Computers in Industry 164 (2025) 104187

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.sciencedirect.com/journal/computers-in-industry

Automated corner grading of trading cards: Defect identification and confidence calibration through deep learning

Lutfun Nahar [a,b], Md. Saiful Islam [c,*], Mohammad Awrangjeb [a], Rob Verhoeve [d]

[a] Griffith University, Nathan, QLD 4111, Australia
[b] International Islamic University Chittagong, Kumira, Chittagong 4318, Bangladesh
[c] School of Information and Physical Sciences, The University of Newcastle, Callaghan, NSW 2308, Australia
[d] Media8, Newheath Dr, Arundel QLD 4214, Australia

ARTICLE INFO

Keywords:
Defect detection
Classification
Confidence calibration
Deep learning
Transfer learning

ABSTRACT

This research focuses on trading card quality inspection, where defects have a significant effect on both the quality inspection and grading. The present inspection procedure is subjective which means the grading is sensitive to mistakes made by individuals. To address this, a deep neural network based on transfer learning for automated defect detection is proposed with a particular emphasis on corner grading which is a crucial factor in overall card grading. This paper presents an extension of our prior study, in which we achieved an accuracy of 78% by employing the VGG-net and InceptionV3 models. In this study, our emphasis is on the DenseNet model where convolutional layers are used to extract features and regularisation methods including batch normalisation and spatial dropout are incorporated for better defect classification. Our approach outperformed prior findings, as evidenced by experimental results based on a real dataset provided by our industry partner, achieving an 83% mean accuracy in defect classification. Additionally, this study investigates various calibration approaches to fine-tune the model confidence. To make the model more reliable, a rule-based approach is incorporated to classify defects based on confidence scores. Finally, a human-in-the-loop system is integrated to inspect the misclassified samples. Our results demonstrate that the model's performance and confidence are expected to improve further when a large number of misclassified samples, along with human feedback, are used to retrain the network.

1. Introduction

A trading card, also known as a collectible card (e.g., one sold on Collectible Madness), is fabricated paper or cardboard and usually has different artwork, pictures, facts or figures that are associated with certain themes such as sports, video games and other topics. Enthusiasts collect these cards extensively and they are often connected to specific interests. Well-known collectibles such as Magic: The Gathering and Pokémon cards have intrinsic value that is closely related to their condition and quality (Grading). A vital component of the industry is the quality assessment or card grading which is carried out by carefully assessing four key factors: surface, edge, corner and centring. All these four factors are taken into consideration when evaluating the quality and grading of a trading card using a predefined scale, ranging from 1 to 10 (PSA, 2024). However, these cards are being examined and graded manually which makes the procedure prone to irregularities and subjective. During manual grading, handling, touching and shuffling of the cards may cause some deterioration. This comprises frayed corners, bent edges and surface blemishes. Furthermore, assessing a single card takes careful attention and can be time-consuming which slows down the grading process. In response to these challenges, this research introduces a novel approach for automating the detection of defects in one of the four grading criteria: corner grading.

When performing corner grading, the grader carefully examines each corner and searches for defects like dents, scratches, fraying and discolouration. The final corner grade that is assigned to the card is determined by summing up the deduction points which are initially estimated based on the severity of these defects. According to this evaluation, a card with a deduction score of one (1) is in poor condition, 0.5 indicates a card with medium faults, 0.25 represents a card has minor defects while zero (0) indicates it is in good condition with no defects. A trading card has eight corners (four on the front and four on the back) and each corner is categorised into a distinct class based on the type and size of defects. The final corner grade for the card is determined by deducting the sum of the eight deduction points from

[*] Corresponding author.
E-mail addresses: lutfun.nahar@griffithuni.edu.au (L. Nahar), Saiful.islam@newcastle.edu.au (M.S. Islam), m.awrangjeb@griffith.edu.au (M. Awrangjeb), rob.verhoeve@psgrading.net (R. Verhoeve).

https://doi.org/10.1016/j.compind.2024.104187
Received 7 April 2024; Received in revised form 4 September 2024; Accepted 12 September 2024
Available online 19 September 2024

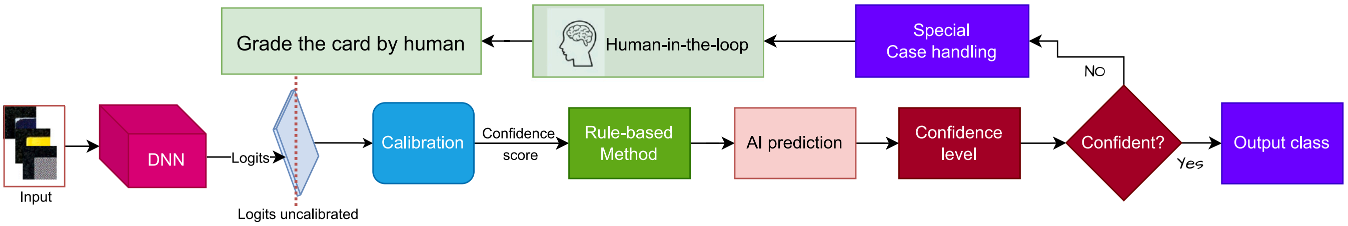


Fig. 1. Schematic diagram of the main contribution in this paper.

the highest grade which is 10. For example, if for a card three of its 8 corners are in good shape while the other five have minor defects, its grade would be ascertained as follows: $10-(0.25\times5+0\times3) = 10-1.25 = 8.75$.

In our prior work (Nahar et al., 2023), various transfer learning based Convolutional Neural Networks (CNNs) are applied for automatic defect detection. Although these models provide a good accuracy for corner classification but accuracy alone is inadequate in real-world situation. The system reliability is necessary to determine whether the system is confident enough for the specific instance in corners with a good accuracy. For system reliability, the network's confidence, which is commonly shown by the softmax output, is essential. Deep neural networks (DNNs) are often overconfident, which means that their actual output probabilities may not be a realistic reflection of the probabilities, leading to poor calibration. The machine learning community has noticed this calibration difficulty, especially for deep networks. The uncalibrated model can lead to dangerous situations, such as in autonomous vehicles, where decisions are made depending on the confidence score in detected objects (Bojarski et al., 2016; He et al., 2015). In healthcare (Jiang et al., 2011), there is a risk if a life-threatening disease is identified incorrectly with high confidence. Therefore, calibration increases the system reliability by minimising the confidence for misclassified samples while maintaining confidence for correctly classified ones.

To address this issue, various calibration approaches, such as Temperature Scaling (TS), Matrix Scaling (MS), and Vector Scaling (VS) (Guo et al., 2017; Platt, 2000) are explored in these study for corner defect classification. Besides integrating the calibration techniques with the previous work (Nahar et al., 2023), this research also works to enhance the accuracy of the corner classification by exploring more deeper transfer learning based model, which provides more distinct input features through shortcut connections of different lengths, and effectively reduce the vanishing gradient problem. Some advanced techniques are employed, for instance, focal loss (Lin et al., 2017) to solve the problem of class imbalance that frequently arises in machine learning tasks and data augmentation to expand the dataset. To further enhance the model's performance, regularisation techniques including batch normalisation and spatial dropout are incorporated. This study uses a real-world dataset collected from our industry partner (Media8). Moreover, to make the model more robust a rule-based method with human-in-the-loop is integrated to assess the instance, where machine shows low confidence. The main contributions of this study shown in Fig. 1 are as follows.

- This study focuses on a cutting-edge deep learning framework, especially a transfer learning based DNN model, DenseNet, to automate the trading card's corner grading process. In experiments, different settings are proposed by freezing layers at different depths, and finally, the best approach yields superior corner grading results. It provides accurate and efficient grading of card corners by using deep learning.
- To make the model more reliable, various calibration techniques are explored and fine-tuning the probability offers better performance.
- Additionally, a rule-based method is introduced that utilises the confidence scores to determine the class of the defect which enhances the accuracy in corner grading. Finally, the human-in-the-loop system is included to inspect the instance which machine's confidence level in its decisions falls below a certain threshold.

The rest of the paper is organised as follows: Section 2 provides a summary of the most recent works in industrial defect detection techniques. Our proposed methodology is described in Section 3. Section 4 presents the experimental results, while Section 5 concludes the paper by highlighting the contributions and future works.

2. Related work

While there is a shortage of dedicated research, particularly addressing the corner grading of trading cards, a number of studies have explored in quality inspection on other domain in industrial perspective using deep learning techniques and computer vision. In a broader context defect detection methods can be mainly grouped into two: semi-automated and automated. Semi-automated methods rely on manual feature engineering, while automated methods, which are deep learning-based, learn features directly from raw data. This makes automated methods more adaptable and effective for complex and diverse datasets. Related work on these two categories is discussed in this section.

2.1. Semi-automated methods

Semi-automated method such as general image processing and classical machine learning techniques are highly effective when defects on the products having little variation and defects appear on surfaces in a consistent pattern. For example, To identify corner defects on tiles, Singh and Yadav (2014b,a) utilised general image processing techniques and mathematical operations on tiles, achieving a 96% accuracy rate. Matić et al. (2013) introduced edge and contour-based methods for detecting cracked corners on tiles, achieving a commendable 90% accuracy. Yıldız et al. (2016) used K-nearest neighbour (KNN) to classify fabric defects with 96% accuracy, utilising features extracted through Gray-Level Co-Occurrence Matrix analysis. Lei and Zuo (2009) also applied KNN to identify gear cracks, employing a two-stage feature selection and weighting technique. Their method achieved 90% accuracy.

2.2. DL-based automated methods

Defect detection technique can be categorised into two main tasks based on the objectives: defect detection or classification and defect localisation. Defect detection or classification focuses on identifying whether a defect is present and determining its type. On the other hand, defect localisation aims to find the exact location of the defect within an image or surface. Both tasks can be approached using supervised or unsupervised learning strategies. Supervised techniques provide high performance in defect inspection but require large labelled datasets, while unsupervised learning avoids the need for large labelled datasets but generally offers lower performance. Although deep neural networks

(DNNs) perform very well in detection but struggle with overconfidence or underconfidence, which is a key challenge in machine learning. Research is currently focused on studying factors that impact the calibration of these networks.

The following subsections are organised as follows: Section 2.2.1 describes defect detection or classification, Section 2.2.2 covers defect localisation and Section 2.2.3 discusses related work on model calibration.

2.2.1. Defect detection or classification

In the wider scope of defect classification or detection across various products, supervised models such as CNN-based, pre-trained, transfer learning and hybrid network with attention mechanism models have demonstrated exceptional performance. Moreover, unsupervised models such as Generative Adversarial Networks (GAN), are effective in scenarios where there are limited labelled data. Related work on these models is described in this subsection.

- CNN-based Models: Lin et al. (2019) implemented a CNN-based model, LEDNet, for defect classification on LED chips, achieving 95% accuracy. Liu et al. (2022) successfully applied deep convolutional neural networks to detect spots and print error on fabric, resulting in an affordable and highly accurate system. For CNN model with random weight, Yi et al. (2016) built up an end-to-end surface defects recognition system that generates saliency maps as the classification results of seven types of steel strip defects. Kim et al. (2021) also developed a CNN model to inspect tiny defects on circuit boards with a skip-connected convolutional autoencoder. This autoencoder was trained to decode original non-defect images from defect images. The experimental results revealed that a simple autoencoder model delivered promising performance, achieving a detection rate of up to 98%.
- Transfer Learning-based Models: Research (Konovalenko et al., 2022; Kim et al., 2021) shows that it is more effective to use transfer learning-based model for training rather than building a CNN from scratch for a small dataset. For example, Rolland et al. (2022) achieved a remarkable 96% accuracy by employing a transfer learning-based model, ResNet, for scoring the hairiness of cotton leaves. Yang et al. (2020) employed transfer learning and a pre-trained SqueezeNet model to address the critical issue of inspecting laser welding defects in power batteries. The study collected 34,537 images to create 2-class and 7-class datasets. The model achieved 99.57% accuracy in the 2-class task and 95.58% in the 7-class task and outperformed other CNN models. Konovalenko et al. (2022) focused on the development and exploration of 14 neural networks aiming to identify defects on the surface of metal. A modified VGG16 (Liu et al., 2019) was suggested to detect texture-based defects on fabrics with 98.1% accuracy, while Chakraborty et al. (2022) proposed a CNN method for identifying the printing defect, hole, spot on fabric with good accuracy.
- Hybrid Networks: Hybrid network and attention mechanism get more popular in defect identification. Sun et al. (2016) proposed a hybrid network that combines machine vision and artificial neural networks to automate the inspection of thermal fuses. Another work on surface defects also done by Liang et al. (2024) where they used CNN with lightweight attention mechanism, could predict the diverse and unpredictable defects caused by various shapes, random positions, and complex types. The model achieved an accuracy of 94.2% on custom datasets and performed exceptionally well on the public NEU-DET dataset. Ren et al. (2018) also employed a pretrained CNN model for surface inspection where patch-by-patch features were extracted and then these features were used to train the classification model. Finally, they obtained pixel wise prediction by using the trained multinomial logistic regression classifier over input image. They directly apply the DeCAF model (Donahue et al., 2013) for defect-specific feature extraction. In addition, several reviews (Tang et al., 2022; Saberironaghi et al., 2023; Bai et al., 2024; Jha and Babiceanu, 2023; Tulbure et al., 2022) are also available in the field of defect detection for industrial purposes to identify defects in various products. These reviews provide detailed insights into the methodologies, techniques, and advancements in defect detection across different industries.
- GAN-based Models: In industry applications, it is often challenging to obtain fully labelled data making unsupervised approaches particularly effective in such cases. The most popular unsupervised CNN models for defect detection is the GAN-based method. Lian et al. (2020) proposed a defect exaggeration model, where CNN network is combined with GAN to generate flawless image and identify tiny surface defects. To improve the defect recognition process, Niu et al. (2020) developed a surface defect-generation adversarial network and applied it on defect-free images to enlarge the defective dataset.

2.2.2. Defect localisation

In recent years, there has been a significant increase in scholarly research on visual defect localisation problems using object detection techniques, namely, YOLO based model. Kou et al. (2021) developed an end-to-end defect detection model based on YOLO-V3, utilising an anchor-free feature selection mechanism and specially designed dense convolution blocks to improve feature reuse, feature propagation, and network characterisation. Experiment showed that the proposed model outperformed other comparison models, achieving 71.3% mAP (mean average precision) on the GC10-DET dataset and 72.2% mAP on the NEUDET dataset.

Moreover, Meng et al. (2022) employed YOLOv5 and shuffleNetv2 to detect corner cracks on steel, achieving an outstanding accuracy of 99.64%. Yao and Li (2022) also proposed the YOLOv3-Tiny network for defect detection in the surface of light guide plates of LCD devices, which often have complex textures, low contrast, and various defect types. By incorporating overlapping pooling and a spatial attention mechanism, the network improved feature extraction. Experiments showed that the system achieved 99.50% mean average precision, a 99.61% F1-score and meeting high-precision. Dong et al. (2024) presented SNF-YOLOv8 network for real-time detection of surface cracks in large stamped metal parts. It detected large cracks with 98.8% accuracy and small cracks with 96.4% accuracy. Another work on surface defects in selective laser melting was proposed by Wang et al. (2022), who introduced a deep learning characterisations method and feature fusion. Their experiments outperformed other state-of-the-art methods in defect analysis.

2.2.3. Calibration of DL-based methods

DNNs, despite their advancement and high accuracy in numerous tasks, face difficulties with overconfidence or underconfidence. Addressing these challenges has become a key point in the machine learning community. Guo et al. (2017) explored some factors which influence the calibration of deep neural networks. Their work described that when a deep network is overfitted to Negative Log Likelihood (NLL) (Hastie et al., 2009), it achieves good accuracy but becomes overconfident, impacting calibration.

There are two types of DNN calibration techniques: measure-based and probabilistic. Under the influence of Bayesian theory, probabilistic methods (Bernardo and Smith, 2009) define all neural network parameters in priors and use them to estimate conditional probabilities. Although Monte Carlo (MC) dropout simplifies these methods but they are still time-consuming. Conversely, compared to probabilistic techniques, measure-based systems provide an easier calibration. These approaches aim to minimise network miscalibration by minimising a loss function which is measured by some calibration metrics. Measure-based approaches, including histogram binning (Zadrozny and

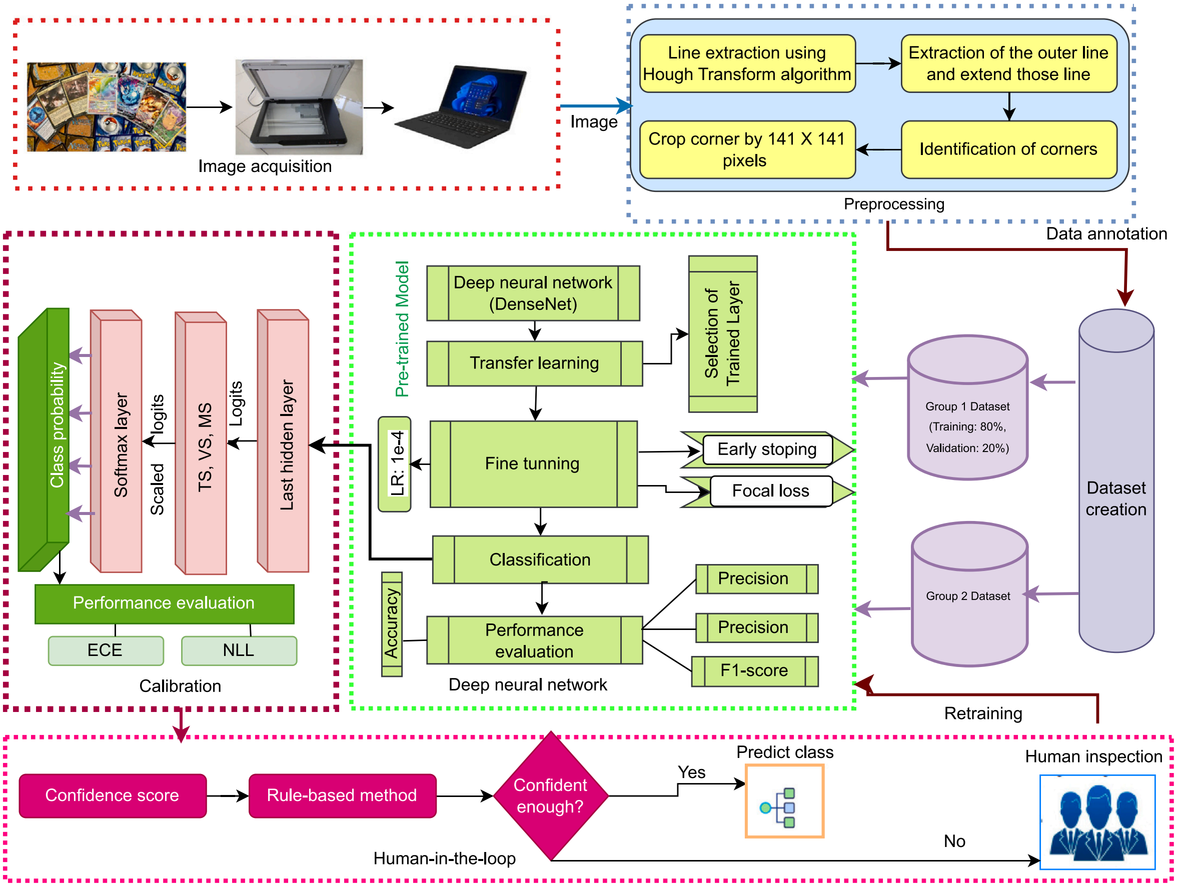


Fig. 2. Schematic diagram of the proposed framework. Notes: TS = Temperature Scaling, MS = Matrix Scaling, VS = Vector Scaling, ECE = Expected Calibration Error, and NLL = Negative Log Likelihood.

Elkan, 2001), isotonic regression (Zadrozny and Elkan, 2002), TS (Guo et al., 2017), platt-scaling (Platt, 2000) and Bayesian binning (Pakdaman Naeini et al., 2015), fine-tune the softmax layer without altering the DNN model weights.

In this paper, we explore the effectiveness of DenseNet-based (Huang et al., 2016) deep transfer learning for detecting defects in trading cards, leveraging its efficacy demonstrated in various real-life defect identification and classification methods (Zhu et al., 2020; Banús et al., 2021; Lu et al., 2021; Dai et al., 2021; Dong et al., 2022). Moreover, it focuses on measure-based calibration, using TS (Guo et al., 2017), MS, and VS (Platt, 2000) to achieve superior calibration. After calibration, confidence scores are used within the rule-based method for prediction, based on which an instance is subjected to human examination if its confidence level is lower than a predetermined threshold. We also demonstrate how these methods can be integrated into a human-in-the-loop system for industrial applications of vision-based analytics.

3. The proposed framework

Fig. 2 illustrates the schematic diagram of the proposed system. This process begins by capturing images through a scanner, and then go through a number of preprocessing steps to prepare the dataset. To guarantee quality and accuracy, human professionals meticulously label the dataset. The DenseNet model architecture (Chauhan et al., 2021; Huang et al., 2016) is selected as the main choice for classifying corner defects. Then, model calibration techniques are incorporated to make the model more reliable and enhance its performance. Finally, a rule-based approach is employed, where confidence ratings derived from the calibrated model are used to make classification decisions. Furthermore, a continuous learning system is introduced by using a human-in-the-loop analytic to refine and optimise the model.

A summary of the important symbols and acronyms used in the paper is presented in Table 1.

Table 1
Symbols and acronyms used in the paper.

Symbols/Acronyms	Meaning
CNN	Convolutional Neural Network
DNN	Deep Neural Network
ECE	Expected Calibration Error
FNR	False Negative Rate
FPR	False Positive Rate
HITLA	Human-in-the-loop Analytics
MS	Matrix Scaling
NLL	Negative Log-Likelihood
TS	Temperature Scaling
TP	True Positive
TNR	True Negative Rate
VS	Vector Scaling
TTCS	Top Two Confident Score
σ-Accuracy	TP within a subset and entire test dataset (Group 4), in threshold-based method
δ-Accuracy	TP within a subset and entire test dataset (Group 4), in distance-based method

3.1. Image acquisition

In this approach, image acquisition gets priority by emphasising on capturing high-quality images. We reduce issues such as light reflections and shadows in our method by switching from a camera

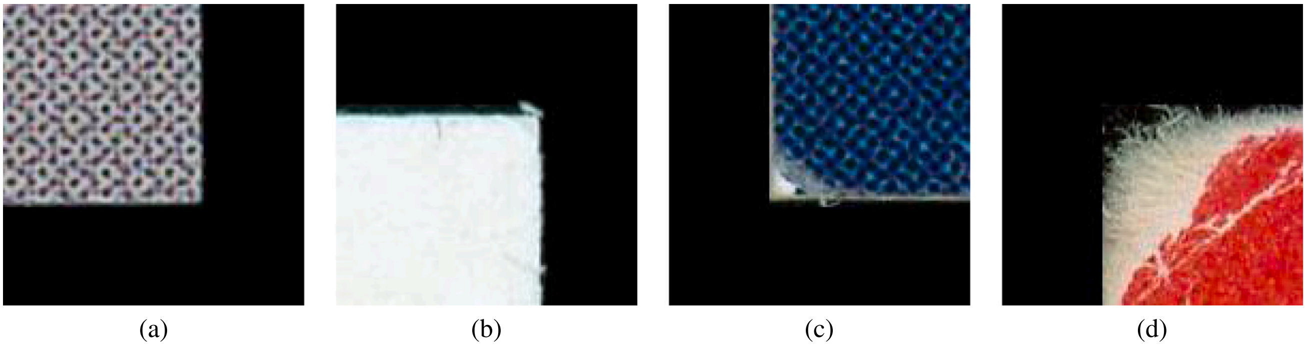


Fig. 3. (a) No defects (Class 0), (b) Small defects (Class 0.25), (c) Medium defects (Class 0.5), and (d) Large defects (Class 1).

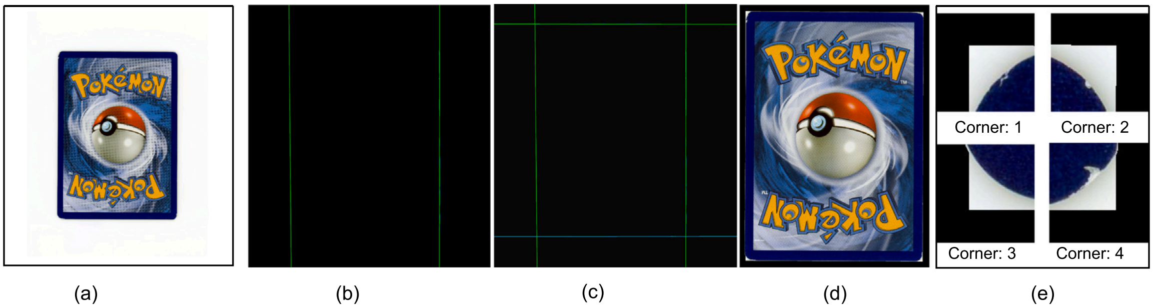


Fig. 4. (a) Original Image, (b) Vertical lines, (c) Vertical and horizontal lines, (d) Cropped image from background and (e) Corner images.

to an Epson V600 scanner, which provides high-quality images. This scanner ensures excellent image quality and constant lighting for efficient defect detection in card. Through experimental evaluation of photos at resolutions ranging from 300 to 2400 dpi, we determine that 1200 dpi provides the best balance between pre-processing time and defect visibility. Additionally, a standardise scanning procedures are used to maintain consistency and help with foreground–background separation. For example, trading cards with white borders are scanned against a black background whereas others are scanned against a white background.

3.2. Dataset

A real-world dataset consisting of 593 sports cards, provided by our industry partner Media8 (Media8), are scanned using an affordable scanner (e.g., Epson V600) in 1200 dpi, resulting in a total of 4744 corner datasets. The dataset is divided into two: Group 1 (80%) and Group 2 (20%). In Group 1 dataset, among 3795 photos, 1058 samples show small defects (Class 0.25), 302 images show medium defects (Class 0.5) and 587 photos show major defects (Class 1) and 1848 images have defect-free corners (Class 0). There are a total of 950 images available for Group 2 dataset where 265 samples show small defects (Class 0.25), 77 samples in Class 0.5 have medium defects, Class 0 has 461 images which are defect free and 147 photos show major defects (Class 1). Fig. 3 displays the sample images representing four different classes from the dataset.

3.3. Data preprocessing

After capturing an image, our preprocessing steps are employed to generate the corner dataset. Instead of relying on the Harris corner technique, we adopt an alternative corner detection method that prioritises identifying intersection points. The process begins with contour extraction, followed by the application of a bilateral filter to reduce noise in the image. Subsequently, the Hough transform algorithm is utilised on the contour image to extract lines. Upon obtaining the outer horizontal and vertical lines, we pinpoint the intersection points as corners. These corner points serve as the basis for cropping a region of 141 × 141 pixels, capturing individual corner images of the card. Given that the entire card measures 2.5 inches in width and 3.5 inches in height, this cropping size ensures the inclusion of relevant information around each corner.

Fig. 4 illustrates the sequential steps involved in creating our corner dataset. Fig. 4(a) represents the original image and Fig. 4(b) illustrates the extracted vertical lines using the Hough algorithm, Fig. 4(c) shows the intersection points, Fig. 4(d) represents the cropped image from background and Fig. 4(e) represents the subsequent cropping regions of 141×141 pixel from the original image. This method is designed to systematically capture and isolate corners for dataset creation.

3.4. Dataset splitting

This section explains how we partition and use the dataset among various components of our model architecture to ensure unbiased model performance. Fig. 5 illustrates the strategic diagram for dataset splitting. The dataset was first divided into two groups, Group 1 holding 80% of the data and Group 2 holding 20%. Group 1 dataset undergoes data augmentation with 80% of the augmented data being used for model training and the remaining 20% being utilised for validation. Table 4 displays the initial testing accuracy using Group 2 dataset.

Moreover, Group 2 is split up into two subsets: Group 3 (40%) and calibration (60%). The trained model M1 is calibrated and fine-tuned using the calibration dataset. Group 3 dataset is then divided into two: Group 4 (80%) and holdout (20%) datasets. Group 4 dataset is used as a testing dataset to assess the calibrated model (M2). Using the same dataset (Group 4), the performance of the uncalibrated model M1 and the calibrated model M2 are compared. Lastly, we find occurrences that fall below a threshold using Group 4 dataset. These instances are then used to retrain the model shown in the HITLA workflow, as described in Section 3.8.3. After retraining, we compare the performance of the retrained model to the initial model M1 using the holdout dataset as the testing data.

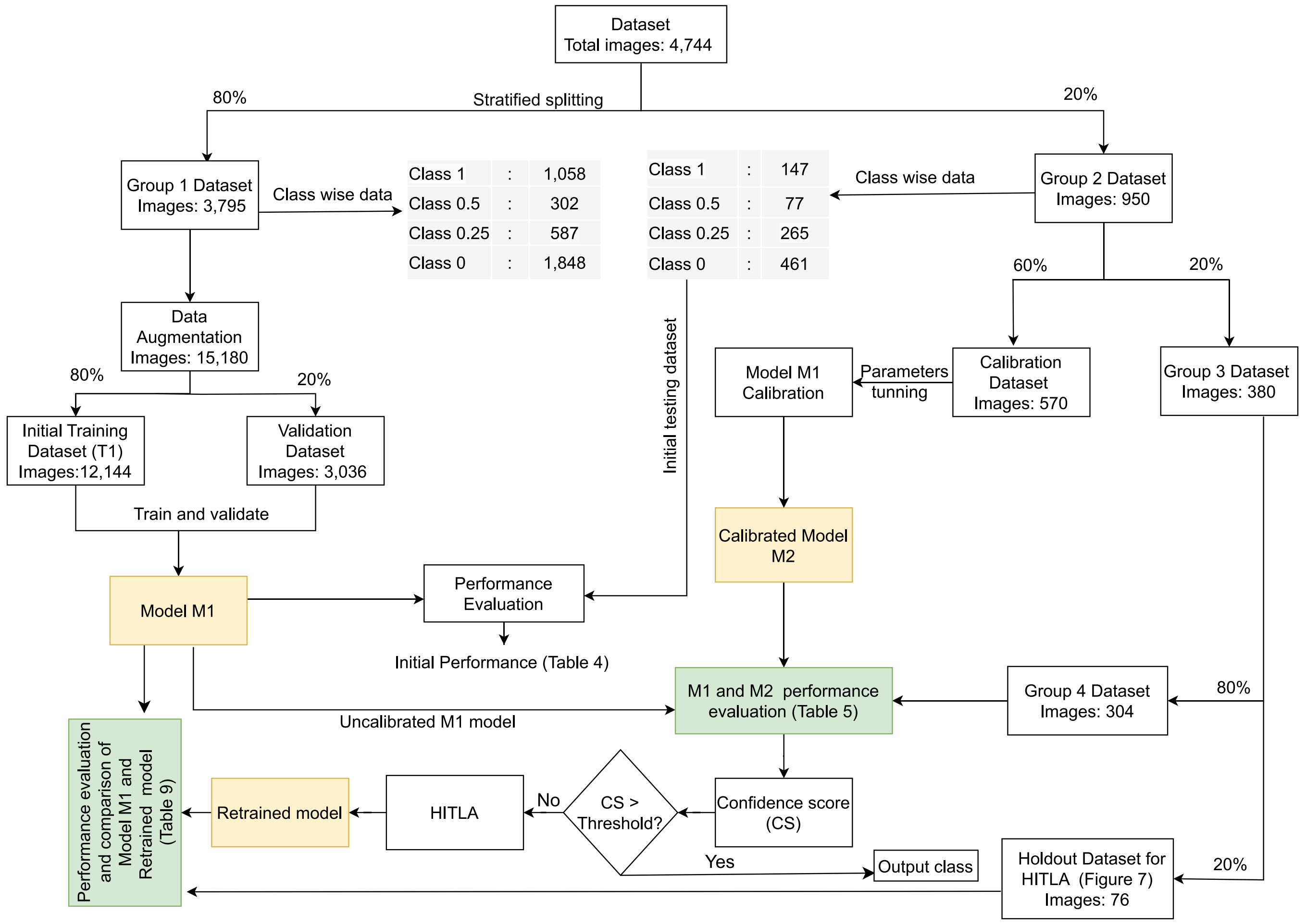


Fig. 5. Strategic dataset splitting framework. Note: No defects (Class 0), Small defects (Class 0.25), Medium defects (Class 0.5), and Large defects (Class 1), HITLA: Human-in-the-loop Analytics.

Therefore, the testing datasets (i.e., Group 4 and the holdout datasets) are not involved at all in model training, instead they are used for relevant model performance evaluation and comparison. For example, Group 4 dataset is used for comparing the calibrated and uncalibrated models. Also, the holdout dataset is used for performance comparison between the initial (uncalibrated) model and the retrained model via HITLA procedure.

3.5. Model architecture

The foundation of this research is built upon the DenseNet201 architecture (Huang et al., 2016). Since each layer in a conventional CNN is sequentially connected, problems like gradient explosion or disappearing make it difficult for the deep network to go deeper and wider. Enabling shortcut connections allows the DenseNet model to address the vanishing gradient problem by facilitating the skipping of at least two levels in the network architecture. The architecture of DenseNet consists of the input layer, transition layers, four Dense Blocks, and global average pooling layer. The transition layers consist of a 1×1 convolutional layer, a batch normalisation layer, and a 2×2 average pooling layer with a stride of 2. Specifically, global average pooling, similar to conventional pooling methods, undergoes a more aggressive reduction of a feature map from $w \times w \times c$ to $1 \times 1 \times c$.

Fig. 6 depicts the architecture of DenseNet201, utilised for automated corner grading in trading cards. However, generating features for the classification model demands a huge set of data and it can be expensive to train a huge dataset. Deep Transfer Learning (DTL) exhibits the potential to generate significant results with minimal data. Hyper parameter tuning in DTL models can improve the performance. Two major benefits of using DTL in this study are that it can train deep networks such as DenseNet on a small dataset (4744 images) (Nahar et al., 2023) and improve model robustness by initialising weights with pre-trained ImageNet weight (Deng et al., 2009). Moreover, if we train the model with less data, there is a possibility of overfitting the model. The easiest method to handle the overfitting issue is to increase the size of the dataset by employing some data augmentation techniques.

Therefore, in the training phase, we utilise augmented images as input for the DNN to extract visual features from different DenseNet models. It needs to be clarified that there are no set preliminary standards for determining when to freeze the layer or adjust the learning weight in DenseNet201. Therefore, to retrain DenseNet201 with different depths, some configurations are suggested by freezing different layers of DenseNet201, expressed as DenseNet201:A, DenseNet201:B, DenseNet201:C, and DenseNet201:D. The average accuracy of the suggested approaches is shown in the result section. Each configuration was run five times using the same training and validation datasets (Group 1 dataset) with different seed. In the configuration, it is indicated that DenseNet201:A retrains all of its layers, but DenseNet201:B retrains the final three dense blocks while freezing the initial block. DenseNet201:C transfers its convolution layer to the Dense Block 2 and its Transitional Layer 2 to the Dense Block 3 and its Dense Block 4 directly with pre-trained weights and they are updated with a learning rate as 10^{-4}. In DenseNet:D, Transitional Layer 3 with the last one Dense Block is retrained. The network consists of a single fully connected layer with 256 neurons which take the extracted visual features as the input and produce the output as a prediction vector. A 30%

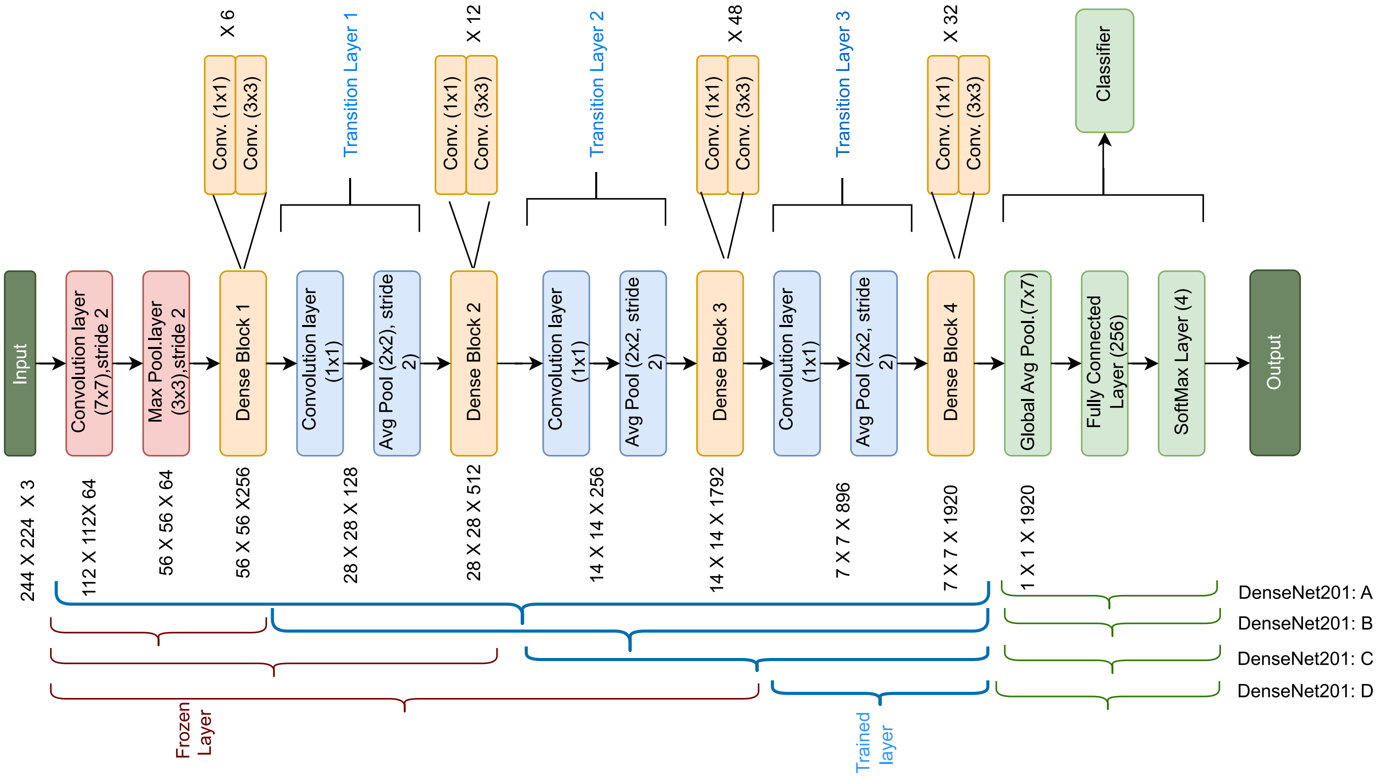


Fig. 6. Proposed model architecture for trading card corner defect classification.

dropout rate is added to the model to stop overfitting. Table 4 presents the specific outcomes in terms of accuracy, F1-score, recall, precision, true negative rate (TNR), false positive rate (FPR) and false negative rate (FNR). The observed trend suggests that retraining denser blocks correlates with higher accuracy.

3.6. Calibration

In real-time applications, it is more important to ensure the reliable prediction of deep neural networks. Despite remarkable advancements in performance across many works, modern deep neural networks often exhibit poor calibration in terms of output confidence. Aligning the predicted probabilities of a DNN model with the true probabilities of the events is known as calibration. Typically in any classification model, a set of raw scores or logits is found as the output of a DNN model. These scores represent the model's confidence in assigning a sample to each class and then transform into probabilities using a softmax function. However, these uncalibrated raw probabilities do not always adequately reflect the actual possibility of the occurrences. For example, a sample may not actually have an 80% chance of belonging to Class A even though the projected probability is 0.8 for Class A. In order to improve the model's projected probabilities and make them more accurate representations of the true probability, a calibration technique for DNNs is introduced. The choice of calibration method depends on the specific characteristics of the DNN model and the dataset. Here, three calibration techniques are used, TS, MS and VS (Guo et al., 2017; Platt, 2000).

3.6.1. Temperature scaling

TS (Guo et al., 2017) is a post-processing calibration method which applies a single parameter, t, to the logits of a classification model and then uses the softmax layer function to obtain calibrated probabilities. The value of temperature t is minimised by reducing the value of negative log-likelihood (NLL) function on the validation set, guaranteeing that the calibrated probabilities accurately represent the likelihood of each class. It also adjusts the confidence levels of the model's predictions. The ideal value of t is determined through experimentation on a validation set.

3.6.2. Matrix and vector scaling

The process of matrix scaling (Platt, 2000) entails transforming the logits linearly, as in $S_y = \hat{y}_i(x, W, b) = \max \Sigma(W.h_i + b)(k)$, where the logit layer is the input of the softmax function. On the validation set, the parameters W (size $K{\times}K$) and b (size K) are optimised with respect to the NLL. However, vector scaling, where W (size $K{\times}K$) is a diagonal matrix, is a more relaxed variant form of matrix scaling.

3.7. Measures for calibration

Two commonly utilised measurement metrics in assessing calibrated models are NLL and Expected Calibration Error (ECE).

3.7.1. NLL

To calibrate a neural network, it is important to assess the proximity of the network's softmax layer output to the true probability $Q(y|x)$ (Hastie et al., 2009). The calibration can be quantified by measuring the similarity between $S_y(x)$ and $Q(y|x)$ functions. Since only samples from the exact distribution $Q(y|x)$ are usually available (e.g., from a validation set), calibration can be evaluated using Gibbs inequality, as shown in Eq. (1).

$$-E_{Q(x,y)}[\log(Q(y|x))] \le -E_{Q(x,y)}[\log(P(y|x))] \tag{1}$$

where E represents the expected value. The minimum value of $-E_{Q(x,y)}[\log(P(y|x))]$ happens when $P(y|x)$ is similar to the true conditional distribution, $Q(y|x)$. This disparity is applicable for any distribution function, $P(y|x)$. The NLL is defined as the empirical estimation of $-E_{Q(x,y)}[\log(P(y|x))]$ and can be reformulated in deep neural networks as follows.

$$NLL = -\sum_{(x_i,y_i)\sim Q(x,y)} \log(S_{y_i}(x_i)) \tag{2}$$

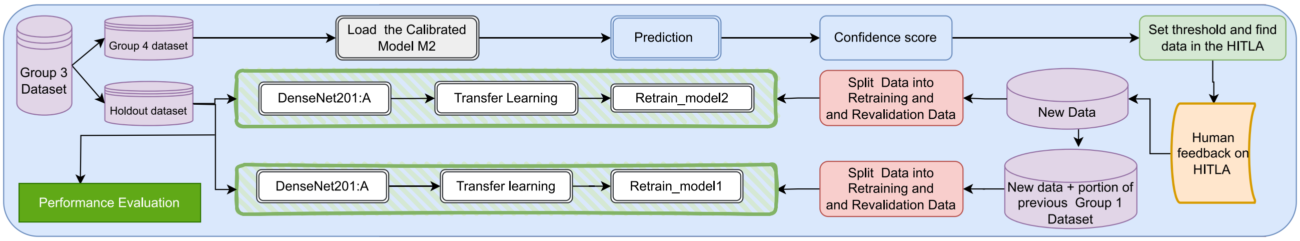

Group 3 Dataset
Group 4 dataset
Holdout dataset
Load the Calibrated Model M2
Prediction
Confidence score
Set threshold and find data in the HITLA
DenseNet201:A
Transfer Learning
Retrain_model2
Split Data into Retraining and and Revalidation Data
New Data
Human feedback on HITLA
Performance Evaluation
DenseNet201:A
Transfer learning
Retrain_model1
Split Data into Retraining and and Revalidation Data
New data + portion of previous Group 1 Dataset

Fig. 7. Workflow for the proposed Human-in-the-loop Analytics (HITLA) and model retraining.

where (x_i, y_i) are samples coming from the true conditional distribution $Q(x, y)$. The NLL is used as a measure of calibration, quantifying the similarity between the true conditional distribution $Q(y|x)$, and probability function $S_y(x)$ with a smaller NLL which indicate better calibration.

3.7.2. ECE

A possible definition of calibration is the relation between confidence and accuracy (Pakdaman Naeini et al., 2015). The dissimilarity between the confidence assigned by a classification model and the real probability of correctly classifying a given sample is called miscalibration. To perform the calculation, the confidence range $[0, 1]$ is divided into L equally spaced confidence bins. Samples are then assigned to each bin B_l according to their confidence range. Next, for each subset B_l, it calculates the weighted absolute difference between accuracy and confidence using the following formula.

$$ECE = \sum_{l=1}^{L} \frac{|B_l|}{N} |acc(B_l) - conf(B_l)| \tag{3}$$

Here, N is the total number of samples.

3.8. Rule-based method

A rule-based method is proposed to further improve the accuracy of the model's predictions based on the confidence scores obtained through the calibration process. The different rule-based methods applied in this research are as follows:

3.8.1. Threshold-based method

After obtaining the calibrated scores or probabilities for a test image, a distinct threshold τ is applied to these probabilities to predict an appropriate class. For instance, a test image is identified as belonging to Class A if the related calibrated probability is greater than τ. If two or more probabilities are greater than τ or all the probabilities are less than τ, we classify the image differently (e.g., via human) or leave the sample unclassified.

3.8.2. Distance-based method

The distance Δ between the top two confidence scores (TTCS) (P_1 and P_2, respectively, where $P_1 \geq P_2$ and $\Delta = P_1 - P_2$) indicates the degree of uncertainty in the model's prediction. For a given test sample, it expresses the highest degree of confidence, if the model has in its top prediction relative to the next most likely prediction. A larger gap implies more confidence in the model's prediction, while a smaller gap signifies an uncertainty. So, we define a threshold ρ for Δ to make a judgment in the context of this rule-based procedure. If $\Delta > \rho$, the test sample belongs to the class related the highest probability P_1. Otherwise, we take further action such as seeking human review.

3.8.3. Human-in-the-loop and special case handling

Calibrated probabilities act as a measure to assess the level of uncertainty in the model's predictions. When a sample receives low confidence scores, a flag is raised, prompting the system to route these instances for human review. Alternatively, the decision making process can be delegated to an expert system, ensuring careful consideration and validation of predictions with lower confidence, thus enhancing the overall reliability of the model's outputs. This approach helps mitigate potential errors and ensures more accurate and trustworthy predictions in situations of uncertainty.

Table 2
Experimental environment of our approach.

Environment	Configuration parameters
GPU	Mac M1 Pro 14 cores
CPU	Mac M1 Pro 8 cores
Python Platform	macOS-12.4-arm64-arm-64 bit
Deep learning framework	Tensorflow 2.9.2, Keras 2.9.0
Programming language	Python 3.9

Fig. 7 illustrates the workflow for model retraining and evaluation after human review. Firstly, we load our calibrated model M2, and then split Group 3 dataset into two: Group 4 and holdout datasets and predict the confidence score on Group 4 dataset. Then, we apply threshold-based and distance-based methods to the predictions and set thresholds to identify data in the HITLA. After that, we again split the data that passed through the HITLA into retraining and revalidation datasets and apply transfer learning using the weights from DenseNet201:A (model M1) to retrain the model, Retrain_model2, using the retraining dataset. Furthermore, for experimental purposes, we create a new dataset by combining new data from HITLA with a portion of the previous Group 1 dataset, split it into retraining and revalidation datasets, and then retrain the model, Retrain_model1. Finally, we test both models, Retrain_model1 and Retrain_model2, on the holdout dataset shown in Section 4.

4. Experiments

The entire experiments are conducted in four parts. The first part involves analysing the performance of the DenseNet model, the second part compares the results of different post-calibration methods, the third part presents the classification using rule-based method and the fourth part analyses the results, specifically examining the extent of human-in-the-loop involvement and the performance of the retrained model. The experimental environment and parameter settings are provided in Table 2.

4.1. Benchmark models

This research focuses on analysing the structure of pre-trained models and fine-tuning them to achieve the best results. Other two architectures of different depths, namely DenseNet121, and DenseNet169 are also tested to evaluate the best model performance with the same configuration. The three benchmark models, DenseNet121, DenseNet169, and DenseNet201, exhibit the same structural architecture illustrated in Fig. 6, with a small variation residing in the number of 1×1 and 3×3 layers. A summary of the benchmark models with different setting is provided in Table 3.

Table 3
Summary of the different settings of the benchmark models.

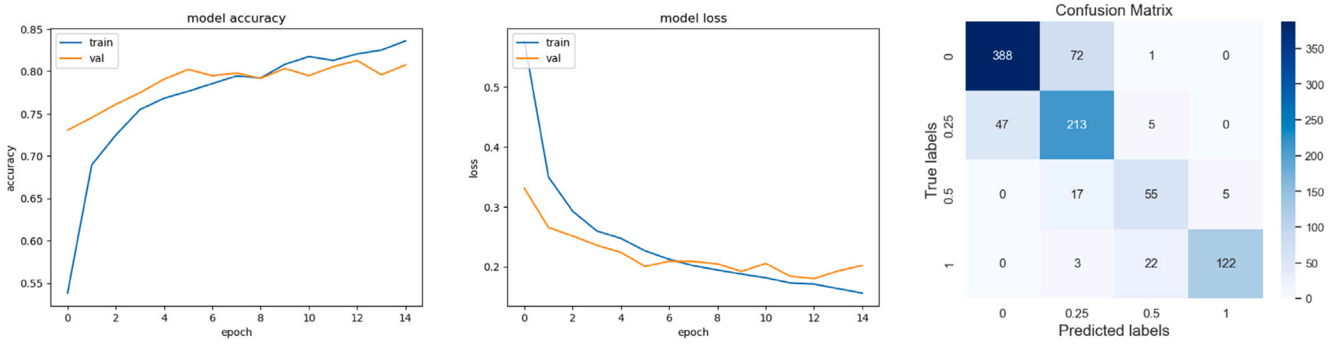
Learning setting	Frozen layer	Trained layer	New layer
DenseNet201:A	Frozen Layer = 0	All Layer	FCL, Softmax
DenseNet201:B	CP, DB1	TL1, DB2, TL2, DB3, TL3, DB4	FCL, Softmax
DenseNet201:C	CP, DB1, TL1, DB2	TL2, DB3, TL3, DB4	FCL, Softmax
DenseNet201:D	CP, DB1, TL1, DB2, TL2, DB3	TL3, DB4	FCL, Softmax

Note: CP = First block of convolution layer and pooling layer, TL = Transition Layer, DB = Dense Block, FCL = Fully connected layer.

Table 4
Performance comparison of different classification models of DenseNet for corner classification.

Model	Model architecture	Precision (%)	Recall (%)	TNR (%)	F1-score (%)	FPR (%)	FNR (%)	Accuracy (%)
DenseNet201	**DenseNet201:A**	**83**	**83**	**91**	**83**	**9**	**17**	**83**
	DenseNet201:B	81	79	90	79	10	21	79
	DenseNet201:C	79	78	86	78	14	22	78
	DenseNet201:D	68	70	81	67	19	30	70
DenseNet169	DenseNet169:A	81	80	89	80	11	20	80
	DenseNet169:B	81	79	86	78	14	21	79
	DenseNet169:C	79	79	86	77	14	21	79
	DenseNet169:D	77	77	85	75	15	23	77
DenseNet121	DenseNet121:A	81	81	90	81	10	19	81
	DenseNet121:B	81	80	89	80	11	20	81
	DenseNet121:C	80	79	86	79	14	21	79
	DenseNet121:D	75	76	86	75	14	24	76

Note: TNR = True negative rate, FPR = False positive rate, FNR = False negative rate.



(a) (b) (c)

Fig. 8. Performance of DenseNet201:A: (a) Accuracy, (b) Model loss and (c) Confusion matrix.

In Table 4 the comparison results across four different settings of DenseNet121, DenseNet169, and DenseNet201 are outlined. The goal is to identify the best configuration associated with accuracy. Two key and significant observations are: (i) DenseNet201 demonstrates superior performance compared to DenseNet121 and DenseNet169, achieving an accuracy of 83%. It also exhibits a True Negative Rate (TNR) of 91% and an F1-score of 83%. Notably, it maintains a False Positive Rate (FPR) of 9% and a False Negative Rate (FNR) of 17%. DenseNet201's deeper neural structure allows it to extract more complex features, maybe the source of this increase. (ii) DenseNet201:A exhibits better results compared to others DenseNet201:B, DenseNet201:C, and DenseNet201:D in terms of precision, recall and F1-score and accuracy. Due to fixed parameter values, DenseNet201:B, DenseNet201:C, and also DenseNet201:D were unable to learn certain features from image data, which may have contributed to their low performance when compared to the fine-tuned model DenseNet201.

Though in this experiment precision, recall, FPR, and FNR are assessed, accuracy takes precedence in industrial contexts due to its pivotal role in maintaining customer satisfaction and ensuring product quality. Unlike human graders who may accept minor discrepancies, for example, as grading a card as an 8 when predicted as a 7.5, machine learning models are expected to consistently achieve high accuracy levels. While precision and recall provide detailed insights, accuracy is the primary measure of whether the model meets industry standards. Therefore, aligning evaluation metrics with industry specific needs ensures models consistently meet accuracy expectations for operational success. Additionally, Fig. 8 illustrates the accuracy and loss curve for initial training T1 and validation dataset (Group 1) using DenseNet201:A. It is shown that the validation accuracy peaks at 83% in the 15th epoch and subsequently stabilises. Furthermore, Fig. 8(c) depicts the confusion matrix for DenseNet201:A, where the majority of the test samples are correctly classified, particularly Class 0, Class 0.25, Class 0.5, and Class 1. Moreover, to solve the class imbalance problem, focal loss has been used which provides less error and diminishes to less than 0.2.

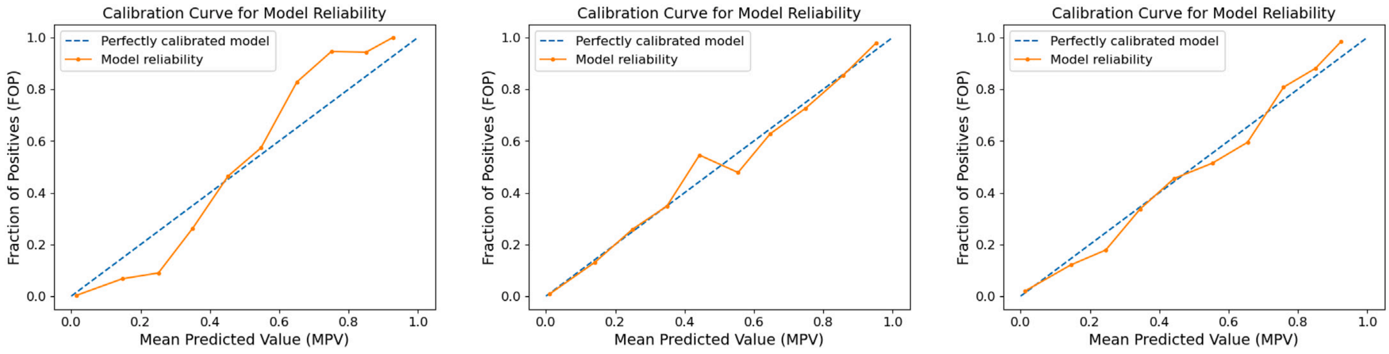
4.2. Calibration performance

This part focuses on calibrating the model to improve the reliability of a decision making application and explore different measure-based post-processing techniques. Group 2 dataset is split into two parts: calibration (60%) and Group 3 (40%). From Group 3 dataset 80% (i.e., Group 4 dataset) is used for testing purpose to evaluate calibration performance, while the remaining 20% is kept as a holdout dataset used later in the HITLA. For the TS method, hyperparameters are fine-tuned on the calibration dataset based on the returned temperature (t) to reduce the NLL. Search interval and step size for t depend on the number of classes in the dataset and the model's accuracy.

Table 5
Performance comparison of different measure-based calibration methods for different models for corner classification.

Model	Uncalibrated				Temperature scaling				Matrix scaling				Vector scaling			
	ACC.	NLL	ECE	RE	ACC.	NLL	ECE	RE	ACC.	NLL	ECE	RE	ACC.	NLL	ECE	RE
DenseNet201	**83%**	0.48	12%	0.23	**83%**	**0.35**	**8%**	0.15	82%	0.42	**8%**	**0.13**	**83%**	0.42	**8%**	**0.13**
DenseNet169	82%	0.44	16%	0.29	82%	0.4	8%	0.15	82%	0.36	8%	0.15	**83%**	**0.34**	**7%**	**0.13**
DenseNet121	81%	0.47	16%	0.23	82%	0.4	**11%**	0.18	82%	0.41	**11%**	**0.16**	**83%**	**0.39**	**11%**	0.17
ResNet152	**81%**	0.61	16%	0.21	**81%**	0.48	**11%**	**0.15**	79%	**0.46**	12%	0.18	80%	0.45	**11%**	0.17
ResNet101	76%	0.63	15%	0.19	76%	0.56	**14%**	**0.25**	76%	0.55	**14%**	0.26	77%	**0.54**	**14%**	0.27
ResNet50	73%	0.69	16%	0.2	73%	0.63	14%	0.25	**75%**	**0.62**	**8%**	**0.15**	75%	**0.62**	13%	0.23
VGG16	78%	0.49	13%	0.17	78%	0.46	12%	0.2	**79%**	0.47	12%	0.21	78%	0.45	**9%**	0.16
VGG19	78%	0.5	19%	0.3	**79%**	0.47	**17%**	0.29	78%	0.47	18%	**0.24**	78%	**0.46**	18%	0.28
InceptionV3	74%	0.5	17%	0.27	**76%**	**0.46**	**11%**	0.22	75%	0.48	12%	0.23	75%	**0.46**	12%	0.23

Note: ACC. = Accuracy, NLL = Negative log likelihood, ECE = Expected calibration error and RE = Root mean square error.



(a) (b) (c)

Fig. 9. (a) Model reliability curve for DenseNet201:A : (a) Uncalibrated (b) Calibrated with temperature scaling, (c) Calibrated with matrix and vector scaling.

The experimental analysis is reported based on three calibration metrics: Root Mean Square Error (RMSE), ECE, and NLL, as previously discussed in Section 3.5. In Table 5, a comparison between TS, MS and VS are performed. TS uses a single parameter to fine-tune the softmax layer output whereas the other two methods apply linear functions to the logit layer. Here, the bin size is 10 for all experiments. Among various DenseNet variations, DenseNet201:A yields superior results. Consequently, we apply calibration to other models configuration DenseNet201:A, DenseNet169:A and DenseNet121:A. From the experimental result shown in Table 5, it is revealed that TS outperforms others in calibrating the network, especially InceptionV3 and ResNet152. For DenseNet201, DenseNet169, DenseNet121, VGG16 and VGG19, VS provides better results and the calibration error improvement is more significant. For DenseNet201, the accuracy remains consistent at 83% both before and after calibration. However, the ECE shows a significant improvement, decreasing from 12% before calibration to 8% after calibration. Additionally, the NLL reduces notably from 0.48 to 0.35 following TS calibration. Fig. 9(a) shows the result of the model before calibration for DenseNet201:A and Fig. 9(b) shows curve after calibration using TS while Fig. 9(c) for matrix and vector scaling. It is shown VS and MS curve is clearly in close alignment with the perfectly calibrated curve in this instance.

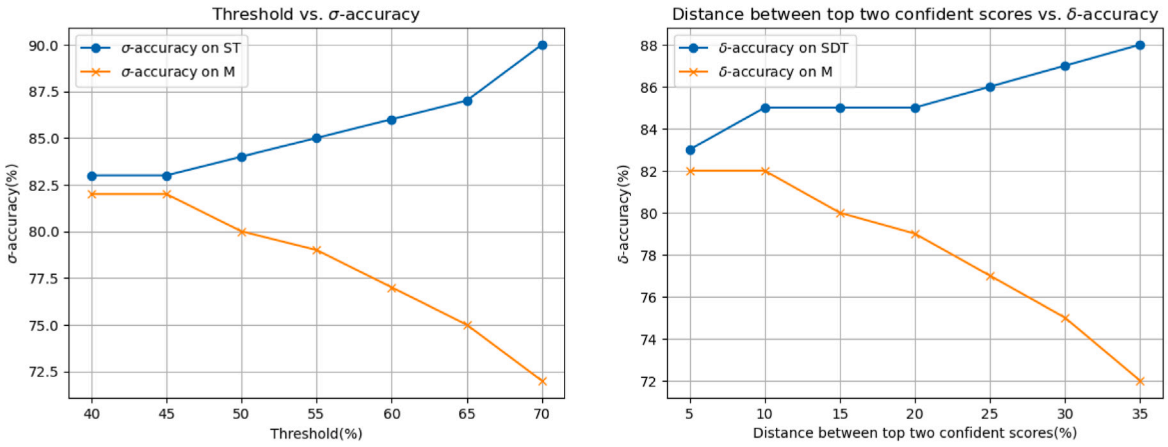
4.3. Rule-based classification

The output of the rule-based method after calibrating the model is analysed. By feeding the calibrated output to the rule-based method, the system can make more accurate decisions and leverage the model's uncertainties to achieve higher accuracy. Here, for this section the term "test dataset" refers to Group 4 dataset.

In our experiment, the accuracy of each rule-based method is measured using the number of TP classifications (Θ) in two aspects: with respect to the size (M) of the test dataset (Group 4 dataset) and the number of test samples fall above the set threshold. For the threshold-based method, its accuracy, namely σ-accuracy, is first estimated by dividing Θ with M. Also, we estimate the σ-accuracy by dividing Θ with the number of test samples (ST) above threshold τ. Fig. 10(a) shows these two estimations for different τ in orange and blue coloured graphs, respectively. The increase of τ above 50% tends to decrease σ-accuracy with respect to the entire test dataset (Group 4 dataset) shown in orange graph, but it tends to increase σ-accuracy with respect to the test sample above τ (blue graph). This means at the higher threshold the proposed system becomes more confident for the truly classified samples.

Similarly, for the distance-based method, its accuracy, namely δ-accuracy, is first estimated by dividing Θ with M. In addition, we estimate the δ-accuracy by dividing Θ with the number of test samples (SDT) above the distance threshold ρ. Fig. 10(b) shows the same graphs: orange graph for the entire test dataset and blue graph for samples above ρ. The increase of ρ above 5% tends to decrease δ-accuracy with respect to the entire test dataset (orange graph), but it tends to increase δ-accuracy with respect to the test sample above ρ (blue graph).

Moreover, Tables 6 and 7 provide more details about the experimental results. For instance, in Table 6, it is shown that $\tau = 0.7$ yields 90% σ-accuracy for samples above τ and 70% across the entire testing dataset. Gradually decreasing the threshold (τ) decreases subset accuracy while increasing accuracy for the total dataset until $\tau = 0.4$. At $\tau = 0.35$ the model encounters situations involving multi-class recognition. In this work, the selection of the class with the highest probability takes precedence, rendering thresholds such as 0.35 irrelevant for decision-making. Similarly, Table 7 shows that using the gap $\rho = 0.05$ results in 251 out of 296 samples correctly identified. The δ-accuracy is 84% for test samples above ρ (296) but 82% across the entire testing dataset (304). At $\rho = 0.35$ improves separability and confidence, yielding 88% δ-accuracy within the subset but dropping to



(a) (b)

Fig. 10. (a) σ-accuracy and δ-accuracy curve for DenseNet201:A: (a) threshold-based method, (b) distance-based method. Note: M = Size of the test dataset (Group 4 dataset), ST = Sample above threshold (τ) in threshold-based method, SDT = Sample above threshold (ρ) in distance-based method.

Table 6
σ-accuracy of threshold-based method for the DenseNet201:A.

M = Size of the test dataset	Threshold (τ)	Class	ST = Sample above τ	True Positive (Θ)	σ-accuracy on ST	σ-accuracy on M
304	0.7	One	235	213	0.90	0.70
	0.65	One	259	226	0.87	0.74
	0.6	One	281	242	0.86	0.79
	0.55	One	291	249	0.85	0.81
	0.5	One	303	254	0.83	0.83
	0.45	One	304	255	0.83	0.83
	0.4	One	304	255	0.83	0.83
	0.35	NA	NA	NA	NA	NA

Table 7
δ-accuracy of distance-based method for the DenseNet201:A.

M = Size of the test dataset	Gap in TTCS (ρ)	SDT = Sample above ρ	True Positive (Θ)	δ-accuracy on SDT	δ-accuracy on M
304	0.05	296	251	0.84	0.82
	0.1	292	250	0.85	0.82
	0.15	285	244	0.85	0.80
	0.2	283	243	0.85	0.79
	0.25	274	236	0.86	0.77
	0.3	262	228	0.87	0.75
	0.35	248	219	0.88	0.72
	0.4	NA	NA	NA	NA

Note: TTCS: Top two confident score. NA: Not Applicable.

72% across the entire testing dataset (M). Threshold, such as 0.4 lack sufficient data for analysis.

The results demonstrate that how threshold selection affects model accuracy and confidence. Therefore, for a high-confidence system in the industrial perspective, the threshold should be set higher to ensure greater confidence in the system's decisions.

4.4. HITLA: Retrained and base model comparison

This section focuses on HITLA to identify the number of data points requiring human intervention. Following this identification, we proceed to retrain the model using the feedback obtained for those data points that fall below the specified thresholds. Table 8 presents a comparative analysis of data requiring human intervention based on varying thresholds using both threshold-based and distance-based methods. In the threshold-based method, starting with a threshold of 0.7, 18% of data are flagged for human feedback and gradually decreasing to 0% at τ = 0.4 and below. Similarly, in the distance-based method, a distance of 0.35 between the top two confident scores resulted in 18% of data points necessitating human feedback. The percentages of data are declining as the distance are decreased. This table shows how threshold adjustments in classification models impact human judgement in HITLA.

After receiving human feedback, we experimentally retrain the model. Table 9 identifies significant data flagged for HITLA intervention at thresholds 0.7 and 0.65, and in the distance-based method at distances such as 0.35 and 0.30. These findings lead to human review and model retraining, as illustrated in Fig. 7. Notably, at a threshold of 0.7 in the threshold-based method, Retrain_model1 achieves 84%

Table 8
Data passed in HITLA by varying thresholds.

Threshold-based method		Distance-based method	
Threshold (τ)	HITLA (%)	Gap in TTCS (ρ)	HITLA (%)
0.7	**18**	**0.35**	**18**
0.65	11	0.3	13
0.6	6	0.25	9
0.55	3	0.20	6
0.5	1	0.15	6
0.45	0	0.10	3
0.4	0	0.05	2

Note: TTCS = Top two confident score, HITLA = Human-in-the-loop Analytics.

Table 9
Performance comparison between base and retrained models.

	Thres-holds	Data in the HITLA (%)	Model	Precision (%)	Recall (%)	F1-score (%)	TNR (%)	FPR (%)	FNR (%)	ACC. (%)
Threshold-based Method	0.7	18	**Retrain_model1**	**84**	**84**	**84**	**90**	**10**	**16**	**84**
			Retrain_model2	82	82	82	89	11	17	82
	0.65	11	Retrain_model1	81	82	82	91	9	18	82
			Retrain_model2	81	81	80	87	13	19	81
Distance-based Method	0.35	18	**Retrain_model1**	**84**	**84**	**84**	**90**	**10**	**17**	**84**
			Retrain_model2	83	83	82	90	10	17	83
	0.3	13	**Retrain_model1**	**85**	**85**	**85**	**90**	**10**	**15**	**85**
			Retrain_model2	81	81	80	87	13	19	81
Model M1			DenseNet201:A	82	82	82	91	9	17	82

Note: TNR = True Negative Rate, FPR = False Positive Rate, FNR = False Negative Rate, ACC. = Accuracy, HITLA = Human-in-the-loop Analytics.

accuracy, while Retrain_model2 attains 82%, same as the base model's performance. Similarly, at a threshold of 0.65, 11% of data are flagged, resulting in a slight decrease to 81% accuracy for Retrain_model1 compared to the base model. In the distance-based method, where 18% of data requires intervention at a distance of 0.35, Retrain_model1 achieves 84% accuracy, whereas Retrain_model2 maintains 83%. At a distance of 0.30, 13% of data go to HITLA, resulting in Retrain_model1 achieving 85% accuracy and Retrain_model2 achieving 81%, just 1% below the base model's performance. This analysis demonstrates that model retraining based on the proportion of HITLA affects accuracy, allowing for improved decision-making when new data is introduced. This result shows that the retrained model's accuracy remains close to the original model for the small dataset, indicating that collecting significant data from the industry for retraining can yield favourable outcomes.

Notably, in this study, we evaluate our model by retraining it when 10% of the data passes through HITLA process. The process involves iterative data refinement through HITLA, enabling us to fine-tune our model based on feedback. As retraining models from scratch poses significant computational challenges and considerable time, we employ transfer learning, leveraging weights from our previously trained model to retrain the model. This strategy is crucial for ensuring the ongoing effectiveness of our approach in trading card grading within industrial applications. As this is an ongoing research topic (Wilchek et al., 2023; Wu et al., 2022; Hoi et al., 2021; Mosqueira-Rey et al., 2023), it focuses on understanding when and how the model should undergo continuous retraining in industrial settings.

4.5. *Discussion and future work*

This research presents an automated transfer learning-based approach for trading card grading, accompanied by various post-processing calibration techniques to address DNN overconfidence issues. Additionally, human assistance is integrated to evaluate low-confidence cases. DenseNet serves as the benchmark model within the transfer learning framework for corner defect classification and optimisation. Our experimental results demonstrate the effectiveness of the DenseNet:A model compared to other models. Moreover, calibration of DNN model and various rule-based increased the confidence in its classifications. Through experimentation, we observe how these methods impact the accuracy of the model. However, this research is limited by its utilisation of a small dataset, which can be addressed by employing various data augmentation techniques (Mumuni and Mumuni, 2022). Another drawback is the inclusion of human inspection for low-confidence cases introduces which is still time consuming and it could potentially be mitigated by retraining the model with human feedback (Caballero-Ramirez et al., 2023; Rožanec et al., 2024).

Furthermore, efforts will be directed towards enhancing the performance of corner defect classification. One strategy involves establishing an ensemble DenseNet (Wang et al., 2023) by combining multiple transfer learning settings. Future studies could also explore the generation of defect localisation information (Peng et al., 2024), which would assist graders in making more accurate classifications. Note: If the paper is accepted for publication, the code and other materials will be available on GitHub for future research and enhancement.

5. Conclusion

This paper investigates different DNN models in the card corner grading and shows the impact of calibration on refining decision making. While deep neural networks have shown substantial improvements in accuracy, they often suffer from overconfidence. To tackle this issue, post-processing calibration methods are implemented. This approach, requiring no retraining of the network, proves versatile for calibrating pre-trained models in various practical scenarios.

In the card grading context, the final confidence score is pivotal in defect classification and detection with other samples referred for human inspection. An ideal detection system should convey awareness of the decision border, certainty in incorrect classifications and uncertainty in misclassifications. The suggested DenseNet201 model exhibited overconfidence when it is trained on the dataset (initial training dataset T1 in Fig. 5). To remedy this, Group 2 dataset is divided into 60% for the calibration dataset, where three calibration methods are applied and 40% for the Group 3 dataset. From Group 3 dataset 80% of the data is used as testing data to evaluate the performance of the calibrated model. The experimental results which are presented in Table 5 show notable improvements in calibration, which raise the referral system's effectiveness. Samples that have confidence levels higher than a threshold set by a rule-based system are considered correctly classified in the referral system. To ensure a more thorough investigation instances below the confidence level are sent for human scrutiny. In addition, this calibration process not only addresses the overconfidence of the model but also establishes a reliable and effective framework for decision making in card corner grading.

CRediT authorship contribution statement

Lutfun Nahar: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Md. Saiful Islam:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization. **Mohammad Awrangjeb:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Funding acquisition. **Rob Verhoeve:** Validation, Supervision, Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Lutfun Nahar reports financial support and equipment, drugs, or supplies were provided by Media8 Pty. Ltd. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

References

Bai, D., Li, G., Jiang, D., Yun, J., Tao, B., Jiang, G., Sun, Y., Ju, Z., 2024. Surface defect detection methods for industrial products with imbalanced samples: A review of progress in the 2020s. Eng. Appl. Artif. Intell. 130, 107697.

Banús, N., Boada, I., Xiberta, P., Toldrà, P., Bustins, N., 2021. Deep learning for the quality control of thermoforming food packages. Sci. Rep. 11 (1), 21887.

Bernardo, J.M., Smith, A., 2009. Bayesian Theory, vol. 405, John Wiley & Sons.

Bojarski, M., Testa, D.D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., Zieba, K., 2016. End to end learning for self-driving cars.

Caballero-Ramirez, D., Baez-Lopez, Y., Limon-Romero, J., Tortorella, G., Tlapa, D., 2023. An assessment of human inspection and deep learning for defect identification in floral wreaths. Horticulturae 9 (11).

Chakraborty, S., Moore, M., Parrillo-Chapman, L., 2022. Automatic defect detection for fabric printing using a deep convolutional neural network. Int. J. Fashion Des. Technol. Educ. 15 (2), 142–157.

Chauhan, T., Palivela, H., Tiwari, S., 2021. Optimization and fine-tuning of DenseNet model for classification of COVID-19 cases in medical imaging. Int. J. Inf. Manage. Data Insights 1 (2), 100020.

Collectible Madness, Collectible card. https://collectiblemadness.com.au (Last accessed on 4 September, 2024).

Dai, W., Li, D., Tang, D., Jiang, Q., Wang, D., Wang, H., Peng, Y., 2021. Deep learning assisted vision inspection of resistance spot welds. J. Manuf. Process. 62, 262–274.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. In: CVPR. pp. 248–255.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T., 2013. DeCAF: A deep convolutional activation feature for generic visual recognition. arXiv preprint 32.

Dong, Y.-Y., Huang, Y.-S., Xu, B.-L., Li, B.-C., Guo, B., 2022. Bruise detection and classification in jujube using thermal imaging and DenseNet. J. Food Process Eng. 45 (3), e13981.

Dong, X., Zhang, C., Wang, J., Chen, Y., Wang, D., 2024. Real-time detection of surface cracking defects for large-sized stamped parts. Comput. Ind. 159–160, 104105.

Grading, S., Grading standard and policy. https://www.psacard.com/resources/gradingstandards.

Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q., 2017. On calibration of modern neural networks. In: Proceedings of the 34th ICML. In: Proceedings of ML Research, vol. 70, PMLR, pp. 1321–1330.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, second ed. In: Springer Series in Statistics.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. CVPR 770–778.

Hoi, S.C., Sahoo, D., Lu, J., Zhao, P., 2021. Online learning: A comprehensive survey. Neurocomputing 459, 249–289. http://dx.doi.org/10.1016/j.neucom.2021.04.112.

Huang, G., Liu, Z., Weinberger, K.Q., 2016. Densely connected convolutional networks. CVPR 2261–2269.

Jha, S.B., Babiceanu, R.F., 2023. Deep CNN-based visual defect detection: Survey of current literature. Comput. Ind. 148, 103911.

Jiang, X., Osl, M., Kim, J., Ohno-Machado, L., 2011. Calibrating predictive model estimates to support personalized medicine. J. Am. Med. Inform. Assoc.: JAMIA 19, 263–274.

Kim, J., Ko, J., Choi, H., Kim, H., 2021. Printed circuit board defect detection using deep learning via a skip-connected convolutional autoencoder. Sensors 21 (15).

Konovalenko, I., Maruschak, P., Brezinová, J., Prentkovskis, O., Brezina, J., 2022. Research of U-net-based CNN architectures for metal surface defect detection. Machines 10 (5), 327.

Kou, X., Liu, S., Cheng, K., Qian, Y., 2021. Development of a YOLO-V3-based model for detecting defects on steel strip surface. Measurement 182, 109454.

Lei, Y., Zuo, M.J., 2009. Gear crack level identification based on weighted k nearest neighbor classification algorithm. Mech. Syst. Signal Process. 23 (5), 1535–1547.

Lian, J., Jia, W., Zareapoor, M., Zheng, Y., Luo, R., Jain, D.K., Kumar, N., 2020. Deep-learning-based small surface defect detection via an exaggerated local variation-based generative adversarial network. IEEE Trans. Ind. Inform. 16 (2), 1343–1351. http://dx.doi.org/10.1109/TII.2019.2945403.

Liang, F., Zhao, L., Ren, Y., Wang, S., To, S., Abbas, Z., Islam, M.S., 2024. LAD-net: A lightweight welding defect surface non-destructive detection algorithm based on the attention mechanism. Comput. Ind. 161, 104109.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. In: ICCV. pp. 2980–2988.

Lin, H., Li, B., Wang, X., Shu, Y., Niu, S., 2019. Automated defect inspection of LED chip using deep convolutional neural network. J. Intell. Manuf. 30 (6), 2525–2534.

Liu, Q., Wang, C., Li, Y., Gao, M., Li, J., 2022. A fabric defect detection method based on deep learning. IEEE Access 10, 4284–4296.

Liu, Z., Zhang, C., Li, C., Ding, S., Dong, Y., Huang, Y., 2019. Fabric defect recognition using optimized neural networks. J. Eng. Fibers Fabrics 14, 1558925019897396.

Lu, T., Han, B., Chen, L., Yu, F., Xue, C., 2021. A generic intelligent tomato classification system for practical applications using DenseNet-201 with transfer learning. Sci. Rep. 11 (1), 15824.

Matić, T., Vidović, I., Hocenski, Ž., 2013. Real time contour based ceramic tile edge and corner defects detection. Tech. Gazette 20 (6), 1063–1070.

Media8, Media8. https://psgrading.net.

Meng, X., Luo, S., Wang, W., Zhu, M., 2022. A detection model for corner cracks of continuous casting strand based on deep learning. Ironmak. Steelmak. 49 (10), 1048–1056.

Mosqueira-Rey, E., Hernandez-Pereira, E., Alonso-Rios, D., Bobes-Bascaran, J., Fernandez-Leal, A., 2023. Human-in-the-loop machine learning: a state of the art. Artif. Intell. Rev. 56 (4), 3005–3054.

Mumuni, A., Mumuni, F., 2022. Data augmentation: A comprehensive survey of modern approaches. Array 16, 100258.

Nahar, L., Islam, M.S., Awrangjeb, M., Verhoeve, R., Tuxworth, G., 2023. DeepCorner-Net: A deep learning approach for automated corner grading in trading cards. In: International Conference on Digital Image Computing: Techniques and Applications. IEEE, pp. 24–31.

Niu, S., Li, B., Wang, X., Lin, H., 2020. Defect image sample generation with GAN for improving defect recognition. IEEE Trans. Autom. Sci. Eng. 17 (3), 1611–1622.

Pakdaman Naeini, M., Cooper, G., Hauskrecht, M., 2015. Obtaining well calibrated probabilities using Bayesian binning. Proc. AAAI Conf. Artif. Intell. 29 (1).

Peng, J., Shao, H., Xiao, Y., Cai, B., Liu, B., 2024. Industrial surface defect detection and localization using multi-scale information focusing and enhancement ganomaly. Expert Syst. Appl. 238, 122361.

Platt, J., 2000. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Adv. Large Margin Classif. 10.

PSA, 2024. Card grading. https://www.psacard.com/services.

Ren, R., Hung, T., Tan, K.C., 2018. A generic deep-learning-based approach for automated surface inspection. IEEE Trans. Cybern. 48 (3), 929–940.

Rolland, V., Farazi, M.R., Conaty, W.C., Cameron, D., Liu, S., Petersson, L., Stiller, W.N., 2022. HairNet: a deep learning model to score leaf hairiness, a key phenotype for cotton fibre yield, value and insect resistance. Plant Methods 18 (1), 8.

Rožanec, J.M., Montini, E., Cutrona, V., Papamartzivanos, D., Klemenčič, T., Fortuna, B., Mladenić, D., Veliou, E., Giannetsos, T., Emmanouilidis, C., 2024. Human in the AI loop via xAI and active learning for visual inspection. In: Soldatos, J. (Ed.), Artificial Intelligence in Manufacturing: Enabling Intelligent, Flexible and Cost-Effective Production Through AI. Springer Nature Switzerland, Cham, pp. 381–406.

Saberironaghi, A., Ren, J., El-Gindy, M., 2023. Defect detection methods for industrial products using deep learning techniques: A review. Algorithms 16 (2).

Singh, R., Yadav, G.C., 2014a. Classifying corner defects from square ceramic tile at production phase. Int. J. Comput. Sci. Trends Technol. (IJCST) 2 (4).

Singh, R., Yadav, G.C., 2014b. Corner defect detection based on inverse trigonometric function using image of square ceramic tiles. Int. J. Eng. Comput. Sci 3 (09), 8047–8055.

Sun, T.-H., Tien, F.-C., Tien, F.-C., Kuo, R.-J., 2016. Automated thermal fuse inspection using machine vision and artificial neural networks. J. Intell. Manuf. 27 (3), 639–651.

Tang, Y., Sun, K., Zhao, D., Lu, Y., Jiang, J., Chen, H., 2022. Industrial defect detection through computer vision: A survey. In: 2022 7th IEEE International Conference on Data Science in Cyberspace. DSC, pp. 605–610.

Tulbure, A.-A., Tulbure, A.-A., Dulf, E.-H., 2022. A review on modern defect detection models using DCNNs – deep convolutional neural networks. J. Adv. Res. 35, 33–48.

Wang, R., Cheung, C.F., Wang, C., Cheng, M.N., 2022. Deep learning characterization of surface defects in the selective laser melting process. Comput. Ind. 140, 103662.

Wang, G., Zhang, Y., Zhang, F., Wu, Z., 2023. An ensemble method with DenseNet and evidential reasoning rule for machinery fault diagnosis under imbalanced condition. Measurement 214, 112806.

Wilchek, M., Hanley, W., Lim, J., Luther, K., Batarseh, F.A., 2023. Human-in-the-loop for computer vision assurance: A survey. Eng. Appl. Artif. Intell. 123, 106376.

Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., He, L., 2022. A survey of human-in-the-loop for machine learning. Future Gener. Comput. Syst. 135, 364–381.

Yang, Y., Yang, R., Pan, L., Ma, J., Zhu, Y., Diao, T., Zhang, L., 2020. A lightweight deep learning algorithm for inspection of laser welding defects on safety vent of power battery. Comput. Ind. 123, 103306.

Yao, J., Li, J., 2022. AYOLOv3-tiny: An improved convolutional neural network architecture for real-time defect detection of PAD light guide plates. Comput. Ind. 136, 103588.

Yi, L., Li, G., Jiang, M., 2016. An end-to-end steel strip surface defects recognition system based on convolutional neural networks. Steel Res. Int. 88.

Yıldız, K., Buldu, A., Demetgul, M., 2016. A thermal-based defect classification method in textile fabrics with K-nearest neighbor algorithm. J. Ind. Text. 45 (5), 780–795.

Zadrozny, B., Elkan, C., 2001. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. ICML 1.

Zadrozny, B., Elkan, C., 2002. Transforming classifier scores into accurate multiclass probability estimates. SIGKDD.

Zhu, Z., Han, G., Jia, G., Shu, L., 2020. Modified densenet for automatic fabric defect detection with edge computing for minimizing latency. IEEE Internet Things J. 7 (10), 9623–9636.

Lutfun Nahar received her B.Sc. (Hons.) and M.S. degree in Computer Science and Engineering from the University of Chittagong, Bangladesh, in 2010 and 2011, respectively. She joined at International Islamic University Chittagong, Bangladesh in 2013 and worked as an assistant professor from 2017 until 2022. Currently, she is pursuing the Ph.D. degree with the deep object quality assessment at the School of Information and Communication Technology, Griffith University, Australia. Her current research interests are in the areas of machine learning, image processing and artificial intelligence.

Md. Saiful Islam (Senior Member, IEEE) is a Senior Lecturer at the School of Information and Physical Sciences, The University of Newcastle, Australia. He has completed his Ph.D. in Computer Science and Software Engineering at the Swinburne University of Technology, Australia, in February 2014. He received his B.Sc. (Hons.) and M.S. degree in Computer Science and Engineering from the University of Dhaka, Bangladesh, in 2005 and 2007, respectively. His current research interests are in the areas of data management, artificial intelligence, big data and security analytics.

Mohammad Awrangjeb is a Senior Lecturer at the School of Information and Communication Technology, Griffith University, Australia. He has completed his Ph.D. in Computer Science at the Monash University, Australia, in 2008. He received his M.S. degree in Computer Science and Engineering from the National University of Singapore (NUS), Singapore, in 2004 and B.Sc. (Engineering) in Computer Science from the Bangladesh University of Engineering and Technology (BUET), in 2001.His research interest includes object extraction and modelling from remote sensing data, image processing and fusion of remote sensing data.

Rob Verhoeve has undertaken the role of General Manager of Platinum Standard Grading, overseeing the card grading business. He brings over 28 years of experience in customer-focused banking roles, with 15 of those years as a Bank Branch Manager. He has a broad range of skills and experience in customer service, people management, financial management, and project management. Additionally, he has served as an umpire for both Australian Rules football and cricket over the past 25 years.