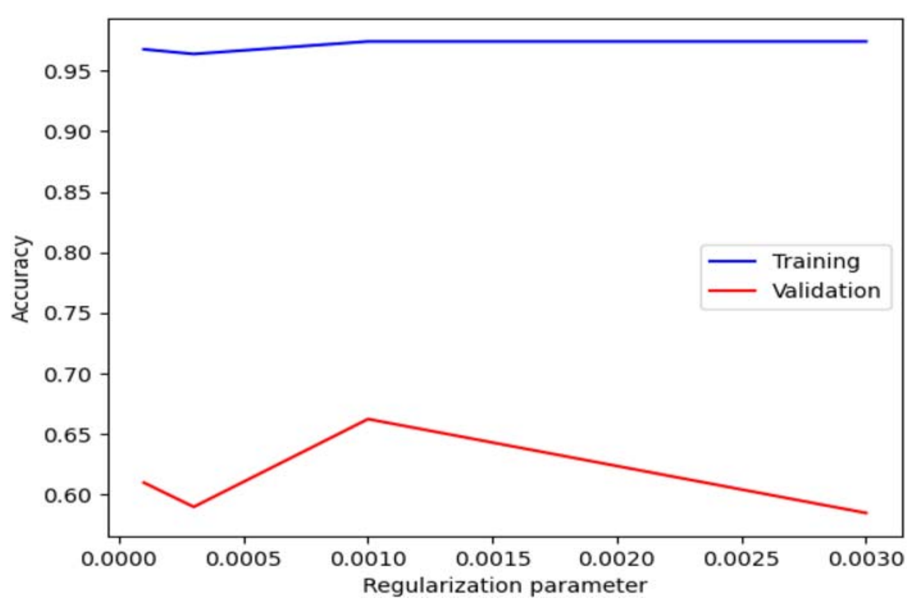


Figure 7. Training and validation accuracy for different regularisation parameters.

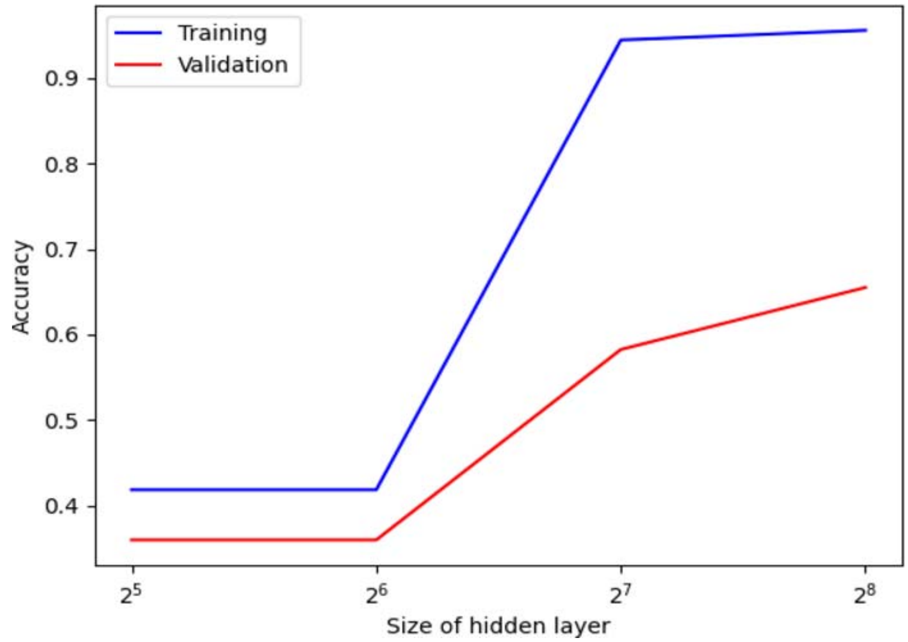


Figure 8. Training and validation accuracy for different hidden layers.

the size of the dataset and variations among image classes. Figure 4 shows the proposed CNN architecture containing a 9×9-sized filter on the first convolutional layer, and a 5×5 sized filter on second and third convolutional layers. Valid padding was used in the convolutional layers as an empirical approach. A pooling layer having 2×2 sized filters and applied with a stride of 2 was added next to the convolutional layer. The feature map extracted from 512×512 pixels size input image was reduced in this layer. The selection of these sizes was also based on previous research studies as well as the empirical approach (Abd Jelil, Zeng, Koehl, & Perwuelz, 2013; Albawi et al., 2017; Çelik, Dülger, & Topalbekiroğlu, 2014; Das, Wahi, Sundaramurthy, Thulasiram, & Keerthika, 2019; Devarakonda et al., 2018; Gao et al., 2019; Li, Grandvalet, & Davoine, 2020; Takase et al., 2018; Wan, Zeiler, Zhang, Cun, & Fergus, 2013). The softmax activation function has always been used at the output layer, as it supposed to ensure the sum of all probabilities to be 1. For

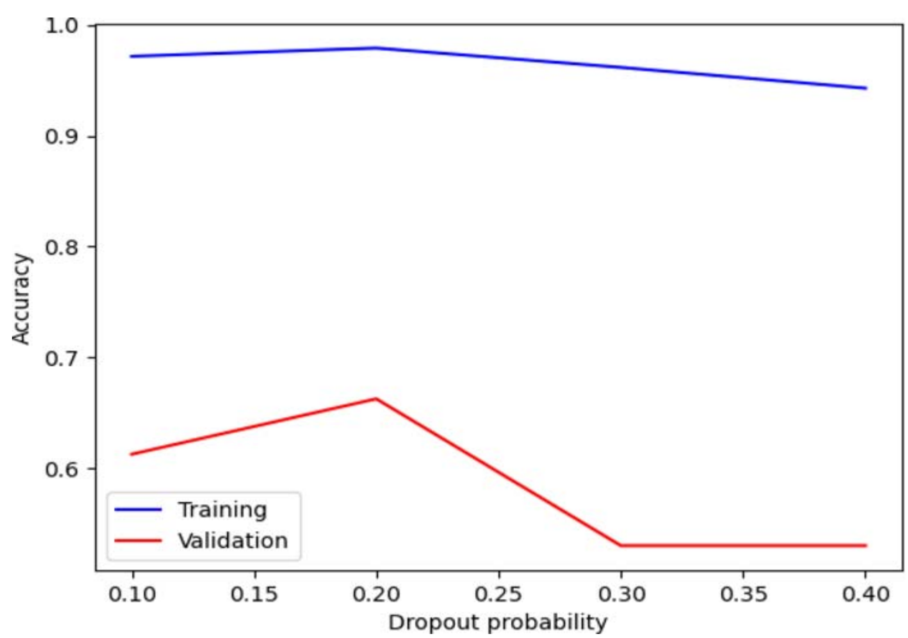
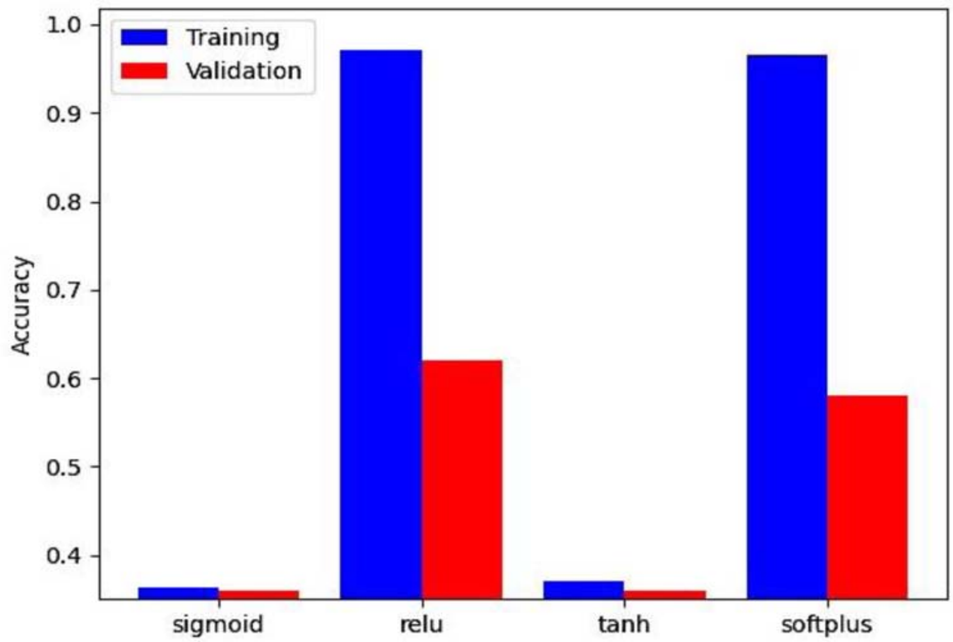
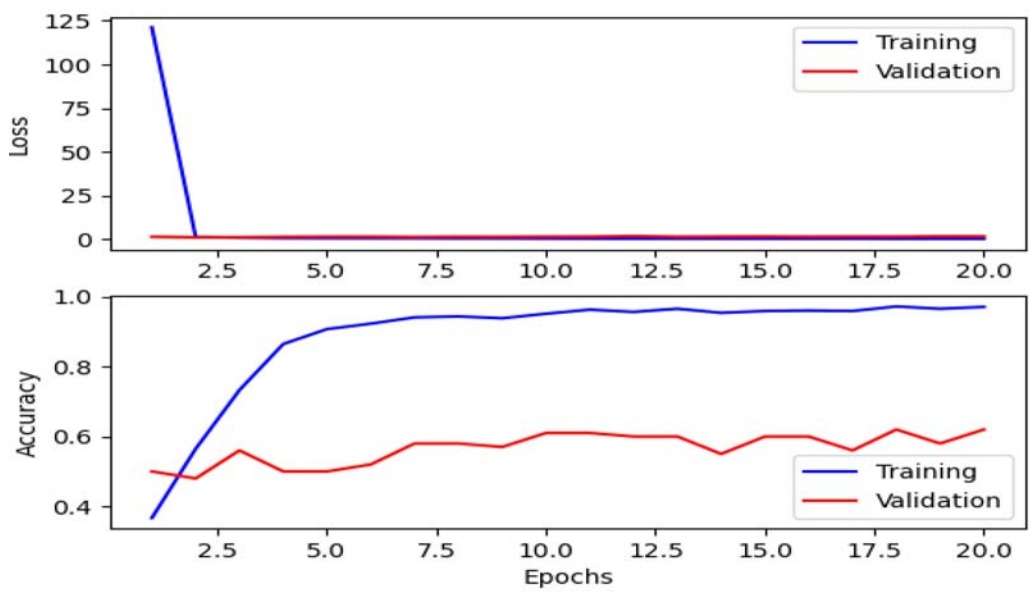


Figure 9. Training and validation accuracy for different dropout probabilities.



(a)



(b)

Figure 10. (a) Training and validation accuracy of various activation functions with the custom CNN and (b) plots of loss and accuracy vs. epochs during the training of the best model (custom CNN with ReLU).

the other layer(s), four kinds of activation functions (ReLU, Tanh, Softplus, and Sigmoid) were tried. After tuning the CNN, the same set of hyperparameters with highest accuracy and lowest loss was used to run more dense layers based neural network architectures with including VGG-16, VGG-19, DenseNet-201, Inception-V3 and Xception, which are most popular for transfer learning-based CNN networks (Simonyan & Zisserman, 2012; Zhao, Zhang, & Zhang, 2021).

3.3. Evaluation criteria

The performance of a given algorithm is commonly measured using sensitivity or recall, specificity or precision and detection accuracy measures expressed in

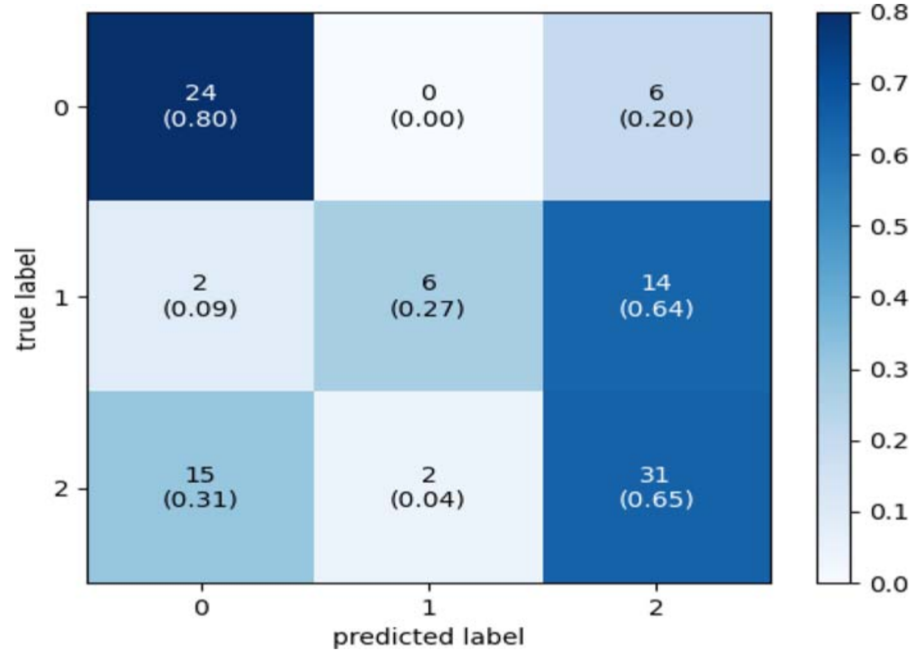


Figure 11. Confusion matrix of predictions on the testing dataset with the best model (custom CNN with ReLU).

the following Equations (6)–(8) (Ngan, Pang, & Yung, 2011).

$$\text{Recall} = \frac{\text{True Positive (TP)}}{\text{True Positive(TP)} + \text{False Negative (FN)}}$$

$$\text{Precision} = \frac{\text{True Positive (TP)}}{\text{True Positive(TP)} + \text{False Positive (FP)}}$$

$$\text{Test or Detection Accuracy} = \frac{\text{TP} + \text{FN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}}$$

4. Results and discussion

4.1. Hyperparameter selection

The hyperparameter adjustment generated displays for training and validation, losses and accuracy. These plots were used to select the final parameter values based on the highest degree of training and validation accuracy.

4.1.1. Learning rate

The Adam optimiser was applied to examine training loss using four rates (0.0001, 0.0003, 0.001, and 0.003). The model using the 0.0003 learning rate indicated the highest training accuracy (i.e. about 75%) (Figure 5).

4.1.2. Batch size

Various batch sizes (16, 32, 64, and 128) were used to record the model's performance on the training dataset. The model using the batch size 16 indicated the highest training accuracy (74%) as shown in Figure 6. The higher training accuracy helped to achieve higher training and validation accuracy in the following steps.

4.1.3. Regularisation parameter

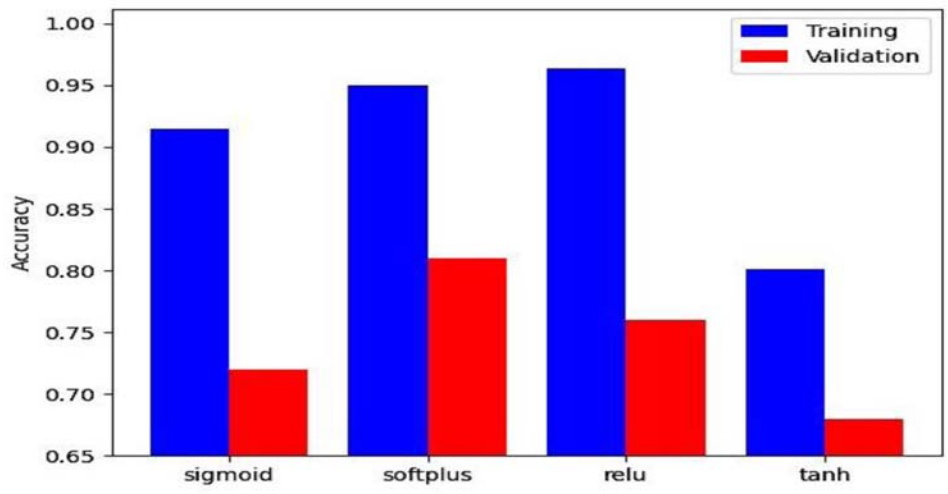
Four different regularisation parameters (0.0001, 0.0003, 0.001, and 0.003) were used to record the model's performance on the training dataset. The model with $\lambda = 0.001$ has the best performance in terms of the highest training (98%) and validation accuracy (67%) as shown in Figure 7.

4.1.4. Number of neurons in hidden layers

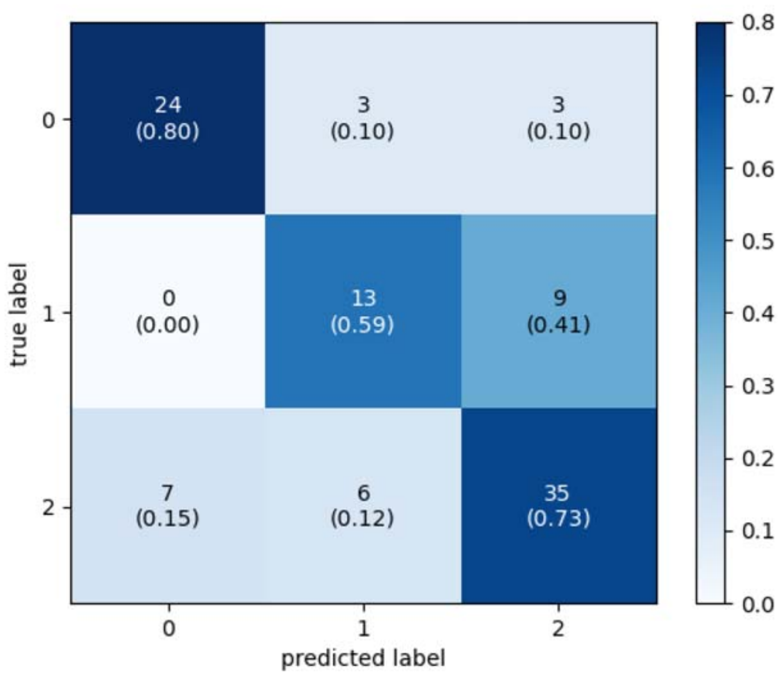
Four different values representing a number of neurons in the hidden layers (32, 64, 128, and 256) were used to record the model's performance on the training dataset. Model with 256 neurons in the hidden layer has the best performance in terms of highest training (97%) and validation accuracy (66%) as shown in Figure 8.

4.1.5. Dropout layer

Four different dropout layers (0.1, 0.2, 0.3, and 0.4) were used to record the model's performance on the training dataset. Model with a dropout probability of 0.2 has the best performance in terms of highest training (98%) and validation accuracy (67%) as shown in Figure 9.



(a)

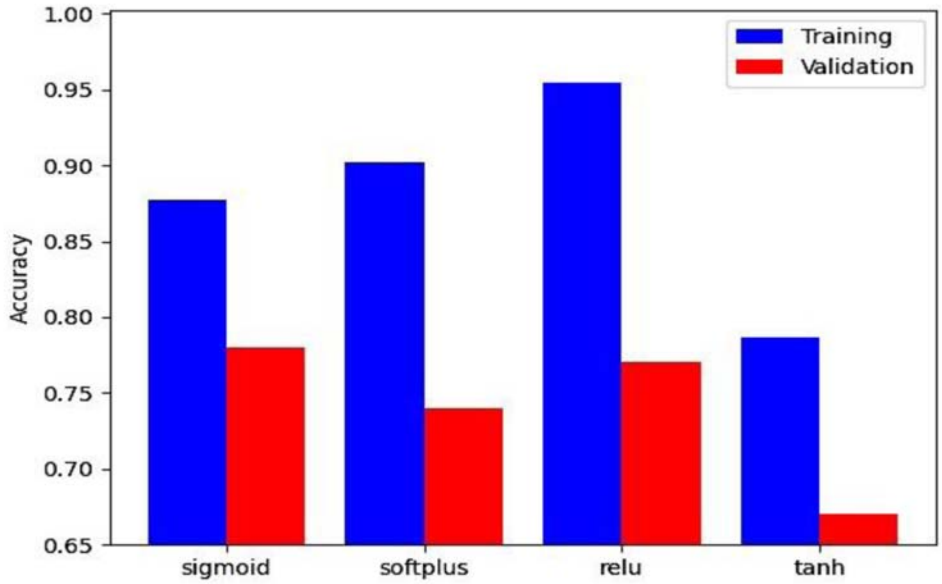


(b)

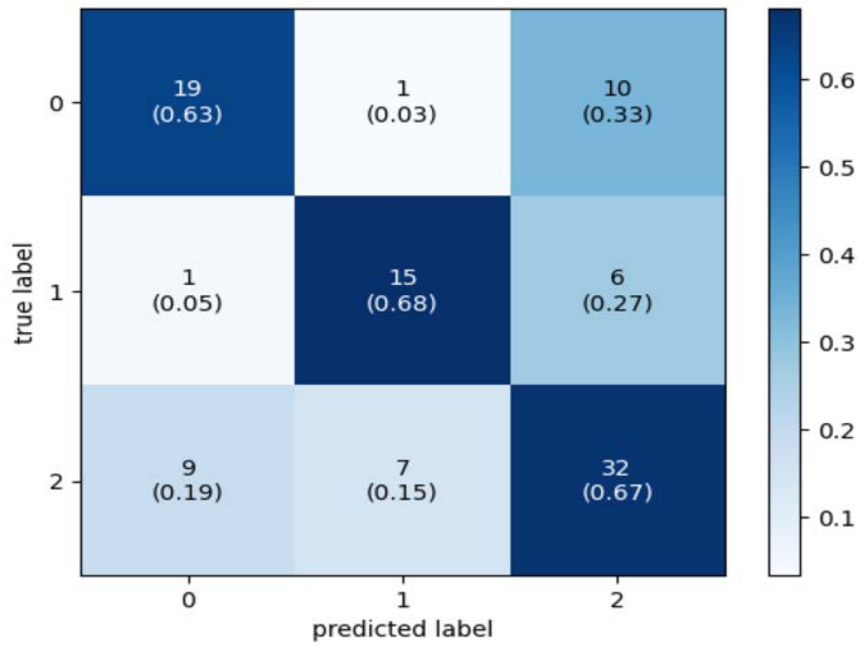
Figure 12. (a) Training (blue) and Validation (red) accuracy generated from using the four activation functions for VGG16 network (b) Confusion Matrix of predictions on the testing dataset with the VGG16 model and Softplus function.

4.1.6. Activation function

After all the other hyperparameters were tuned, the models were finally trained three times with each activation functions and the best-trained model based on the validation accuracy was stored. Figure 10(a) shows that ReLU outperforms the other activation functions in terms of validation accuracy (and training accuracy). The CNN model with ReLU achieved about 96% training accuracy and 62% validation accuracy, which was higher than that of models developed with ReLU, Sigmoid and Tanh. Figure 10(b) shows the training and validation loss and accuracy plots found based on model trained with ReLu.



(a)



(b)

Figure 13. (a) Training (blue) and Validation (red) accuracy generated from using the four activation functions for VGG19 network (b) Confusion Matrix of predictions on the testing dataset with the VGG19 model and Sigmoid function.

Table 2: Comparison among various network architectures

Network architectures	Recall	Precision	Test accuracy
Model developed with simple CNN	0.59	0.60	61%
Model developed with VGG16	0.71	0.70	72%
Model developed with VGG19	0.66	0.65	66%

4.2. Evaluation

After training the custom CNN with Softplus, the test dataset was used to generate class predictions. The confusion matrix generated from the custom CNN with the best activation function indicates that model accuracy of

27% for detecting misprints and 65% for detecting colour spots (Figure 11).

The hyperparameters that indicated the highest accuracy were used to run the learning models using the network architectures VGG-16, VGG-19, DenseNet-201, Inception-V3, and Xception. All four activation functions were attempted in the same manner consistent with the process used for the CNN (Section 4.1). The training and validation showed that DenseNet-201, Inception-V3, and Xception had very poor performance compared to custom CNN, VGG-16 and VGG-19. Therefore, the training and validation accuracy, and confusion matrix plots generated from the VGG16 (Figure 12(a,b)) and VGG19 (Figure 13(a,b)) were only presented here. The VGG16 model achieved 59% accuracy for detecting misprints and 73% for detecting colour spots (Figure 12(b)). The VGG19 model suggested 68% accuracy for detecting misprints and 67% for detecting colour spots (Figure 13b).

The CNN model was compared with VGG-16 and VGG-19 based models to determine the final model for printed fabric defect detection (Table 2). The VGG16 indicated highest recall (0.71), precision (0.70) and test accuracy (72%) compared to both the VGG19 model and the custom CNN. The CNN indicated comparatively less recall (0.59), precision (0.60) and accuracy (61%) compared to both VGG16 and VGG19. This is likely attributable to the CNN's simpler architecture and comparatively smaller training dataset compared to the VGG16 and VGG19 models.

Although previous research showed higher accuracy on defect detection; such as Tiwari and Harma (2015) and Kang et al. (2015) achieved 93.2% and 93% accuracy, respectively, neither of these research used popular and effective CNN model to classify defects among unknown test images. Moreover, these researches primarily focused on finding fabric manufacturing defects such as hole, needle mark, and cracks on the printed fabric instead of detecting printing faults such as colour spots and misprints. Hence our research proposes a novel methodology that is entirely different from the methodologies presented in the previous research. Nevertheless, the accuracy achieved by VGG16 model (72%) still needs to be improved in the future. The anticipated reasons for this low accuracy are small dataset and complex print designs. Although image augmentation was used to expand the training dataset, the initial database was too small for a computer vision-based image classification research. Furthermore, while finding a defect on a grey or white fabric is easy for human eyes, detecting coloured spots or print variation on a printed fabric would be more difficult, which applies to computer as well.

5. Research limitations and future research directions

A number of limitations associated with the study should be noted. Primarily, the relatively small dataset likely limited training, validation and test accuracy. As this work continues fabric images will be continuously added to the dataset to support future inquiry. Although the existing dataset is populated by actual print images, software-generated images can also be used in order to expand the dataset. To focus this research, print patterns included in the database were limited to floral and dot prints, which represent only a portion of textile print production. Future research can benefit from the inclusion of additional print patterns such as those with geometric shapes. Likewise, this work focuses on two major defects and can be expanded to include additional challenges (i.e. colour bleeding and wicking) to provide a more comprehensive tool for fabric ADD.

Researchers suggest that Stochastic Gradient (SGD) and Adabelief may potentially generate improvements in learning accuracy compared to the Adam optimiser (Keskar & Socher, 2017; Zhuang et al., 2020). Therefore, the incorporation of these optimisers into future research efforts is a logical next step in developing robust models. Additionally, a broader range of hyperparameter values can provide more nuanced insight into model performance. Moreover, this research can be used further with a line camera and server setup to facilitate real-time printed fabric defect detection in the industry.

6. Conclusion

Automatic defect detection for fabric printing, which involves numerous colours, is particularly challenging for producers and ultimately costly to the entire fashion supply chain. This research provides a cursory analysis of the potential to integrate machine learning into traditional printing processes. Additionally, it develops an automatic defect detection (ADD) model employing a convolutional neural network (CNN) on the training, validation and testing image dataset to facilitate real-time deployment in the rotary screen printing process. Continued advances in ADD, aided by machine learning suggest a important potential for reducing waste and enhancing quality in fashion supply chains that are heavily reliant on printed fabrics to serve their respective markets.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The research received research support grant from the Institute Textile Engineers and Technologists, Bangladesh.

ORCID

Samit Chakraborty http://orcid.org/0000-0002-4315-1779

References

Abd Jelil, R., Zeng, X., Koehl, L., & Perwuelz, A. (2013). Modeling plasma surface modification of textile fabrics using artificial neural networks. *Engineering Applications of Artificial Intelligence*, *26*(8), 1854–1864. doi:10.1016/j.engappai.2013.03.015

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. 2017 International Conference on Engineering and Technology (ICET), 1–6. https://doi.org/10.1109/ICEngTechnol.2017.8308186

Anh, N. T. H., & Giao, B. C. (2019). An empirical study on fabric defect classification using Deep network models. In T. K. Dang, J. Küng, M. Takizawa, & S. H. Bui (Eds.), *Future data and Security engineering* (Vol. 11814, pp. 739–746). Springer International Publishing. https://doi.org/10.1007/978-3-030-35653-8_54

Balaji, K., & Lavanya, K. (2019). Medical image analysis with deep neural networks. In A. K. Sangaiah (Ed.), *Deep learning and parallel computing environment for bioengineering systems* (pp. 75–97). Elsevier. https://doi.org/10.1016/B978-0-12-816718-2.00012-9

Bandara, P., Bandara, T., Ranatunga, T., Vimarshana, V., Sooriyaarachchi, S., & Silva, C. D. (2018). Automated fabric defect detection. 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer), 119–125. https://doi.org/10.1109/ICTER.2018.8615491

Basirat, M., & Roth, P. M. (2018). The quest for the golden activation function. ArXiv:1808.00783 [Cs, Stat]. http://arxiv.org/abs/1808.00783

Çelik, Hİ, Dülger, L. C., & Topalbekiroğlu, M. (2014). Development of a machine vision system: Real-time fabric defect detection and classification with neural networks. *The Journal of The Textile Institute*, *105*(6), 575–585. https://doi.org/10.1080/00405000.2013.827393

Chakraborty, S. (2016). A detailed study on environmental sustainability in knit composite industries of Bangladesh. *American Journal of Environmental Protection*, *5*(5), 121. https://doi.org/10.11648/j.ajep.20160505.13

Chakraborty, S., Moore, M., & Chapman, L. P. (2021a). Automatic defect detection (ADD) approaches in textiles and apparel. *Journal of Textile and Apparel, Technology and Management*, Special Issue(2021), 1–24.

Chakraborty, S., Moore, M., & Parrillo-Chapman, L. (2021b). Automatic defect detection of print fabric using convolutional neural network. ArXiv:2101.00703 [Cs]. http://arxiv.org/abs/2101.00703

Čiarnienė, R., & Vienažindienė, M. (2014). Agility and responsiveness managing fashion supply chain. *Procedia – Social and Behavioral Sciences*, *150*, 1012–1019. https://doi.org/10.1016/j.sbspro.2014.09.113

Cotton Inc. (2020). Standard fabric defect glossary. Cotton Incorporated. https://www.cottoninc.com/quality-products/textile-resources/fabric-defect-glossary/

Das, S., Wahi, A., Sundaramurthy, S., Thulasiram, N., & Keerthika, S. (2019). Classification of knitted fabric defect detection using artificial neural networks. 2019 International Conference on Advances in Computing and Communication Engineering (ICACCE), 1–5. https://doi.org/10.1109/ICACCE46606.2019.9079951

DeepAI. (2020). *Feature extraction*. DeepAI. https://deepai.org/machine-learning-glossary-and-terms/feature-extraction#:~:text=Feature%20extraction%20is%20a%20process,more%20manageable%20groups%20for%20processing

Devarakonda, A., Naumov, M., & Garland, M. (2018). AdaBatch: Adaptive Batch Sizes for Training Deep Neural Networks. ArXiv:1712.02029 [Cs, Stat]. http://arxiv.org/abs/1712.02029

Dsouza, J. (2020). *Image classification in data science*. Towardsdatascience. https://towardsdatascience.com/image-classification-in-data-science-422855878d2a

Eldessouki, M. (2018). Computer vision and its application in detecting fabric defects. In *Applications of computer vision in fashion and textiles* (pp. 61–101). Elsevier. https://doi.org/10.1016/B978-0-08-101217-8.00004-X

Essid, O., Laga, H., Samir, C., & Pan, Z. (2018). Automatic detection and classification of manufacturing defects in metal boxes using deep neural networks. *PLOS ONE*, *13*(11), e0203192. doi:10.1371/journal.pone.0203192

Fabijańska, A., & Jackowska-Strumiłło, L. (2012). Image processing and analysis algorithms for yarn hairiness determination. *Machine Vision and Applications*, *23*(3), 527–540. https://doi.org/10.1007/s00138-012-0411-y

Gao, C., Zhou, J., Wong, W. K., & Gao, T. (2019). Woven fabric defect detection based on convolutional neural network for binary classification. In W. K. Wong (Ed.), *Artificial intelligence on fashion and textiles* (Vol. 849, pp. 307–313). Hong Kong: Springer International Publishing. https://doi.org/10.1007/978-3-319-99695-0_37

Ghaderpanah, P., Mokhtari, F., & Latifi, M. (2015). Evaluation of false-twist textured yarns by image processing. *Indian Journal of Fibre & Textile Research*, *40*, 399–404.

Ghosh, A., Hasnat, A., Halder, S., & Das, S. (2014). A proposed system for cotton yarn defects classification using probabilistic neural network. *International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014)*, 1–6. Jaipur, India: IEEE.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge, USA: The MIT Press.

Guan, M., Zhong, Z., Rui, Y., Zheng, H., & Wu, X. (2019). *Defect detection and classification for plain woven fabric based on deep learning*. 2019 Seventh International Conference on Advanced Cloud and Big Data (CBD), 297–302. https://doi.org/10.1109/CBD.2019.00060

Habib Md., T., & & Rokonuzzaman, M. (2010). *Selection of distinguishing features for fabric defect classification using neural network*. 2010 13th International Conference on Computer and Information Technology (ICCIT), 482–487. https://doi.org/10.1109/ICCITECHN.2010.5723905

Hanbay, K., Talu, M. F., Özgüven, ÖF, & Öztürk, D. (2019). Real-time detection of knitting fabric defects using shearlet transform. *Tekstil ve Konfeksiyon*, *29*(1), 1–10. https://doi.

org/10.32710/tekstilvekonfeksiyon.482888
Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18* (7), 1527–1554. doi:10.1162/neco.2006.18.7.1527
Ide, H., & Kurita, T. (2017). *Improvement of learning for CNN with ReLu activation by sparse regularization*. 2017 International Joint Conference on Neural Networks (IJCNN), 2684–2691. https://doi.org/10.1109/IJCNN.2017.7966185
Jahangir Alam, S. M., Hu, G., & Roy, S. (2021). Analysis of a printed complex image quality checking method of fabric cloth for development of an automated quality checking system. *Signal, Image and Video Processing*, 15(1), 195–203.
Jing, J., & Ren, H. (2020). Defect detection of printed fabric based on RGBAAM and image pyramid. *Autex Research Journal*, *20,* 1–7.
Kabir, S. M. F., Chakraborty, S., Hoque, S. M. A., & Mathur, K. (2019). Sustainability assessment of cotton-based textile wet processing. *Clean Technologies*, *1*(1), 232–246. doi:10.3390/cleantechnol1010016
Kang, X., Yang, P., & Jing, J. (2015). Defect detection on printed fabrics via Gabor filter and regular band. *Journal of Fiber Bioengineering and Informatics*, *8*(1), 195–206. doi:10.3993/jfbi03201519
Kavukcuoglu, K., Sermanet, P., Boureau, Y.-L., Gregor, K., Mathieu, M., & LeCun, Y. (2010). Learning convolutional feature hierarchies for visual recognition. *Advances in Neural Information Processing Systems*, *23*(NIPS 2010), 1090–1098.
Keskar, N. S., & Socher, R. (2017). Improving Generalization Performance by Switching from Adam to SGD. ArXiv:1712.07628 [Cs, Math]. http://arxiv.org/abs/1712.07628
Kuo, C.-F. J., Hsu, C.-T. M., Chen, W.-H., & Chiu, C.-H. (2012). Automatic detection system for printed fabric defects. *Textile Research Journal*, *82*(6), 591–601. doi:10.1177/0040517511426615
LeCun, Y. (2020). LeNet-5, convolutional neural networks. Yann.Lecun.Com. http://yann.lecun.com/exdb/lenet/
Li, Y., Ai, J., & Sun, C. (2013). Online fabric defect inspection using smart visual sensors. *Sensors*, *13*(4), 4659–4673. doi:10.3390/s130404659
Li, M., Cui, S., & Xie, Z. (2015). Application of Gaussian mixture model on defect detection of print fabric. *Journal of Textile Research*, *36*(8), 94–98.
Li, X., Grandvalet, Y., & Davoine, F. (2020). A baseline regularization scheme for transfer learning with convolutional neural networks. *Pattern Recognition*, *98*, 107049, 1–10. doi:10.1016/j.patcog.2019.107049
Luo, P., Wang, X., Shao, W., & Peng, Z. (2019). *Towards understanding regularization in batch normalization*. ArXiv:1809.00846 [Cs, Stat]. http://arxiv.org/abs/1809.00846
Md Mottalib, M., Md. Habib, T., Rokonuzzaman, M., & Ahmed, F. (2015). Fabric defect classification with geometric features using Bayesian classifier. Proceedings of 2015 3rd International Conference on Advances in Electrical Engineering (ICAEE 2015). http://ieeexplore.ieee.org/servlet/opac?punumber=7502220
Mei, S., Wang, Y., & Wen, G. (2018). Automatic fabric defect detection with a multi-Scale convolutional denoising autoencoder network model. *Sensors*, *18*(4), 1064. doi:10.3390/s18041064
Miles, L. W. C. (1994). *Textile printing* (2nd edn). Bradford: Society of Dyers and Colourists.
Missinglink. (2020). *Convolutional neural networks fully connected layers in convolutional neural networks: the complete guide*. Missinglink.Ai. https://missinglink.ai/guides/convolutional-neural-networks/fully-connected-layers-convolutional-neural-networks-complete-guide/
Mlnotebook. (2017). Convolutional neural networks—basics an introduction to CNNs and deep learning. *MLNOTEBOOK*. https://mlnotebook.github.io/post/CNN1/
Ngan, H. Y. T., Pang, G. K. H., & Yung, N. H. C. (2011). Automated fabric defect detection – a review. *Image and Vision Computing*, *29*(7), 442–458. doi:10.1016/j.imavis.2011.02.002
Ouyang, W., Xu, B., Hou, J., & Yuan, X. (2019). Fabric defect detection using activation layer embedded convolutional neural network. *IEEE Access*, *7*, 70130–70140. doi:10.1109/ACCESS.2019.2913620
Pan, R., Gao, W. W., Qian, X., & Zhang, X. (2010). Defect detection of printed fabrics using normalized cross correlation. *Journal of Textile Research*, *31*(12), 134–138.
Park, S., & Kwak, N. (2017). Analysis on the dropout effect in convolutional neural networks. In S.-H. Lai, V. Lepetit, K. Nishino, & Y. Sato (Eds.), *Computer vision – ACCV 2016* (Vol. 10112, pp. 189–204). Taipei: Springer International Publishing. https://doi.org/10.1007/978-3-319-54184-6_12
Rahaman, G. M. A., & Hossain, M. M. (2009). *Automatic defect detection and classification technique from image: a special case using ceramic tiles*. ArXiv:0906.3770 [Cs]. http://arxiv.org/abs/0906.3770
Schneider, D., & Aach, T. (2012). *Vision-based in-line fabric defect detection using yarn-specific shape features* (P. R. Bingham & E. Y. Lam, Eds.; p. 83000G). https://doi.org/10.1117/12.907268
Schneider, D., Holtermann, T., & Merhof, D. (2014). A traverse inspection system for high precision visual on-loom fabric defect detection. *Machine Vision and Applications*, *25*(6), 1585–1599. doi:10.1007/s00138-014-0600-y
Sharan, R. (2011). *Dyeing, printing & processing defects*. https://www.slideshare.net/rajeevsharan/dyeing-printing-processing-defects
Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6* (1), 1–48.
Simonyan, K., & Zisserman, A. (2012). *Very deep convolutional networks for large-scale visual recognition*. https://www.robots.ox.ac.uk/~vgg/research/very_deep/#pub
Stanford. (2020). CS231n convolutional neural networks for visual recognition. *Stanford.Edu*. https://cs231n.github.io/convolutional-networks/
Stitchdiary. (2017). *Classify the fault: defects due to dyeing & printing*. https://medium.com/@stitchdiary/classify-the-fault-defects-due-to-dyeing-printing-23dc4df50a03
Tabassian, M., Ghaderi, R., & Ebrahimpour, R. (2011). Knitted fabric defect classification for uncertain labels based on Dempster–Shafer theory of evidence. *Expert Systems with Applications*, *38*(5), 5259–5267. doi:10.1016/j.eswa.2010.10.032

Takase, T., Oyama, S., & Kurihara, M. (2018). Effective neural network training with adaptive learning rate based on training loss. *Neural Networks*, *101*, 68–78. doi:10.1016/j.neunet.2018.01.016

Tavanai, H., Palhang, M., Hosseini, S. A., & Mollahosseini, H. (2006). Detection of color displacement in fabric printing through image analysis. *Journal of the Textile Institute*, *97* (4), 333–340. doi:10.1533/joti.2005.0148

Tiwari, V., & Harma, G. (2015). Automatic fabric fault detection using morphological operations on bit plane. *International Journal of Computer Science and Network Security*, *15*(10), 1–6.

Tong, L., Wong, W. K., & Kwong, C. K. (2017). Fabric defect detection for apparel industry: A nonlocal sparse representation approach. *IEEE Access*, 1–1. doi:10.1109/ACCESS.2017.2667890

Vernon, D. (1991). *Machine vision-Automated visual inspection and robot vision*. Hoboken, NJ: Prentice Hall.

Wan, L., Zeiler, M., Zhang, S., Cun, Y. L., & Fergus, R. (2013). Regularization of neural networks using DropConnect. *Proceedings of the 30th International Conference on Machine Learning, PMLR*, *28*(3), 1058–1066.

Wang, L., Xu, B., & Gao, W. (2018). Multi-perspective measurement of yarn hairiness using mirrored images. *Textile Research Journal*, *88*(6), 621–629. doi:10.1177/0040517516685281

Wen, J. J., & Wong, W. K. (2018). Fundamentals of common computer vision techniques for fashion textile modeling, recognition, and retrieval. In *Applications of computer vision in fashion and textiles* (pp. 17–44). Elsevier. https://doi.org/10.1016/B978-0-08-101217-8.00002-6

Wong, W. K., & Jiang, J. L. (2018). Computer vision techniques for detecting fabric defects. In *Applications of computer vision in fashion and textiles* (pp. 47–60). Elsevier. https://doi.org/10.1016/B978-0-08-101217-8.00003-8

Xia, D., Jiang, G., & Ma, P. (2016). Warp-knitted fabric defect segmentation based on the non-subsampled wavelet-packet-based Contourlet transform. *Textile Research Journal*, *86* (19), 2043–2055. https://doi.org/10.1177/0040517515619356

Yumpu. (2015). *A Defect Analysis of Rotary Screen vs. Digital Textile Printing*. Yumpu.Com. https://www.yumpu.com/en/document/read/32913332/a-defect-analysis-of-rotary-screen-vs-digital-textile-printing

Zhao, X., Zhang, M., & Zhang, J. (2021). Ensemble learning-based CNN for textile fabric defects classification. *International Journal of Clothing Science and Technology*, 1–15.

Zhuang, J., Tang, T., Ding, Y., Tatikonda, S., Dvornek, N., Papademetris, X., & Duncan, J. S. (2020). AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients. ArXiv:2010.07468 [Cs, Stat]. http://arxiv.org/abs/2010.07468