Inspection System for Glass Bottle Defect Classification based on Deep Neural Network

Niphat Claypo[1], Saichon Jaiyen[2], Anantaporn Hanskunatai[3]
Automation, Innovation, Intelligence, and Data Science Research Unit-Department of Computer Science-School of Science, King Mongkut's Institute of Technology Ladkrabang, Chalong Krung 1, Chalong Krung Road, Lat Krabang Sub-District, Lat Krabang District, Bangkok 10520, Thailand[1, 3]
King Mongkut's University of Technology Thonburi, Pracha Uthit Road, Bang Mot Sub-District, Thung Khru District Bangkok 10140, Thailand[2]

***Abstract*—The problem of defects in glass bottles is a significant issue in glass bottle manufacturing. There are various types of defects that can occur, including cracks, scratches, and blisters. Detecting these defects is crucial for ensuring the quality of glass bottle production. The inspection system must be able to accurately detect and automatically determine that the defects in a bottle affect its appearance and functionality. Defective bottles must be identified and removed from the production line to maintain product quality. This paper proposed glass bottle defect classification using Convolutional Neural Network with Long Short-Term Memory (CNNLSTM) and instant base classification. CNNLSTM is used for feature extraction to create a representation of the class data. The instant base classification predicts anomalies based on the similarity of representations of class data. The convolutional layer of the CNNLSTM method incorporates a transfer learning algorithm, using pre-trained models such as ResNet50, AlexNet, MobileNetV3, and VGG16. In this experiment, the results were compared with ResNet50, AlexNet, MobileNetV3, VGG16, ADA, Image threshold, and Edge detection methods. The experimental results demonstrate the effectiveness of the proposed method, achieving high classification accuracies of 77% on the body dataset, 95% on the neck dataset, and an impressive 98% on the rotating dataset.**

***Keywords*—Convolution neural network; glass bottle; defect detection; long shot-term memory; inspection machine**

I. Introduction

Defect detection plays a vital role in glass bottle production, safeguarding product quality, consumer safety, brand reputation, and value. By investing in robust flaw detection systems and processes, manufacturers can maintain impeccable standards, safe, and visually appealing glass bottles to their customers [1,2]. Common types of defects that can occur in glass bottles include: 1) Stones are foreign stone grains embedded in the glass, degrading the quality of the bottle. 2) Tears are deformed breaks or fractures on the surface of the bottle. 3) Blisters are raised or swollen areas caused by uneven cooling during manufacturing, affecting both strength and appearance. 4) Cracks are breaks in the surface that compromise the structural integrity of bottle.

The inspection machine uses various technologies to detect defects, including cameras, lasers, and sensors. The inspection machine uses cameras to capture images of the products, and the software analyzes these images to detect defects. When defects are detected, the software sends a signal to the control system to remove the faulty product from the production line.

As for the problem with traditional inspection machines, they cannot inspect defects in complex areas and require the use of imaging techniques to visualize the defects. Because these machines utilize image processing techniques and require various parameter settings to be adjusted by users for defect detection. It results in the inability to completely remove defective bottles from the production line, undermining confidence in the manufacturing process. The detection method on inspection machine for detecting the defects must be accurate, precise, capable of recognizing defect well, and fast in the learning process of defect patterns.

This research proposes a new method for detecting defects in glass bottles using a deep neural network for extracting deep features and instance-based classification. The method uses CNN combined with LSTM to recognize distinctive features of defects and extract those features. The training process is designed to create a set of CNNLSTM models with fewer training iterations, helping to reduce the training time. The instance-based classification is used to classify the defect in the images. It reduces parameter settings by users for defect detection, resulting in a reduction in user workload.

The rest of this paper is organized as follows: Section II resents a background study of glass bottle defect detection techniques. Section III explains the glass bottle defect dataset and describes the proposed glass bottle defect classification method. Section IV explains the evaluation methods, the experimental setup, and the experimental outcomes; and finally. Section V suggests directions for future works and concludes the paper.

II. Related Works

The existing defect detection techniques mainly focused on the defect in glass bottles. The related methods are as follows. Latina et al. [3] presented defect detection method for detecting defects in glass bottles for the purpose of reusing them and highlights the limitations of the manual inspection in micro, small, and medium enterprises (MSMEs). The research introduced a cost-effective deep learning-based method using the SSD MobileNetV2 model to detect various defects in glass bottles. The method used transfer learning and data augmentation techniques to achieve high accuracy, with up to

98.07% overall system accuracy. Gong et al. [4] presented a machine vision system designed for the automatic online inspection of defects in transparent labels on curved glass bottles. The system used an area-array camera and a custom-made blue dome illumination device to capture high-quality still images by minimizing reflection. To address the challenge of distorted curved geometry, a deformable template matching method was employed for precise defect location. The method also included an adaptive threshold selection strategy that effectively detected small scratches by using global and local threshold values along with a Gaussian fitting algorithm. Additionally, techniques such as skeleton extraction and distance transformation were applied to detect the complete edge contour of Chinese characters with a special font, considering the golden edge printing error. Field tests demonstrated an impressive detection accuracy of 99.5% at a speed of 60 bottles per minute, covering over 60,000 bottles. Vitis et al. [5] proposes an algorithm that achieves high detection accuracy while significantly reducing processing time. By using adaptive thresholding and analyzing luminous intensity variations, the algorithm effectively detects blob and airline defects while mitigating the impact of tube curvature, rotation, and vibration. Comparative evaluations demonstrate an 86% reduction in processing time, a 268% increase in throughput, and improved detection accuracy compared to existing methods. The algorithm also incorporates Region of Interest reduction techniques and a tuning procedure for parameter adjustment during production batch changes. The performance of the algorithm was assessed in a real environment, and it successfully identified misclassified tubes, suggesting its practical applicability. Zhang et al. [6] suggested a machine learning-based acoustic defect detection (LearningADD) system to replace manual inspection. The system used an improved Hilbert-Huang transform (HHT) to extract features from acoustic signals and a Shuffled Frog Leaping Algorithm (SFLA) to select features. Five deployment strategies were compared and optimized to improve real-time performance. The LearningADD system was validated using data from a real-life beverage factory. The F-measure of the system reached 98.48%. The proposed deployment strategies were verified using experiments on private cloud platforms. The Distributed Heavy Edge deployment strategy outperformed other strategies, with a defect detection time of less than 2.061 seconds for 99% of bottles. Zhou et al. [7] proposed a surface defect detection framework, which consisted of three main components. Firstly, a novel localization method called entropy rate superpixel circle detection (ERSCD) was introduced. It combined least-squares circle detection, entropy rate superpixel (ERS), and an improved randomized circle detection to accurately identify the region of interest (ROI) on the bottle bottom. The ROI was then divided into two measurement regions: the central panel region and the annular texture region. For defect detection in the central panel region, a method named frequency-tuned anisotropic diffusion super-pixel segmentation (FTADSP) was proposed. It integrated frequency-tuned salient region detection (FT), anisotropic diffusion, and improved superpixel segmentation to accurately detect defect regions and boundaries. For defect detection in the annular texture region, a strategy called wavelet transform multiscale filtering (WTMF) was proposed. It employed wavelet transform and a multiscale filtering algorithm to reduce texture influence and enhance robustness to localization errors. Zhou et al. [8] presented a new apparatus for real-time bottle bottom inspection. The apparatus used a combination of Hough circle detection and size prior to locating the bottom of the bottle. The region of interest was then divided into three measurement regions: central panel region, annular panel region, and annular texture region. A saliency detection method was used to find defective areas in the central panel region. A multiscale filtering method is used to search for defects in the annular panel region. Template matching was combined with multiscale filtering to detect defects in the annular texture region.

The problem with traditional methods of detecting anomalies on glass bottles using image processing is the difficulty in detecting anomalies, especially when there are color and shape variations on the bottles. Inconsistent lighting conditions also pose challenges in detecting anomalies. The disadvantage of using deep neural networks for detecting counterfeit on glass bottles is the complexity involved in building and training the network. Deep neural network-based methods require a large amount of data and longer training time. Additionally, fine-tuning and selecting appropriate parameters for deep networks can be challenging.

III. Methodology

A. Glass Bottle Dataset

There are two main categories of glass bottle inspection machines: straight glass bottle inspection machines and rotary glass bottle inspection machines. Fig. 1 shows an example of an inspection machine for glass bottles. The straight glass bottle inspection machine is an automated system that utilizes advanced technologies, such as computer vision and image processing, to detect and identify defects in glass bottles. It captures images of the bottles from different angles and analyzes them in real-time to ensure product quality and prevent issues during production and transportation. The straight glass bottle inspection machine is depicted in Fig. 1(a). On the other hand, the rotating glass bottle inspection machine is a specialized automated system designed specifically for inspecting glass bottles using a rotating mechanism. By rotating the bottles, it enables a comprehensive examination of the entire surface, ensuring a high level of accuracy in defect detection. The rotating glass bottle inspection machine is shown in Fig. 1(b).

The bottle inspection machine utilizes a vision camera-based inspection method for fully automated visual inspection of glass bottles. The process is as follows. Set the bottles in the designated area on the conveyor belt of the bottle inspection machine. The bottles are properly aligned and spaced to ensure accurate and consistent imaging. Provide proper lighting conditions for capturing clear and well-illuminated bottle images. Set up the camera parameters such as focus, exposure, and white balance to optimize image quality and clarity. Initiate the image capture process using the bottle inspection machine to trigger the camera to capture images of the bottles as they pass through the inspection area. The machine utilizes two cameras to capture images of the bottle: one camera is focused on the neck, while the other is focused on the body of

the bottle. Fig. 2 illustrates the composition of the side wall of glass bottles; (a) displays images from the neck dataset, (b) provides an example of the body dataset, and (c) showcases sample bottle images from the rotating dataset. The body dataset for glass bottles comprises 571 images, with 368 images showing defects and 203 images without defects. Similarly, the neck dataset consists of 570 images, including 250 images with defects and 320 images without defects. As for the rotating dataset, it comprises 120 images, with 70 images displaying defects and 50 images representing undamaged bottles. Each image in the body and neck datasets is standardized to a size of 900x800 pixels (width x height), ensuring consistency across the dataset. On the other hand, the images in the rotating dataset are resized to a dimension of 1024x800 pixels (width x height). The captured images are transmitted to the detection system for analysis. If any defects, such as stones, tears, or blisters, are identified on the side wall of a bottle, the detection system generates a signal to reject the faulty bottle. This process ensures that only bottles without defects continue downstream in the production line. Fig. 3(a) is shown tear defect, (b) is a blister, and (c) is stone defect on slid wall of bottle. These defects can appear throughout the bottle.

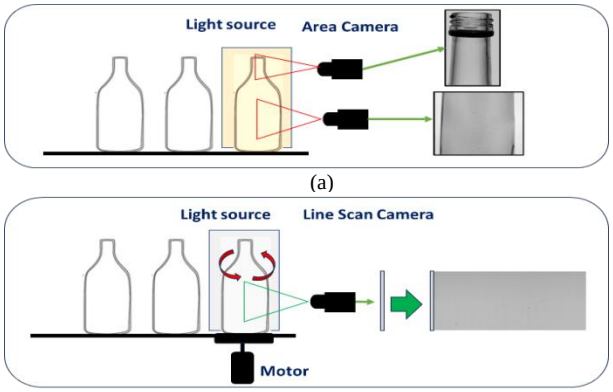

Fig. 1. Inspection machine (a) Straight glass bottle inspection machine (b) Rotating glass bottle inspection machine.

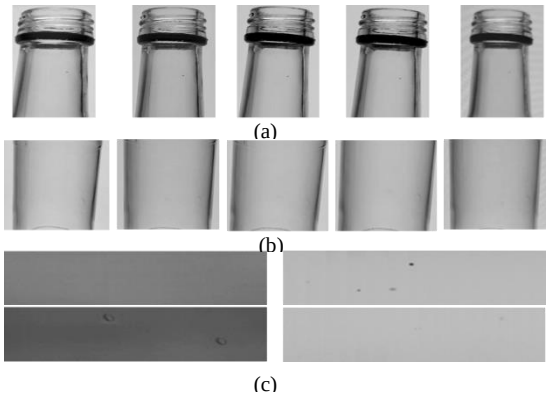

Fig. 2. Glass bottle dataset (a) Neck dataset (b) Body dataset (c) Rotating bottle dataset.

B. The Proposed Method

This section describes the proposed defect detection approach, which is divided into three parts: 1) The network architecture of CNNLSTM for extracting deep features in images, 2) the training process for creating a set of CNNLSTM models and computing the representation of class data, 3) the classification approach that suggests using the extracted deep features from unseen bottle image and applying the distance weight method to classify defect or normal glass bottle images.

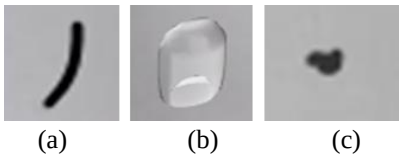

Fig. 3. Defect images. (a) Tear; (b) Blister; (c) Stone.

1) Network architecture of CNNLSTM: CNNLSTM is a convolutional neural network (CNN) combined [9] with LSTM (Long Short-Term Memory) [10] to extract deep features in images. CNNs are effective at extracting spatial features from images, while LSTMs are capable of capturing temporal dependencies in sequential data. The fusion of these two architectures enables CNNLSTM to extract deep features from images. The convolutional layers choose pre-trained deep-learning models available in the Keras library, including VGG16, ResNet50, MobileNetV3, and AlexNet. Using a pre-trained model can reduce the amount of time and resources needed to train a model. It already learned to recognize a variety of features and improve the accuracy of the model. Added LSTM to its convolutional layer. The detail of LSTM unit is shown in Fig. 4.

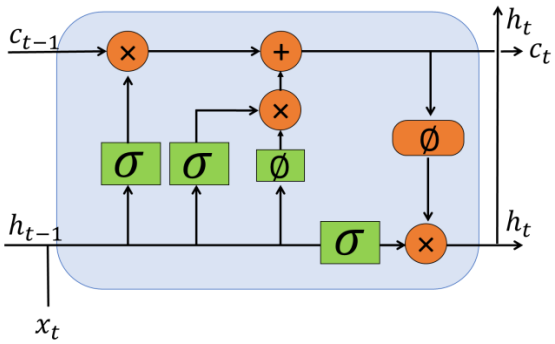

Fig. 4. LSTM.

LSTM comprises multiple memory cells, each consisting of three essential elements: write, read, and forget (delete). During each time step (t), the forget gate unit is updated according to the following process,

$$f_t = \sigma_g(W_f \mathbf{x}_t + U_f h_t + b_f), \quad (1)$$

$$i_t = \sigma_g(W_i \mathbf{x}_t + U_i h_t + b_i), \quad (2)$$

$$o_t = \sigma_g(W_o \mathbf{x}_t + U_o h_t + b_o), \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot, \sigma_c(W_c \mathbf{x}_t + b_c), \quad (4)$$

$$h_t = o_t \odot \sigma_h(c_t), \quad (5)$$

where $\mathbf{x}_t$ is the input data, f_t is the forget gate, i_t is the input/update gate, o_t is the output gate, c_t is the cell state

vector, σ is the sigmoid activation function, $\odot$ is the Hadamard product (element-wise product), and h_t is the output of the LSTM unit. In the final representation layer, an LSTM, which is a Recurrent Neural Network (RNN) suitable for sequential data, receives inputs from previous convolutional layers. The LSTM produces a set of 512 outputs, which are then passed to the next layer. The outputs of the convolutional layers are subjected to Batch-Normalization (BN) before they are fed into the LSTM layer. The fully connected layers contain 1,000 hidden neurons. The activation function of all the convolutional layers (with a max pooling size of 2 × 2) is a Rectified Linear Unit (ReLU) [11]. Deep features are extracted from these layers. The output layer contains one output neuron with a sigmoidal activation function. The size of the input images is set as 224 × 224 × 3 pixels. The CNNLSTM is trained using a Stochastic Gradient Descent (SGD) method [12]. The learning rate is set to 0.01, and the batch size is 8 images. The CNNLSTM neural network architecture is shown in Fig. 5.

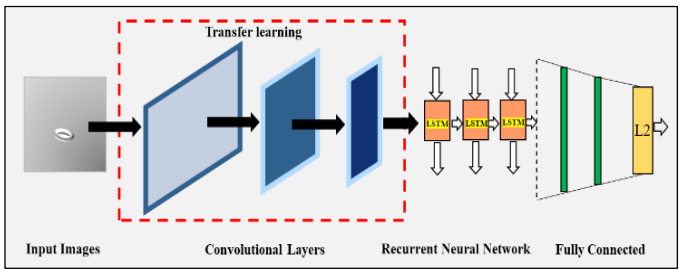

Fig. 5. CNNLSTM.

2) The proposed learning process: Our training process aims to minimize the similarity distance between the extracted deep feature vectors and the centroid of the corresponding class with the same label. Simultaneously, it maximizes the similarity distance between the centroid of good bottles and the centroid of defective bottles.

The overall learning process of creating a set of CNNLSTM and representing class data is proposed in this section. The process of creating a set of CNNLSTM models and generating representations of class data is illustrated in Algorithm 1. The inputs of the learning process include a training dataset **S**, a specified number of sub-training sets K, a threshold value θ, and the pre-trained weight m_{k-1}. The parameter K is a parameter used to determine the number of sub-training sets and the number of CNNLSTM models, which needs to be appropriately adjusted. The value of K affects the number of samples in each sub-training set used for training CNNLSTM and the diversity of CNNLSTM models. θ is a parameter used to define the initial performance of CNNLSTM in each training iteration. The pre-trained weight m_{k-1} helps to reduce training time and resource requirements, improve performance, and enable the model to perform well in limited data conditions.

Algorithm1: Pseudo code of the feature learning network method.

Input:
$\boldsymbol{S}$: A training bottle dataset: $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
K: The number of the sub datasets
θ: A performance threshold value
m_{k-1}: The pre-trained model
Output:
E: A set of CNNLSTM models
C: A set of representation of class data

1: Randomly split the training dataset into K sub datasets: $\boldsymbol{S} \rightarrow \{\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3, \ldots, \boldsymbol{s}_K\}$.
2: **For** $k = 1, 2, \ldots, K$ **do:**
3: MSE $= 0$ and $m_k = m_{k-1}$
4: **While** MSE $< \theta$ **do:**
5: Train model m_k model with $\boldsymbol{s}_k$
6: Extract features $\mathbf{F}_k$ in images by m_k.
7: Normalize all features in $\mathbf{F}_k$ with an L2-norm technique.
8: **For** i **in** $y \in \{0,1\}$ **do:**
9: Compute the representation of each class data $\mathbf{c}_{ik}$ from $\mathbf{F}_k$.
10: **End For**
11: Evaluate the distances from centroid $\mathbf{c}_{ik}$ to every feature of all $\mathbf{F}_k$ instances.
12: Predicted a bottle is normal or abnormal (h) is based on the minimum distance between $\mathbf{c}_{ik}$ and $\mathbf{F}_{k \in N^{ik}}$.
13: Compute means squared error MSE from h with true label y.
14: **End While**
15: $m_k \in E$ and $\mathbf{c}_{ik} \in C$.
16: **End For**

To construct a set of CNNLSTM models, our method involves randomly selecting images in the **S** training dataset and organizing them into several partitions or subsets, i.e., $\mathbf{S} \rightarrow \{s_1, s_2, \ldots, s_k, \ldots, s_K\}$. The purpose of creating these sub-datasets is to foster diversity in CNNLSTM models and generate representation vectors for both normal bottles and bottles with defects.

The current s_k is the training bottle images for a CNNLSTM model, m_k. Let the set models is ensemble E. The convolutional layer of the current model uses pre-train weight of previous round. The output of training with five epochs is m_k (Step 5).

In Step 6, deep features vectors $\mathbf{F}_k$ are extracted from the images in s_k. This process involves removing the output layer from the CNNLSTM and obtaining the output features through the fully connected layers. Let the deep feature vector $\mathbf{f}_{jk}$ be an instance extracted from an image in $\mathbf{F}_k$: $\mathbf{f}_{jk} = m_k(\mathbf{x}_{jk})$. The feature vector consists of 1,000 components, which corresponds to the number of hidden neurons: $\mathbf{f}_{jk} = \{v_1, v_2, \ldots, v_{nk}\}, \mathbf{f}_{jk} \in \mathbf{F}_k$.

Each feature vector is normalized using an L2-norm technique (Step 7). This normalization technique is commonly used to scale and standardize vectors in machine learning and data analysis. The L2-norm, also known as the Euclidean norm or the L2-norm, is a mathematical measure of the length or magnitude of a vector. In the context of feature normalization, the L2-norm technique calculates the square root of the sum of the squares of each component in the vector, resulting in a normalized vector with a magnitude of 1. Here, we provide an explanation of the concept of L2-norm. Consider a deep feature vector as:

$$\mathbf{F}_k = [\mathbf{f}_{1k}, \mathbf{f}_{1k}, \mathbf{f}_{1k}, \dots, \mathbf{f}_{1k}] \quad (6)$$

where

$$|\mathbf{f}_{jk}| = \sqrt{\sum_{l=1}^{n^k} |v_{lk}|^2} \quad (7)$$

where v_{lk} represents an extracted feature within $\mathbf{f}_{jk}$, and n^k denotes the total number of extracted features v in $\mathbf{f}_{jk}$. L2 serves as the final output layer of CNNLSTM, producing the extracted features as the output.

The representation of the class data $\mathbf{c}_{ik}$ is computed as the average of all feature vectors in F_k, which have been extracted from instances of the i-th class sample data (Step 9). In this step, we compute the centroids for the two classes: the centroid of the good bottles (class 0) and the centroid of the bottles with defects (class 1): $y = \{0, 1\}$. These class centroids enable the recognition of patterns on the side walls of the bottles. The centroid vector $\mathbf{c}_{ik}$ is calculated as follows:

$$\mathbf{c}_{ik} = \frac{1}{Nik} \sum_{\mathbf{f}_{jk} \in Nik} \mathbf{f}_{jk} \quad (8)$$

where $\mathbf{c}_{ik}$ constrain the average values $a: \mathbf{c}_{ik} = \{a_1, a_2, \dots, a_n\}$; and Nik is the number of all data instances of class i in s_k. The centroids are representation of class data.

Predicted a bottle is normal class or abnormal class (h) is based on the similarity distance between $\mathbf{c}_{ik}$ and $\mathbf{F}_{k \in N^{ik}}$ (Step 12). The similarity distances between each class are compared using the Euclidean distance to predict the category of the bottle. Predict normal or abnormal bottles by minimizing the similarity distance value defined as

$$\mathrm{h}_j = \arg\min_i \left\| \mathbf{f}_{jk} - c_{ik} \right\|_2, for\ \mathrm{j} = 1 : Nk, \quad (9)$$

where Nk is the number of instances in s_k.

To assess the extraction and classification performance of a trained CNNLSTM model m_k, the Mean Squared Error (denoted as MSE) is measured. If the model error (MSE) exceeds an error threshold value (θ), the deep neural model will undergo an additional training process of 5 epochs (proceed to Step 5). If the condition is met, the deep feature vectors $\mathbf{F}_k$ and the centroid c_{ik} are recomputed. Otherwise, the training process is stopped, and the model is obtained. The training iteration k is completed based on the performance condition. The threshold value θ is used to measure the performance and reduce the number of training iterations in each model generation. It serves as a criterion for selecting a set of models that require fewer training iterations compared to the training process of a normal deep neural network. At this point, the model m_k is added to the ensemble E ($m_k \in$ E), and the centroid c_{ik} is included in the set $c_{ik} \in$ C (Step 15). For the next iteration, we used pre-trained CNNLSTM weights from the previous iteration k to learn the current data $m_{k+1} \rightarrow m_k$. This method eliminates the need to reset the weights each time and only requires fine-tuning on the new set of bottle images. The overview of feature learning method is illustrated in Fig. 6.

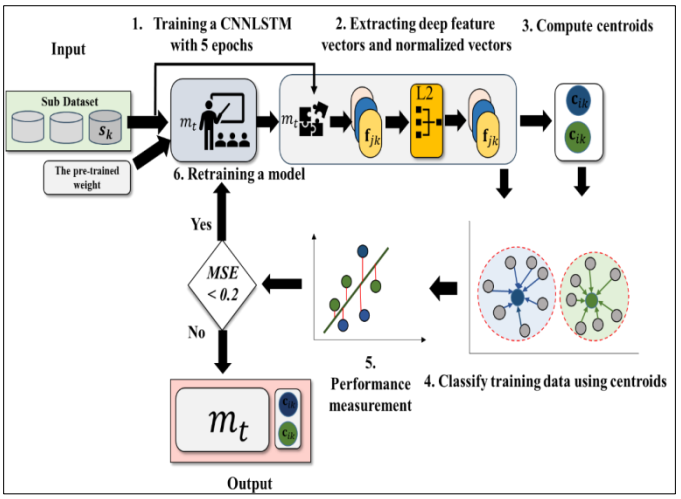

Fig. 6. Over all process of feature learning networks for glass bottle.

3) The classifying process: This section presents a detailed discussion of the proposed similarity voting method. This classification method focuses on utilizing distinctive features of defects for bottle defect detection. The similarity distance is calculated between the representation features of normal bottles and abnormal bottles. The classification steps of our method are outlined in Algorithm 2.

Algorithm2: Pseudocode for classifying bottle defects.

Input:
E: A set of CNNLSTM models
C: A set of centroids of class data
x: A bottle images
Output:

$$h_i = \underset{i}{\operatorname{argmin}} \sum_i \sum_k sd_{ik}$$

1: **For** $k = 1, 2, \dots, K$ **do:**
2: Extract deep features $\mathbf{f}_k = m_k(\mathbf{x})$
3: Normalize the deep feature vectors $\mathbf{f}_k$ by $L2$ -norm
4: **For** $i = 1: y$ **do:**
5: Calculate the distance d_{ik} between centroid $\mathbf{c}_{ik}$ to features $\mathbf{f}_k$
6: **End For**
7: Compute the distance weight sd_{ik}
8: **End For**

The input for Algorithm 2 consists of a set of CNNLSTM models E, the centroid of class data C, and the bottle image **x**.The models in set E differ from the decision boundary models of the same architecture. Extract features in bottle

image using each model in E and combine them to predict the final output. Deep feature vectors are extracted from the bottle image: $\mathbf{f}_k = m_k(\mathbf{x})$ (Line 2). Then, the deep feature vector $\mathbf{f}_k$ is normalized using the L2-norm technique: $\mathbf{f}_k = \mathrm{L2}(\mathbf{f}_k)$ (Step 3). The deep feature vector is normalized to ensure it resides in the same vector space as $\mathbf{c}_{ik}$, enabling the calculation of similarity values. The similarity distance d_{ik} between feature and centroid is expressed as follows:

$$\alpha_{ik} = \|\mathbf{f}_k - \mathbf{c}_{ik}\|_2, for\ k = 1{:}K, \quad (10)$$

where K is the number of ensemble model. Each centroid calculates the similarity distance with the features of the bottle image. The voting weight of each model is computed as follows:

$$sd_{ik} = \frac{\alpha_{ik}}{\sum_i \alpha_{ik}}, \quad (11)$$

where sd_{ik} is the voting weight of class i at model k. The voting weight is determined by calculating the average distance between the centroid and the features of both normal and abnormal bottles (Step 6).

The final output of this method is prediction class that compute as follows:

$$h_i = \underset{i}{\operatorname{argmin}} \sum_i \sum_k sd_{ik}, \quad (12)$$

where h_i is the class or category that the method predicts the input image belongs to base on its defect patterns and relationships from the training bottle images. The proposed classification method is illustrated in Fig. 7.

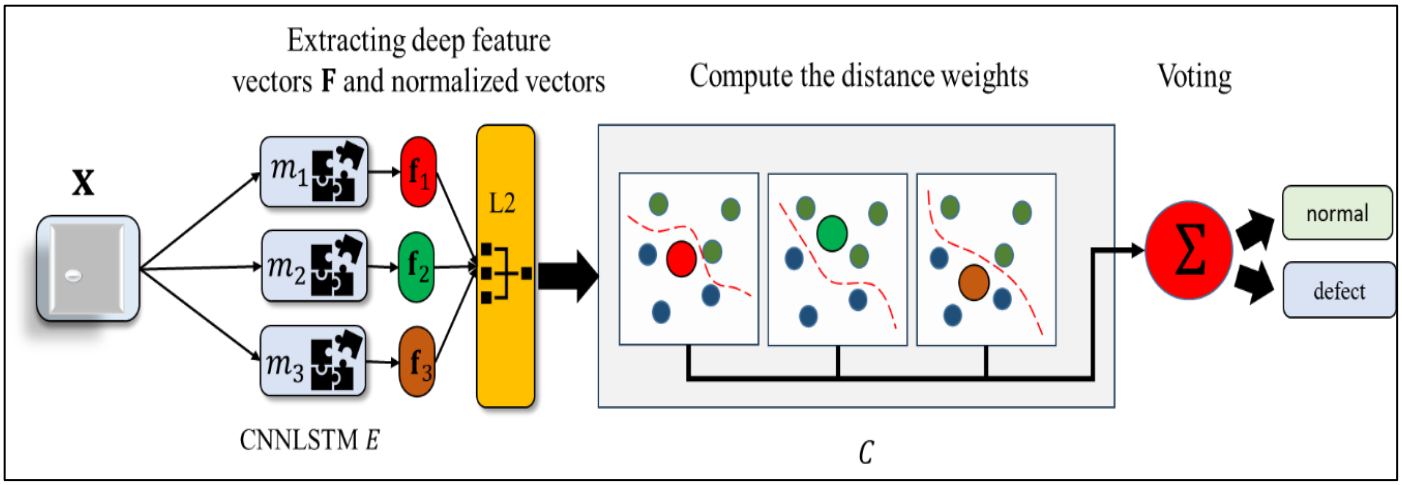

Fig. 7. The overall process of glass bottle classification.

IV. Experimental Results and Discussion

In this section, we present a comprehensive set of experiments conducted to assess the performance of the CNNLSTM method.

A. Experimental Setup

These experiments used three glass bottle image datasets, namely the body dataset, the neck dataset, and the rotating dataset. These datasets consist of three primary types of defects: stone, tear, and blister. The body dataset of glass bottles comprises a total of 571 images. For the training set, 292 images, consisting of 161 images with defects and 131 images without defects were used to train models. The remaining 279 images constituted the testing set, with 207 images containing defects and 72 images without defects. The neck dataset encompasses 570 images of glass bottles. The training set include 288 images, with 150 images exhibiting defects and 138 images without defects. The testing set includes 282 images, with 100 images containing defects and 182 images without defects. The rotating dataset encompasses 120 images of glass bottles. The training set include 80 images, with 47 images exhibiting defects and 33 images without defects. The testing set includes 40 images, with 23 images containing defects and 17 images without defects. The distribution of each defect type in both the training and testing datasets for body, neck and rotating bottles is presented in Table I.

TABLE I. Number of Training and Testing Images for each Defect Type in the Body, Neck and Rotating Datasets

Defect type	Body		Neck		Rotating	
	Train	*Test*	*Train*	*Test*	*Train*	*Test*
Stone	81	127	80	71	20	10
Tear	30	28	20	17	12	5
Blister	50	52	50	12	15	8
Normal	131	72	138	182	33	17
Total	292	279	288	282	80	40

To assess the effectiveness of our proposed method on the defect dataset, various image processing techniques, including image thresholding and edge detection, along with machine learning models such as Anomaly Detection with Autoencoder (ADA), ResNet50, AlexNet, VGG16, and MobileNetV3 were used to construct models for comparison. Image thresholding, a commonly used technique in image processing using OpenCV, involves setting pixel values to either 0 or a maximum value based on a predefined threshold [13]. Edge detection is a

digital image processing technique used to identify and extract edges in an image, representing abrupt changes in intensity or color between adjacent pixels [14]. ResNet50 is a deep convolutional neural network architecture introduced in 2016 [15]. ResNet-50 consists of 50 layers and employs residual blocks, each containing convolutional layers and shortcut connections. It has achieved state-of-the-art performance in various image recognition tasks. AlexNet is a deep convolutional neural network architecture proposed in 2012 [16]. AlexNet comprises eight layers, including five convolutional layers and three fully connected layers. It introduced several key innovations, such as using ReLU as an activation function and applying dropout regularization. VGG16 developed by the Visual Geometry Group (VGG) at the University of Oxford in 2014, is a widely recognized deep convolutional neural network architecture primarily used for image classification [17]. It consists of 13 convolutional layers followed by 3 fully connected layers, and it has demonstrated outstanding performance on numerous computer vision tasks. MobileNetV3, introduced by Google in 2019, is specifically designed for mobile and embedded devices that have limited computational resources [18]. This architecture aims to strike a balance between model size and accuracy, making it suitable for efficient deployment on devices with constrained hardware capabilities. In addition to these models, we also employed the Anomaly Detection with Autoencoder (ADA) technique. ADA detects anomalies by evaluating the reconstruction loss, comparing it to a predefined threshold. If the reconstruction loss surpasses the threshold, the input is classified as an anomaly.

For training our method and deep learning-based models, we used a training set comprising 292 images for the body dataset and 288 images for the neck dataset. The training epochs are set at 500 for VGG16, RestNet50, MobileNetV3, and ADA. The input error threshold, θ, of the proposed method is set to 0.2, and the number of ensemble models, K, is set to 3. All the experiments, including the proposed defect detection method and the comparison methods, were conducted on a personal computer equipped with an AMD Ryzen 7 5000 series processor and an NVIDIA GTX 1650 GPU, allowing for efficient computation and analysis of the results.

B. Experimental Result

Four evaluation metrics are adopted for the analysis: Accuracy, Recall, Precision, and F1-score. These metrics are widely used in image defect detection and provide comprehensive insights into the performance of the methods. They are defined as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (13)$$

$$Recall = \frac{TP}{TP+FN} \quad (14)$$

$$Precision = \frac{TP}{TP+FP} \quad (15)$$

$$F_1 = 2 \times \frac{Precition \times Recall}{Precition+Recall} \quad (16)$$

TP refers to the number of defect bottles (positive) that are correctly predicted as positive by the model. FN represents the number of defect bottles (positive) that are incorrectly predicted as bottle without defects (negative) by the model. TN indicates the number of the bottles without defect (negative) that are correctly predicted as negative by the model. Lastly, FP signifies the number of the bottles without defects (negative) that are mistakenly predicted as positive by the model.

Fig. 8 presents visualizations of defect classification results of CNNLSTM for the body, neck, and rotation of bottle datasets.

The classification accuracy of various methods on the defect datasets is presented in Table II. The results for the body dataset demonstrate that the standalone VGG16 method outperformed all other methods, achieving an accuracy of 80%. ResNet50 is ranked second in terms of accuracy. Among the CNNLSTM models, CNNLSTM-VGG16 achieved the highest accuracy compared to the other CNNLSTM variants. Conversely, the image threshold method exhibited the lowest accuracy on this dataset.

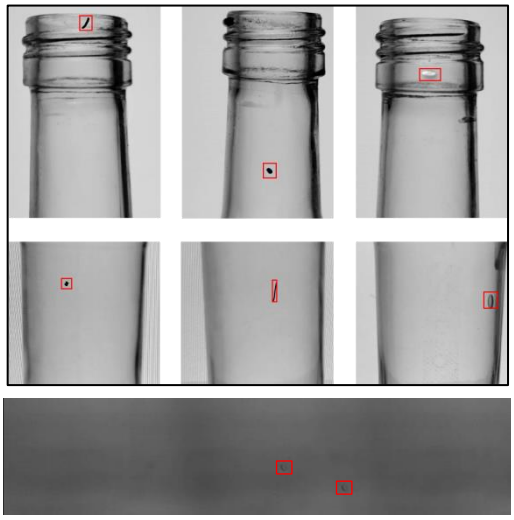

Fig. 8. Examples of classification results of the CNNLSTM.

TABLE II. THE PERFORMANCE ON THE BODY IMAGES, NECK IMAGES AND ROTATING IMAGES DATASETS IN TERMS OF ACCURACY

Methods	Body dataset	Neck dataset	Rotating dataset
	Accuracy (%)	*Accuracy (%)*	*Accuracy (%)*
CNNLSTM-ResNet50	71	**95**	62
CNNLSTM-AlexNet	75	89	97
CNNLSTM-MobileNetV3	64	92	**98**
CNNLSTM-VGG16	77	80	90
ResNet50	79	79	60
AlexNet	74	92	60
VGG16	**80**	85	95
MobileNetV3	70	72	70
ADA	74	75	60
Image threshold	62	65	65
Edge detection	70	72	63

The neck dataset results reveal that CNNLSTM-ResNet50 attained the highest accuracy of 95%, closely followed by CNNLSTM-AlexNet, CNNLSTM-MobileNetV3, CNNLSTM-VGG16, and AlexNet. Among the standalone methods, AlexNet exhibited the highest accuracy on this dataset. Conversely, the image threshold method yielded the lowest accuracy for the neck dataset.

Upon comparing the results of the rotating dataset images, it is obvious that CNNLSTM-MobileNetV3 achieved the highest accuracy. CNNLSTM-AlexNet and the standalone VGG16 also demonstrated good performance with high accuracies. Conversely, ADA exhibited the lowest accuracy percentages on this dataset.

Overall, the results indicate that different methods perform differently on the three datasets. The standalone VGG16 achieved the highest accuracy on the body dataset. CNNLSTM generally achieved the highest accuracy on both the neck and rotating datasets.

In Table III, the performance measures, including Recall, Precision, and F1-score, are presented for the body dataset. VGG16 demonstrates the highest level of correctness, evident from its superior metrics across Recall, Precision, and F1-score. Following closely in terms of correctness, CNNLSTM-AlexNet achieves relatively high Precision and F1-score. On the other hand, Image Threshold exhibits the lowest level of correctness among the methods listed.

TABLE III. THE PERFORMANCE ON THE BODY IMAGES DATASETS IN TERMS OF RECALL, PRECISION, AND F1-SCORE

Methods	Recall (%)	Precision (%)	F1-score (%)
CNNLSTM-ResNet50	80	73	70
CNNLSTM-AlexNet	82	74	74
CNNLSTM-MobileNetV3	71	66	63
CNNLSTM-VGG16	83	75	75
ResNet50	71	75	73
AlexNet	73	74	70
VGG16	**86**	**78**	**77**
MobileNetV3	57	57	57
ADA	51	55	53
Image Threshold	50	53	51
Edge detection	78	72	67

VGG16 outperforms the CNNLSTM method in terms of accuracy on the body image dataset. The VGG16 is advantaged by being pre-trained on a large and diverse collection of images. Its capacity to identify various features associated with bottle defects, coupled with the requirement for a comprehensive and diverse training set, contributes to its superior performance.

Table IV presents the Recall, Precision, and F1-score metrics of the proposed method and the compared methods for neck image datasets. Among the CNNLSTM methods, CNNLSTM-ResNet50 demonstrates the highest recall, precision, and F1-score, indicating its strong performance. CNNLSTM-MobileNetV3 follows closely behind with good recall, precision, and F1-score. Among the individual models, AlexNet demonstrates good performance, achieving high recall, precision, and F1-score. MobileNetV3 exhibits moderate performance with decent recall and F1-score, but its precision is relatively lower. ADA demonstrates moderate performance across all metrics. On the other hand, Image Threshold and Edge detection methods exhibit lower performance, with lower recall, precision, and F1-score.

Table V presents the Recall, Precision, and F1-score metrics of the proposed method and the compared methods for rotating image datasets. Among the CNNLSTM methods, CNNLSTM-MobileNetV3 and CNNLSTM-AlexNet demonstrate outstanding performance, achieving high recall, precision, and F1-score. CNNLSTM-VGG16 also performs well, with consistently high metrics. However, CNNLSTM-ResNet50 exhibits comparatively lower performance across the metrics.When considering individual models, VGG16 stands out with consistently high scores for recall, precision, and F1-score. ResNet50, AlexNet, and MobileNetV3 display varying levels of performance, with some metrics being higher or lower than others. In terms of the additional methods, Image Threshold exhibits relatively high recall but lower precision and F1-score. ADA demonstrates quite low performance across the metrics.

TABLE IV. THE PERFORMANCE ON THE NECK IMAGES DATASETS IN TERMS OF RECALL, PRECISION, AND F1-SCORE

Methods	Recall (%)	Precision (%)	F1-score (%)
CNNLSTM-ResNet50	**95**	**92**	**96**
CNNLSTM-AlexNet	89	85	92
CNNLSTM-MobileNetV3	92	90	93
CNNLSTM-VGG16	80	77	78
ResNet50	79	71	88
AlexNet	92	89	94
VGG16	85	79	90
MobileNetV3	72	62	80
ADA	75	70	75
Image Threshold	65	52	53
Edge detection	72	65	70

Table VI displays the training time and detection time of all detection methods. Among the CNNLSTM methods, CNNLSTM-VGG16 has the longest training time on all three datasets, followed by CNNLSTM-ResNet50. However, CNNLSTM-ResNet50 has the shortest detection time across all datasets, indicating faster inference speed. CNNLSTM-AlexNet and CNNLSTM-MobileNetV3 also demonstrate relatively shorter training and detection times. For the individual models, ResNet50, AlexNet, VGG16, and MobileNetV3 have consistent training and detection times across all three datasets. Among the additional methods, ADA has relatively short training and detection times, while Image

Threshold and Edge detection methods have negligible training times but slightly longer detection times compared to the CNNLSTM and individual models.

Overall, the experimental results demonstrate the effectiveness of the proposed CNNLSTM models for defect detection on all three datasets. The proposed models achieved competitive accuracy and performed well in terms of evaluation metrics. The training and detection times varied among the models, with CNNLSTM-AlexNet demonstrating the shortest training time. These findings can guide the selection of appropriate models based on the trade-off between accuracy and computational time requirements in practical applications.

TABLE V. THE PERFORMANCE ON THE ROTATING IMAGES DATASETS IN TERMS OF RECALL, PRECISION, AND F1-SCORE

Methods	Recall (%)	Precision (%)	F1-score (%)
CNNLSTM-ResNet50	66	71	61
CNNLSTM-AlexNet	**97**	96	96
CNNLSTM-MobileNetV3	**97**	**98**	**98**
CNNLSTM-VGG16	91	90	89
ResNet50	55	58	52
AlexNet	52	79	42
VGG16	95	94	94
MobileNetV3	62	79	52
ADA	58	55	56
Image Threshold	77	69	63

TABLE VI. THE TRAINING TIME AND DETECTION TIME ON THE BODY IMAGES AND NECK IMAGES DATASETS

Methods	Body dataset		Neck dataset		Rotating dataset	
	Training time (second)	*Detection time (second)*	*Training time (second)*	*Detection time (second)*	*Training time (second)*	*Detection time (second)*
CNNLSTM-ResNet50	154.29	0.11	143.55	0.054	65.4	0.054
CNNLSTM-AlexNet	103.59	0.12	59.22	0.065	63.12	0.06
CNNLSTM-MobileNetV3	141.93	0.12	80.69	0.061	60.28	0.062
CNNLSTM-VGG16	304.41	0.074	304.84	0.025	104.44	0.025
ResNet50	1,500	0.5	1,500	0.5	1,500	0.5
AlexNet	690	0.3	690	0.3	690	0.3
VGG16	1,800	0.5	1,800	0.5	1,800	0.5
MoboleNetV3	780	0.3	780	0.3	780	0.3
ADA	180	0.2	180	0.2	180	0.2
Image Threshold	-	0.002	-	0.002	-	0.002
Edge detection	-	0.001	-	0.001	-	0.001

V. CONCLUSION

This research presents an approach for detecting defects in glass bottles using a combination of deep convolutional neural networks with long short-term memory (CNNLSTM) and instance-based classification algorithms. The CNNLSTM combines two types of deep neural networks to extract features and create a representation of class data, which is then used for defect detection. The proposed method is compared with other well-known defect detection methods. Experimental results demonstrate that the proposed method outperforms other defect detection methods in terms of detection performance. Additionally, the proposed method requires less training time, making it suitable for efficient glass bottle defect detection in production lines.

In future work, we intend to extend the application of the proposed algorithm to different types of bottles and explore additional defect types. The flexibility of the CNNLSTM allows for varying the number of models and designing customized detection layers to adapt to specific datasets. Furthermore, there is potential to extend this method to semi-supervised learning, enabling the detection of defects in other product categories. These future directions will further enhance the capabilities and applicability of our approach in defect detection and contribute to the advancement of quality control systems in various industries.

REFERENCES

[1] H. Xie, F. Lu, G. Ouyang, X. Shang and Z. Zhao, "A rapid inspection method for encapsulating quality of PET bottles based on machine vision," *IEEE International Conference on Computer and Communications (ICCC)*, Chengdu, China, pp. 2025-2028, 2017.

[2] L. Fu, S. Zhang, Y. Gong and Q. Huang, "Medicine Glass Bottle Defect Detection Based on Machine Vision," *Chinese Control And Decision Conference (CCDC)*, Nanchang, China, pp. 5681-5685, 2019.

[3] M. A. E. Latina, J. Van Russel R. Dela Cruz and F. D. Delos Santos, "Empty Glass Bottle Defect Detection Based on Deep Learning with CNN Using SSD MobileNetV2 Model," *IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, Boracay Island, Philippines, pp. 1-6, 2022.

[4] W. Gong, K. Zhang, C. Yang, M. Yi and J. Wu, "Adaptive Visual Inspection Method for Transparent Label Defect Detection of Curved Glass Bottle," *International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, Chongqing, China, pp. 90-95, 2020.

[5] G. A. De Vitis, A. Di Tecco, P. Foglia, and C. A. Prete, "Fast Blob and Air Line Defects Detection for High Speed Glass Tube Production Lines," *Journal of Imaging*, vol. 7, no. 11, p. 223, Oct. 2021.

[6] T. Zhang, B. Ding, X. Zhao, G. Liu, Z. Pang, "LearningADD: Machine learning based acoustic defect detection in factory automation," Journal of Manufacturing Systems, vol. 60, pp. 48-58, 2021.

[7] X. Zhou, Y. Wang, Q. Zhu, J. Mao, C. Xiao, X. Lu and H. Zhang, "A Surface Defect Detection Framework for Glass Bottle Bottom Using Visual Attention Model and Wavelet Transform," *in* IEEE Transactions on Industrial Informatics, vol. 16, no. 4, pp. 2189-2201, 2020.

[8] X. Zhou, Y. Wang, Q. Zhu, J. Mao, C. Xiao, X. Lu, and H. Zhang, "Automated Visual Inspection of Glass Bottle Bottom With Saliency Detection and Template Matching." IEEE Transactions on Instrumentation and Measurement, vol. 68, no. 11, pp. 4253-4267, 2019.

[9] S. Montaha, S. Azam, A. K. M. R. H. Rafid, M. Z. Hasan, A. Karim and A. Islam, "TimeDistributed-CNN-LSTM: A Hybrid Approach Combining CNN and LSTM to Classify Brain Tumor on 3D MRI Scans Performing Ablation Study," in IEEE Access, vol. 10, pp. 60039-60059, 2022.

[10] S. Xiang and B. Tang, "CSLM: Convertible Short-Term and Long-Term Memory in Differential Neural Computers," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 9, pp. 4026-4038, 2021.

[11] V. Nair, and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," In Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807-814, 2010.

[12] L. Bottou, "Large-Scale Machine Learning with Stochastic Gradient Descent," In: Lechevallier, Y., Saporta, G. (eds) Proceedings of COMPSTAT'2010. Physica-Verlag HD, 2010.

[13] R. C. Gonzalez and R. E. Woods, *Digital Image Processing* (3th ed.), Pearson International Edition, 2008.

[14] H. Singh, and N. Kaur, "A review of edge detection techniques for image segmentation," *Journal of Computational and Theoretical Nanoscience*, vol.15, no.9, pp.4131-414, 2018.

[15] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770-778, 2016.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Commun. ACM 60, vol. 6, pp. 84–90, 2017.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, pp. 1409–1556, 2014.

[18] A. Howard, M. Sandler, B. Chen, W. Wang, L. C. Chen, M, Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam and Q. Le, "Searching for MobileNetV3," *IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), pp. 1314-1324, 2019.