Deep Learning and Computer Vision-Based System for Detecting and Separating Abnormal Bags in Automatic Bagging Machines

Trung Dung Nguyen, Thanh Quyen Ngo, Chi Kien Ha
Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

***Abstract*—This paper presents a novel deep learning and computer vision-based system for detecting and separating abnormal bags within automatic bagging machines, addressing a key challenge in industrial quality control. The core of our approach is the development of a data collection system seamlessly integrated into the production line. This system captures a comprehensive variety of bag images, ensuring a dataset representative of real-world variability. To augment the quantity and quality of our training data, we implement both offline and online data augmentation techniques. For classifying normal and abnormal bags, we design a lightweight deep learning model based on the residual network for deployment on computationally constrained devices. Specifically, we improve the initial convolutional layer by utilizing ghost convolution and implement a reduced channel strategy across the network layers. Additionally, knowledge distillation is employed to refine the model's performance by transferring insights from a fully trained, more complex network. We conduct extensive comparisons with other convolutional neural network models, demonstrating that our proposed model achieves superior performance in classifying bags while maintaining high efficiency. Ablation studies further validate the contribution of each modification to the model's success. Upon deployment, the model demonstrates robust accuracy and operational efficiency in a live production environment. The system provides significant improvements in automatic bagging processes, combining accuracy with practical applicability in industrial settings.**

***Keywords*—Automatic bagging machines; deep learning; computer vision; bags classification; data augmentation**

I. Introduction

An automatic bagging machine is a type of packaging machinery designed to automatically fill products into bags and then seal them. These machines are widely used in various industries, including food, agriculture, chemical, and manufacturing, for efficient and rapid packaging solutions. Automatic bagging machines can handle a wide range of bag materials, sizes, and types, such as plastic, paper, and fabric bags. An automatic bagging machine comprises several stages and components: a product feeding system, which delivers the product to the bagging area; a weighing and filling system, which ensures that each bag is filled with the correct amount of product; a bag supply and opening unit, which automatically takes a bag from the supply, opens it, and positions it for filling; and a sealing system, which seals the bag by heat sealing, stitching, or using adhesives. Fig. 1 illustrates the comprehensive setup of the automatic bagging machine utilized in our research, highlighting each of the critical components and stages that facilitate the seamless transition from product feeding to the final sealing process. Among the components, the bag supply and opening unit is a critical component, ensuring that bags are consistently and accurately supplied and opened for the product filling process. This unit is designed to handle a variety of bag types and materials, including paper, plastic, and woven fabric, with varying levels of thickness and rigidity. To maintain the maximum performance of automatic bagging machines, constant bag quality is essential at all times. Criteria related to bag quality include the bag mouth, bag position, and bag surface. For the bag mouth, the edges must be perpendicular to the sides and in a straight line. For the bag position, bags must lie flat throughout their entire length and width. For the bag surface, it must be free of folds and/or wrinkles that could result from improper storage. Fig. 3(b) indicates some instances where bag quality does not meet the standards.

Any errors related to bag quality can cause serious problems. For example, a bag that is too weak might tear during the picking or opening process, while a bag with inconsistent dimensions might not align properly with the machine’s mechanisms. These issues can halt production, necessitating manual intervention to clear the jam and restart the machine. Even if abnormal bags are successfully opened and filled, they may not seal properly, potentially compromising the integrity of the packaging. This can affect product safety, shelf life, and customer satisfaction. Inconsistent bag quality can also lead to poor presentation of the final product, affecting brand perception. The failure to properly handle abnormal bags can lead to increased material waste, as bags that are damaged during the process or that fail quality checks after filling and sealing are discarded. This not only increases material costs but can also lead to higher labor costs associated with troubleshooting and rectifying issues caused by using these bags.

To mitigate the issues caused by abnormal bags, manufacturers may implement quality control measures such as pre-screening bags before they enter the supply unit, adjusting machine parameters to better accommodate variation in bag quality, or investing in more advanced detection and handling systems that can adapt to a wider range of bag qualities. Implementing a rigorous quality assurance program with suppliers to ensure that bags meet all necessary specifications before they reach the production line is also crucial. Modern

bag supply and opening units often rely on sensors and automated systems to detect and adjust the bags being fed into the machine. Abnormal bags might not be detected accurately by these systems, leading to misfeeds or incorrect adjustments that can compromise the packaging process.

Fig. 1. The comprehensive setup of the automatic bagging machine utilized in our research.

Over the years, the adoption of machine vision for automatic quality inspection has seen significant advancements across various industries, revolutionizing how quality control is implemented and ensuring higher standards of accuracy and efficiency. In the metal casting industry, machine vision has been instrumental in detecting defects such as cracks, porosity, and misruns on cast parts [1]. The manufacturing industry has broadly embraced machine vision for a range of applications, from verifying product assembly to ensuring the accuracy of labeling and packaging [2]. In the agricultural sector, machine vision has been applied to the inspection of the external quality of date fruits [3]. Automatic rice-quality inspection systems represent another remarkable application [4]. These systems employ machine vision to classify rice grains by size, shape, color, and texture, as well as to detect impurities. In the wood industry, particularly in the inspection of hardwood flooring products, machine vision systems have been developed to detect surface defects such as knots, cracks, and color variation [5]. These systems can inspect flooring panels at high speeds, ensuring that only those meeting strict quality standards reach the consumer. Overall, the developments in machine vision for automatic quality inspection across these varied industries emphasize a trend towards greater automation and precision in quality control processes. By leveraging advanced imaging technologies and machine learning algorithms, industries are not only able to enhance the efficiency of their operations but also significantly improve the quality of their products, benefiting both manufacturers and consumers alike.

In recent years, deep learning has seen remarkable developments, transforming the landscape of production industries with its unprecedented capabilities in data analysis, pattern recognition, and autonomous decision-making. Leveraging vast amounts of data, deep learning algorithms have become proficient at identifying complex patterns and anomalies that elude traditional computational methods. This advancement has been particularly impactful in automating quality control processes, predictive maintenance, and enhancing operational efficiencies across various sectors. However, deploying deep learning models, especially Convolutional Neural Networks (CNNs), on resource-constrained embedded devices poses significant challenges. First and foremost, these devices typically have limited processing power, which can make it difficult to run the computationally intensive operations required by CNNs in real-time. Additionally, embedded devices often have restricted memory capacity, constraining the size of the models that can be deployed and limiting the amount of data that can be processed at once. Energy consumption is another critical concern, as many embedded devices operate on battery power or in energy-sensitive environments. The high computational demands of CNNs can lead to rapid battery depletion or require compromises in performance to conserve energy. Model complexity versus performance trade-offs also present a challenge. Simplifying models to fit the constraints of embedded devices can lead to reduced accuracy and efficacy. Finally, the diversity of hardware in embedded systems necessitates custom optimization for each deployment, increasing development time and complexity. Addressing these challenges requires innovative solutions, including model compression techniques, specialized hardware accelerators, and efficient algorithm design to make CNNs viable for embedded applications.

Based on the above analysis, this paper introduces a novel deep learning and computer vision-based system designed to enhance the efficiency and accuracy of automatic bagging machines by classifying bags as normal or abnormal. By integrating a sophisticated data collection system directly into the production line and employing advanced data augmentation techniques, this study addresses the critical need for high-quality, diversified datasets in machine learning. Central to our approach is the development of a lightweight deep learning model, based on the modified ResNet-18 architecture, which is specifically optimized for deployment on resource-constrained devices such as the Raspberry Pi 4. The contributions of this paper are summarized as follows:

- We demonstrate the efficacy of combining offline and online data augmentation techniques to substantially improve model robustness and generalizability.
- Our customized lightweight ResNet-18 model, featuring an innovative initial convolution layer modification, channel reduction, and the application of knowledge distillation, demonstrates a novel approach to optimizing deep learning models for efficient deployment on embedded systems.
- Through comprehensive comparisons with other CNN models and ablation studies, we provide valuable insights into the model's decision-making processes and its superior performance.
- The successful deployment of our model on the Raspberry Pi 4 not only proves its operational viability in real-world industrial settings but also sets a benchmark for future research in deploying deep learning models on resource-constrained devices.

II. Literature Review

A. Image Classification

Image classification, a pivotal task in the field of computer vision, has experienced significant evolution over the past decade, predominantly shaped by the advent and advancement of CNNs and, more recently, Transformer models.

CNNs have established themselves as the backbone of image classification tasks, marked by their ability to automatically and adaptively learn spatial hierarchies of features from image data. The foundational model, LeNet [6], introduced in the late 1990s, set the stage for the use of CNNs in image recognition. However, it was AlexNet's [7] victory in the ImageNet challenge in 2012 that truly catalyzed the deep learning revolution, highlighting CNNs' potential to achieve remarkable accuracy in classifying images across thousands of categories. Subsequent architectures, such as VGG [8], GoogLeNet [9], ResNet [10], MobileNet [11-13], ShuffleNet [14-15] and EfficientNets [16] have introduced innovations such as deeper networks, inception modules, and residual connections, significantly improving performance on various image classification benchmarks. These developments have not only enhanced the accuracy and efficiency of image classification tasks but have also broadened the application scope of CNNs to include areas like medical image analysis, autonomous vehicles, and surveillance systems, emphasizing their versatility and robustness in extracting meaningful patterns from visual data.

The introduction of Transformers in image classification [17], initially conceived for natural language processing tasks [18], marks the latest significant innovation in the field. The seminal work, "Attention Is All You Need," introduced the Transformer model, which relies on self-attention mechanisms to process data in parallel, significantly reducing the need for sequential data processing and enabling the model to weigh the importance of different parts of the input data. The adaptation of Transformer models for image classification, notably through architectures like Vision Transformer, has opened new avenues for research and application. Unlike CNNs, Transformers do not inherently process spatial hierarchies but instead treat the image as a sequence of patches, applying self-attention to understand the global context of the image, which can lead to superior performance in certain contexts. This paradigm shift toward using Transformers for image classification highlights the field's ongoing evolution and the continuous search for models that can more effectively capture and interpret the complex patterns present in visual data. Despite their promising capabilities, the adoption of Transformer models in image classification also presents challenges, including the need for large-scale datasets for training and higher computational resources, setting the stage for ongoing research and development in optimizing these models for wider application. To address the challenges posed by the adoption of Transformer models in image classification, researchers have focused on designing lightweight and efficient Vision Transformers that require fewer computational resources and can be trained with smaller datasets. Key methods include the introduction of techniques such as model pruning [19-22], where redundant or non-essential parts of the model are removed without significantly impacting performance, and knowledge distillation [23-24], where a smaller, more efficient model is trained to emulate the performance of a larger, more complex model. Additionally, some approaches utilize more efficient self-attention mechanisms [25-30], which reduce the computational complexity by focusing on only the most relevant parts of the input data, and employing token-based methods that decrease the number of input tokens to the Transformer, significantly reducing the computational load while maintaining high accuracy. These innovations represent a significant step towards making Vision Transformers more accessible for a broader range of applications, especially in environments with limited computational capacity.

B. Deep Models for Classification Task in Production Industries

Deep learning models have profoundly impacted production industries by enhancing classification tasks with unprecedented accuracy and efficiency. In the manufacturing domain, Xu et al. [31] proposed a model that leverages high-resolution vision sensors and deep learning techniques to classify and rate multi-category steel scrap. The model significantly improved the accuracy and fairness of steel scrap quality evaluation in recycling processes. Vikanksh et al. [32] introduced the NSLNet framework, which combines ImageNet for feature extraction with adversarial training in the feature space through Neural Structure Learning, aiming to overcome the challenges of limited annotated datasets and decreased prediction accuracy due to image perturbations in steel surface defect identification. In a study by Mathieu et al. [33], a method was introduced involving a three-step approach of data collection, classification, and supervised learning using CNNs. This method aims to automate quality control of open mouth bag sealings in industrial bagging systems. This study contributed a novel CNN architecture for the image classification of open mouth bags, demonstrating promising results in automating quality control within the food industry's industrial bagging systems.

Deep learning models have also found application in the agriculture sector. Padmapriya et al. [34] introduced a multi-stacking ensemble model combined with a novel feature selection algorithm, leveraging both machine learning and deep learning models for accurate multiclass soil classification, essential for smart agriculture advancements. Gill et al. [35] proposed a model that utilizes CNN, Recurrent Neural Network, and Long-short Term Memory deep learning methods for optimal image feature extraction and selection, applying these features to classify fruits effectively. The model outperformed traditional methods in handling the complex and heterogeneous nature of fruit recognition and classification tasks. Recently, Shewale et al. [36] proposed a model that utilizes deep learning, specifically CNNs, combined with image processing, to automatically extract features from leaf images for the identification, classification, and diagnosis of plant leaf diseases. This research provided an automated, high-precision disease diagnosis system for tomato plants that bypasses manual feature engineering and segmentation, offering a scalable solution for crop disease diagnosis globally through the application of deep learning on extensive, real-time image datasets.

In the food industry, specifically for assessing the quality of packaged food, Han et al. [37] introduced a study featuring a rapid, non-destructive method for estimating nut quality. This method uses hyperspectral imaging coupled with deep learning classification, specifically a CNN, to assess the quality of unblanched Canarium indicum kernels based on peroxide values. Kazi et al. [38] explored the use of transfer learning with classical and residual CNN architectures for classifying different types of fruits and their freshness, moving beyond traditional CNN implementations. In the textile industry, deep learning has introduced new capabilities for fabric inspection, identifying weaving faults, and ensuring pattern consistency. Huang et al. [39] introduced an efficient CNN model designed for fabric defect segmentation and detection, which requires only a minimal number of defect samples for training, thus significantly reducing manual annotation costs. Wei et al. [40] introduced the BIVI-ML model that integrates three bioinspired visual mechanisms (i.e., visual gain, attention, and memory) into a deep CNN framework to address the challenges of multilabel textile defect classification, such as intersected defects and label correlations. This approach enhances resolution, focuses attention on defects, and accurately associates relevant labels for effective multilabel classification.

III. METHODOLOGY

In this section, we first introduce the data collection process, detailing the system implemented within the production line to gather a diverse and representative dataset of bag images. Following this, the data augmentation subsection explains how we leverage both offline and online techniques to enhance the dataset's quality and variability. Lastly, we discuss the design and optimization of a lightweight deep model for bags classification, focusing on our custom modifications to the ResNet-18 architecture, which includes adjustments for efficient operation on resource-constrained devices like the Raspberry Pi 4.

A. Data Collection

Designing a deep learning-based system for classifying normal and abnormal bags in automatic bagging machines presents several significant challenges, particularly in the domain of data collection. The effectiveness of such a system is heavily dependent on the quality and variety of the dataset used to train it. One of the primary challenges is the vast diversity of bags. These bags can vary widely in terms of size, shape, material, color, and design. Consequently, it is necessary to obtain the broadest possible variety of bag types to ensure that the control quality can be effectively managed across all potential items the system might encounter. This diversity is critical to developing a robust model capable of accurately identifying anomalies in any given bag. Another significant challenge is the scarcity of abnormal bags. Abnormalities can range from minor defects such as slight tears or misprints to more significant issues like incorrect sizing or completely torn bags. However, these occurrences are typically rare in a well-maintained production environment. This scarcity presents a problem for data collection because deep learning models require a substantial amount of data to learn from. The insufficient number of abnormal bags means the system may not observe enough examples of abnormalities during the data collection phase, leading to a model that might struggle to recognize less common or more subtle defects. Data collection is also hindered by the dynamic conditions under which bagging operations occur. Factors such as lighting, background, and speed of the conveyor can significantly affect the quality of the images captured for training the model. Consistency in these conditions is challenging to maintain, yet critical for training a model that is reliable under the diverse conditions it will encounter in real-world applications.

To tackle the challenges associated with collecting a diverse and representative dataset for training a deep learning system for classifying normal and abnormal bags in automatic bagging machines, a sophisticated data collection system has been implemented directly into the automatic bagging machine's production line, as shown in Fig. 2. This system employs a high-resolution Vieworks CMOS VC-25MC-M/C 30D area camera, renowned for its exceptional image quality and reliability. The camera is equipped with a 6 mm fixed focal lens, providing a wide field of view while maintaining sufficient detail for identifying both gross and subtle abnormalities in bags. The camera's parameters have been meticulously adjusted to optimize the lighting conditions and speeds at which bags are processed on the production line. This adjustment ensures that the images captured are of high quality and reflect the diverse conditions under which the system must operate. By integrating the camera directly into the production environment, the data collection system is able to capture images of bags under the actual conditions they will be encountered, thereby enhancing the realism and applicability of the training data. This setup is linked to a computer system equipped with both a powerful CPU and a GPU. The GPU, in particular, is crucial for processing the high volume of image data in real-time, allowing for immediate feedback and adjustments to the data collection process if needed. This computational power also supports the use of advanced image processing and augmentation techniques, which can artificially expand the dataset by modifying existing images to simulate a

wider range of abnormalities and conditions. All the data collection hardware specifications are shown in Table I.

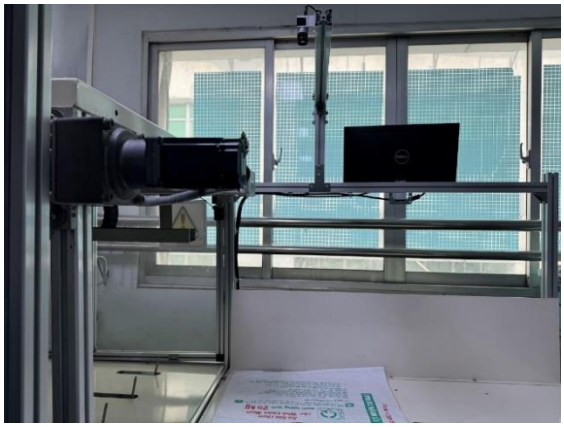

Fig. 2. Data collection system.

TABLE I. DATA COLLECTION HARDWARE SPECIFICATIONS

Hardware	Specifications
Camera	Model: Vieworks CMOS VC-25MC-M/C 30D Type: Area Camera
Lens	Type: 6 mm fixed focal lens Field of View: Wide
Computer	Intel(R) Core(TM) i7-11700K CPU + Nvidia RTX 4080 GPU

To address the issue of the rare occurrence of abnormal bags, the system is designed to flag and store images of any detected abnormalities for further review. This process helps in creating a focused dataset of abnormal bags, which, although smaller in size, is rich in diversity and critical for training the detection system effectively. Furthermore, to mitigate the imbalance between normal and abnormal bag instances, sophisticated data augmentation techniques are employed. This approach increases the representation of abnormal bags in the training dataset without the need for an equivalent increase in the actual occurrence of these abnormalities on the production line.

With the proposed data collection system, 1000 images have been collected. As in practical production, the operator is only concerned with whether the bag is an abnormal bag that needs to be discarded, not with the specific type of abnormality. Therefore, in this paper, the bags were classified into two categories, normal and abnormal. Normal bags are those that appear uniform in shape, with consistent dimensions and no visible defects on the surface. The integrity of each bag is maintained, and there are no signs of tears, misprints, or material weaknesses, as shown in Fig. 3(a). Abnormal bags, on the other hand, display various defects such as irregular shapes, tears, incorrect sealing, or material inconsistencies, as shown in Fig. 3(b). These abnormalities compromise the bag's functionality and potentially disrupt the operation of the machine, necessitating their removal from the production line.

B. Data Augmentation

Data augmentation plays a crucial role in the classification of normal and abnormal bags by significantly enhancing the diversity and volume of training data available for deep learning models. By artificially creating variations of the existing images through techniques such as rotation, noise injection, and perspective transformations, data augmentation helps models become more robust and less sensitive to small changes or imperfections in bag appearances. This process not only improves the model's ability to generalize across different conditions found in production environments but also addresses the challenge of limited samples of abnormal bags, thereby boosting the overall accuracy and reliability of the classification system. Broadly, data augmentation techniques can be categorized into two types: offline and online augmentations. Offline augmentation involves preprocessing and expanding the dataset before training begins. This means creating modified copies of the original images, such as rotated, flipped, or adjusted in terms of brightness and contrast, and adding them to the training set. This approach results in a statically enhanced dataset that the model trains on, allowing for a wide variety of data from the beginning. On the other hand, online augmentation takes place during the model training process itself. In this dynamic approach, images are augmented in real-time and fed into the model. This means that each epoch can present slightly different variations of the images to the model, introducing a richer set of examples over time. Techniques such as random cropping, zooming, or adding noise are applied in real-time, ensuring that the model rarely sees the exact same image twice. This not only improves generalization but also significantly enhances the model's robustness to new, unseen variations of bags.

a) Normal Bags

b) Abnormal Bags

Fig. 3. Examples of normal bags (a) and abnormal bags (b).

In this paper, we employ a comprehensive approach to data augmentation, leveraging both offline and online techniques to enhance the diversity and quality of our dataset for classifying normal and abnormal bags in automatic bagging machines. Initially, the dataset was expanded through offline augmentation methods, specifically focusing on brightness and contrast adjustments, noise injection, color variations, and synthetic abnormality generation. The latter is particularly noteworthy as it involves creating synthetic defects such as tears, holes, or significant shape distortions on images of normal bags. This method plays a crucial role in artificially expanding the collection of abnormal bag examples, avoiding the necessity for such events to happen naturally, and thus overcoming the lack of abnormal instances in the original dataset. Following the offline augmentation phase, we combined the enhanced images with the original ones to construct a new, enriched dataset comprising 3150 images. This dataset includes 1420 images depicting normal bags and 1730 images featuring abnormal bags, reflecting a more balanced distribution between the two categories. Subsequently, this augmented dataset was randomly divided into three subsets: a training set constituting 60% of the total images, a validation set comprising 20%, and a test set making up the remaining 20%. The distribution and details of the images within each subset are outlined in Table II.

TABLE II. NUMBER OF IMAGES IN EACH SUBSET

Subset	Number of images	
	Normal bags	Abnormal bags
Training	852	1038
Validation	284	346
Testing	284	346
Total	1420	1730

To further augment the diversity of data features available during model training, we implemented online augmentation techniques. During the training process, each image batch underwent preprocessing through a combination of methods, including color jittering to simulate varying lighting conditions and color schemes, Gaussian blurring to introduce variability in image sharpness and simulate minor camera focus issues, and random flipping (both horizontally and vertically) to ensure the model can accurately classify bags regardless of their orientation. This blend of online augmentation methods ensures that the model is exposed to a wide array of variations within the training data, significantly enhancing its ability to generalize from the training set to real-world scenarios where bags can appear under different conditions and with various types of abnormalities.

C. Lightweight Deep Model for Bags Classification

Since we use the Raspberry Pi 4 for classifying normal and abnormal bags in the automatic bagging machine, deploying a lightweight deep learning model on this board is necessary. The lightweight model is crucial because it is specifically designed to operate within the constrained computing resources and limited memory capacities typical of embedded systems. This model ensures that the classification process can be executed efficiently in real-time, maintaining high accuracy while minimizing latency, which is essential for integration into production lines. Furthermore, the optimized architecture of the model reduces power consumption, a critical consideration for continuous operation in industrial settings.

In our paper, we selected ResNet-18 as the foundation for our lightweight deep learning model to classify normal and abnormal bags, specifically designed for deployment on the Raspberry Pi 4. This choice was driven by ResNet-18's inherently efficient architecture that strikes an optimal balance between computational demand and model performance. We have conducted several modifications to ResNet-18, including changing the initial convolution layer based on Ghost Convolution [41] and reducing the number of channels, as well as employing knowledge distillation as a training technique to transfer knowledge from a fully trained and fine-tuned ResNet-18 model (teacher) to our optimized lightweight model (student). These enhancements further improve its suitability for real-time applications on hardware with limited computing resources. Compared to its deeper counterparts, such as ResNet-34 or ResNet-50, ResNet-18 offers a more practical solution for deployment on embedded systems like the Raspberry Pi 4. While deeper models might achieve slightly higher accuracy in certain contexts, their increased complexity and higher demand for computational resources make them less suitable for environments where power efficiency and low latency are paramount. The extensive use of Ghost Convolution, along with strategic knowledge distillation, allows our modified ResNet-18 model to maintain a competitive accuracy level while significantly reducing the necessary computational resources and power consumption. This balance is crucial for ensuring that the automatic bagging machine can operate continuously and efficiently in an industrial setting, making ResNet-18 the ideal choice for our application.

1) ResNet-18 architecture: ResNet-18 is a variant of the Residual Network (ResNet) architecture [10], designed to tackle the vanishing gradient problem that arises with increasing network depth. This architecture allows for the training of deep neural networks by introducing residual connections, which enable the flow of gradients through the network without degradation.

The architecture of ResNet-18, as shown in Fig. 4, consists of an initial convolutional layer followed by 16 convolutional layers organized into 8 residual blocks, and ends with an average pooling layer and a fully connected layer. The initial convolutional layer has a 7×7 kernel with 64 filters and a stride of 2. This layer is followed by a 3×3 max pooling layer with a stride of 2, which serves to reduce the spatial dimensions of the input image while preserving important features. Following the initial layers, ResNet-18 is composed of four main stages, each containing two residual blocks. Each block comprises two 3×3 convolutional layers with the same number of filters. The stages are differentiated by the number of filters and the stride of the first convolutional layer in each stage. Specifically, the stages have 64, 128, 256, and 512 filters, respectively. The stride is set to 1 for all blocks except the first block of each stage after the first, where it is set to 2. This design choice reduces the feature map's size as the network gets deeper, increasing the field of view of the convolutional filters.

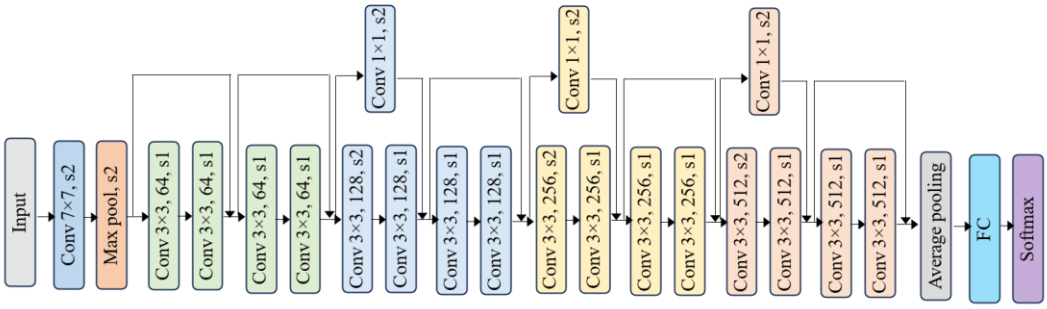

Fig. 4. ResNet-18 architecture.

The innovation of ResNet lies in its residual connections, which skip one or more layers by performing identity mapping and adding the input of the block to its output. These connections help mitigate the vanishing gradient problem by allowing the direct flow of gradients during backpropagation. In ResNet-18, every two layers share a residual connection, forming the backbone of its design. At the end of the network, a global average pooling layer reduces the spatial dimensions to 1×1, effectively summarizing the features extracted by the convolutions into a compact form. This is followed by a fully connected layer, which maps the pooled features to the desired number of output classes, facilitated by a Softmax activation function for classification tasks.

2) Improving initial convolutional layer: To address the high latency associated with the initial convolutional layer in ResNet-18, which is primarily due to its 7×7 convolution with a stride of 2, a modification is proposed to enhance computational efficiency while maintaining the layer's feature extraction capability. This modification involves increasing the stride of the initial convolutional layer to 4 and decreasing the kernel size from 7×7 to 5×5. The rationale behind this is twofold: a larger stride and a smaller kernel size directly reduce the amount of computation required, thereby lowering the latency. Furthermore, to compensate for the potential loss of feature extraction capability due to these reductions, a Ghost Convolution layer [41] is introduced right after the modified 5×5 convolutional layer. Ghost Convolution, as shown in Fig. 5, is a novel neural network architecture optimization technique designed to significantly reduce the computational cost and model size while preserving, or even enhancing, the model's performance. It achieves this by generating additional ghost feature maps from a smaller number of primary feature maps using inexpensive operations, thus efficiently utilizing computational resources. Serving as a pivotal innovation in deep learning, Ghost Convolution plays a crucial role in enabling more efficient and faster neural networks, particularly beneficial for deployment in resource-constrained environments such as mobile devices and edge computing platforms.

Building on the innovative approach outlined above, Table III presents a comparison of FLOPs before and after the modification of the initial convolutional layer to quantify the efficiency gains achieved through our modification. By adjusting the kernel size and stride, and by adding a Ghost Convolution layer, the initial convolutional layer achieves fewer FLOPs compared to the original convolutional layer. Through this strategy, the modified initial layer of ResNet-18 offers reduced latency without substantially compromising the network's performance, making it more suitable for real-time applications or deployment on hardware with limited computational resources such as the Raspberry Pi 4.

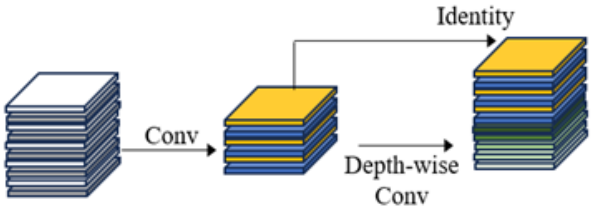

Fig. 5. Ghost convolution structure.

TABLE III. COMPARISON OF FLOPS BEFORE AND AFTER THE MODIFICATION OF THE INITIAL CONVOLUTIONAL LAYER

Configuration	Kernel Size	Stride	Additional Layers	FLOPs (Billions)
Original initial layer	7×7	2	0	1.8
Modified initial layer	5×5	4	1	1.6

3) Modifying channel numbers: In CNNs, the number of channels typically increases as the network progresses deeper. This design strategy originates from the need to capture increasingly complex features from the input data. Initially, layers detect simple patterns and textures, such as edges and colors. As we move deeper into the network, subsequent layers combine these basic features to detect more complex and abstract features, necessitating a larger number of channels to represent this growing complexity effectively.

In the case of ResNet-18, the structure follows this principle closely. The network starts with an initial convolutional layer that has 64 channels. This is followed by four main stages, each comprising two residual blocks. The channel numbers for each stage double as the network goes deeper: the first stage has 64 channels per layer, the second stage has 128 channels, the third stage has 256 channels, and the final stage has 512 channels. This design enables the network to process and extract a rich hierarchy of features from

the input images. However, not all tasks require the full capacity of ResNet-18. For simpler tasks, such as classifying bags as normal or abnormal, the complexity of the model can be reduced without significantly impacting performance. This simplification can lead to gains in efficiency, making the network more suitable for deployment in environments with limited computational resources like the Raspberry Pi 4. Based on extensive experiments, we have found that adjusting the number of channels in each layer of ResNet-18 to 64, 96, 128, and 160 for the respective stages strikes a good balance between performance and computational efficiency for this specific task. Table IV provides a comparison of FLOPs before and after reducing the number of channels in each layer. The results show that modifying the channel numbers significantly reduces the FLOPs of each stage.

TABLE IV. COMPARISON OF FLOPS BEFORE AND AFTER REDUCING THE NUMBER OF CHANNELS IN EACH LAYER

Stage	Original		Modified	
	Channels	FLOPs (Billions)	Channels	FLOPs (Billions)
2	64	0.35	64	0.35
3	128	0.55	96	0.4
4	256	0.4	128	0.3
5	512	0.25	160	0.2
FC layer	-	0.1	-	0.05
Total		1.65		1.3

Reducing the number of channels across the layers of ResNet-18 not only improves the model's computational efficiency, reflected in lower FLOPs, but also reduces the latency of the network during inference. This modification, however, comes at the cost of reduced network capacity. The term capacity here refers to the model's ability to learn from data; a higher capacity enables a network to capture more complex patterns but may also increase the risk of overfitting and require more data to train effectively. For the task of classifying normal and abnormal bags, which is relatively simple, this reduced capacity does not significantly hinder performance and leads to a more efficient model suitable for real-time applications or deployment on hardware with limited computational resources.

4) Knowledge distillation: Knowledge distillation is a powerful technique that can significantly improve the efficiency and accuracy of deploying the proposed model on the Raspberry Pi 4, especially for tasks of classifying normal and abnormal bags in an automatic bagging machine. The essence of knowledge distillation lies in transferring the knowledge from a large, complex teacher model to a smaller, more computationally efficient student model. This process allows the student model to learn the complex decision boundaries and the detailed representations captured by the teacher, without the need for extensive computational resources. In this paper, the teacher model is the fully trained and fine-tuned ResNet-18 model, which has been enhanced for better performance on the task of bag classification. The student model, on the other hand, is the simplified version of ResNet-18 with modifications to lower its computational demands. The distillation process involves running the dataset through both the teacher and student models, using the output probabilities (soft targets) of the teacher model as a guide for training the student model. These soft targets provide richer information compared to hard labels (normal/abnormal), as they contain insights about how the teacher model perceives the differences between classes, including the uncertainty and the relationships among them. To implement knowledge distillation effectively, we use a loss function that combines a traditional classification loss (i.e., cross-entropy against the true labels) with a distillation loss that measures the discrepancy between the teacher's predictions and the student's predictions. The distillation loss employs a temperature parameter to soften the probability distributions, making it easier for the student model to learn from the teacher's outputs. By employing knowledge distillation for the proposed model targeted for deployment on the Raspberry Pi 4, we can reduce model size and computational requirements while improving accuracy and enhancing inference speed.

5) Overall architecture of the proposed model: The overall architecture of the proposed model is shown in Fig. 6. It incorporates several modifications to ResNet-18 to enhance performance while accommodating the computational limitations of embedded systems. Initially, the model introduces a modified initial convolutional layer, where the stride is increased to 4 and the kernel size is decreased from 7×7 to 5×5. This modification aims to capture finer details of input images without excessively burdening the Raspberry Pi's computational resources. Following the initial convolutional layer, a Ghost Convolution layer is introduced. This layer plays a pivotal role in reducing the model's complexity by generating more feature maps from fewer parameters, thus efficiently enhancing the representational capacity without a substantial increase in computational demand. The core of the model is composed of successive Residual Blocks (Res blocks), specifically arranged to progressively refine the feature maps. The number of channels in each layer of ResNet-18 has been adjusted to 64, 96, 128, and 160 for the respective stages, optimizing the balance between computational efficiency and the model's ability to capture relevant features from the image data. This adjustment ensures that the model remains lightweight yet capable of processing the varying complexities of the bag images through its depth. Each Res block employs a combination of 3×3 convolutions, with some blocks incorporating a stride of 2 (denoted as 3×3 conv, s2) to reduce the dimensionality and focus the model's attention on salient features. At the end of the model, a Global Average Pooling layer is utilized to condense the feature maps into a form suitable for classification, effectively reducing the dimensionality and focusing the model's output. This is followed by a Fully Connected (FC) layer that makes the final decision, classifying the input image as either normal or abnormal. Moreover, we apply knowledge distillation as a training technique to transfer knowledge from a fully trained

and fine-tuned ResNet-18 model (teacher) to our optimized lightweight model (student). This approach allows the lightweight model to achieve higher accuracy by learning refined representations and decision boundaries, effectively mimicking the performance of the more cumbersome teacher model without the associated computational overhead.

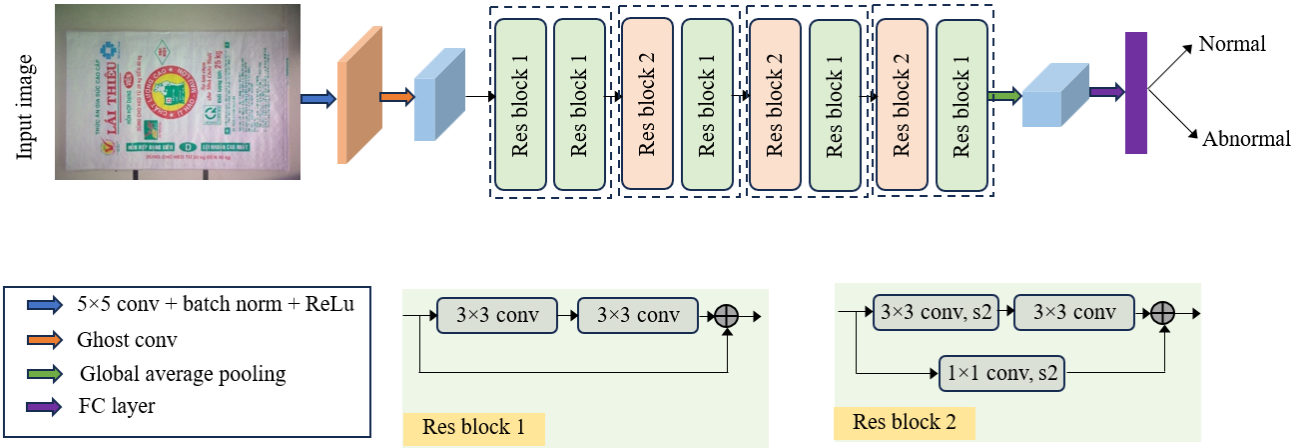

Fig. 6. Overall architecture of the proposed model.

IV. Experiments and Results

In this section, we first introduce a detailed description of the implementation setup and the evaluation metrics used to evaluate the performance of our model. Following this, we proceed into a comprehensive analysis of the results, drawing comparisons between the proposed model and both classical CNNs and existing lightweight neural network architectures. This comparison highlights the strengths and efficiencies of our model. Subsequently, we conduct an ablation study to evaluate the impact of various modules and modifications within our architecture, providing insight into their individual contributions towards the model's overall performance. Lastly, we discuss the deployment of our optimized model on the Raspberry Pi 4 within an actual automatic bagging machine setup, demonstrating its practical application and effectiveness in a real-world scenario.

A. Implementation Setup

For the training of our lightweight network designed to classify normal and abnormal bags in an automatic bagging machine, we employed a high-performance computing setup. The network was trained on a system equipped with an Intel(R) Core(TM) i7-11700K CPU, 32GB of RAM, and an Nvidia RTX 4080 GPU. This hardware configuration, supported by the CUDA 10.1 Toolkit, provided the necessary computational power to effectively train our model using TensorFlow, a popular deep learning framework known for its flexibility and extensive support for CNNs.

The training process lasted for 50 epochs. A batch size of 64 was chosen to balance the trade-off between memory usage and the granularity of the gradient update, ensuring efficient use of the GPU's resources. For optimization, we employed the Stochastic Gradient Descent (SGD) algorithm with a momentum of 0.9 and a weight decay of 0.0001. The initial learning rate was set to 0.01. The learning rate was scheduled to decrease after certain epochs based on performance metrics on the validation set, helping the model to fine-tune its weights more precisely as training progressed. The loss function chosen for this task was cross-entropy, a common choice for classification problems as it quantifies the difference between the predicted probabilities and the actual distribution, driving the model to make more accurate predictions over time.

B. Evaluation Metrics

In the task of classifying normal and abnormal bags, evaluating the performance of the model accurately is crucial to ensure its effectiveness in real-world applications. To achieve this, we employ several evaluation metrics, including accuracy, precision, recall, number of parameters, and FLOPs (Floating Point Operations), each offering unique insights into the model's capabilities and areas for improvement.

Accuracy is the simplest and most intuitive metric, representing the proportion of correctly classified instances (both normal and abnormal bags) to the total number of instances in the dataset. It is calculated using the following formula:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

where, TP (True Positives) and TN (True Negatives) are the correctly identified abnormal and normal bags, respectively, while FP (False Positives) and FN (False Negatives) represent the incorrectly classified instances.

Precision, or positive predictive value, measures the proportion of correctly identified abnormal bags out of all bags predicted as abnormal. This metric is particularly important in scenarios where the cost of falsely identifying a bag as abnormal (when it is not) is high. Precision is defined as the following formula:

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Recall, also known as sensitivity, indicates the proportion of actual abnormal bags that were correctly identified by the model. High recall is essential in ensuring that as many

abnormal bags as possible are detected. The formula for recall is defined as the following formula:

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

In addition to these classification metrics, we also evaluate the model's complexity and efficiency using parameters and FLOPs. Parameters refer to the total number of trainable weights in the model. A lower count indicates a more lightweight model, which is beneficial for deployment on devices with limited computational resources, such as the Raspberry Pi 4. On the other hand, FLOPs provide a measure of the computational workload associated with a single forward pass through the model. It is calculated by adding up all the floating-point operations (additions, multiplications, etc.) involved in generating a prediction.

C. Main Results

1) Comparison with other CNNs models: In our study, we compared the performance of our proposed model against a range of established CNN architectures, including MobileNet-V1, MobileNet-V3, ResNet-50, ShuffleNet-V2, and VGG16 on the test dataset. This comparative analysis aimed to benchmark our model's effectiveness in classifying normal and abnormal bags against these well-known CNNs, with a particular focus on the balance between accuracy and computational efficiency as reflected in the model's precision, recall, parameters, and FLOPs. The comparison results are shown in Table V. The results show that our proposed model significantly outperforms the other architectures in terms of accuracy, achieving a remarkable 93.5%. This indicates a superior ability to correctly classify bags, which is critical in practical applications where misclassification can lead to operational inefficiencies or quality control issues. In terms of precision and recall, our model also leads with scores of 93.0% and 94.0%, respectively. These metrics suggest not only a high rate of correctly identifying abnormal bags but also an impressive capability to detect the majority of actual abnormal bags present in the dataset.

Despite its high performance, the proposed model maintains a moderate number of parameters and FLOPs, illustrating an efficient balance between computational cost and effectiveness. Notably, VGG16, with the largest model size and the highest computational cost, demonstrates lower performance metrics, highlighting the inefficiency of larger models in terms of computational resources versus accuracy gain. On the other hand, MobileNet-V3 and ShuffleNet-V2, known for their high efficiency, show remarkable performance with significantly fewer parameters and FLOPs, emphasizing the effectiveness of architectures designed for operational efficiency. ResNet-50, while having a substantial number of parameters and FLOPs, offers competitive performance metrics, which highlights the balance it strikes between depth and computational efficiency. However, our proposed model's performance, with its comparatively modest computational requirements, suggests that careful optimization and architectural choices can result in models that not only achieve superior accuracy but are also viable for deployment in resource-constrained environments. The results emphasize the importance of optimizing for both model size and computational efficiency without compromising on the task-specific performance, a key consideration for practical applications, especially in environments with limited computational resources like the Raspberry Pi 4.

TABLE V. COMPARISON RESULTS ON THE TEST DATASET OF DIFFERENT MODELS

Model	Accuracy (%)	Precision (%)	Recall (%)	Parameters (Millions)	FLOPs (Billions)
MobileNet-V1	89.5	88.7	90.2	4.2	0.57
MobileNet-V3	91.0	90.5	91.5	2.9	0.22
ResNet-50	92.3	91.8	92.7	25.6	4.1
ShuffleNet-V2	90.2	89.9	90.6	2.3	0.15
VGG16	88.0	87.5	88.5	138	15.5
Our Proposed Model	93.5	93.0	94.0	4.5	1.4

TABLE VI. CLASSIFICATION RESULTS OF DIFFERENT COMBINATIONS ON THE VALIDATION DATASET

Model	Accuracy (%)	Precision (%)	Recall (%)	Parameters (Millions)	FLOPs (Billions)
Original ResNet-18	89.1	88.6	89.2	11.6	1.8
ResNet-18 + IICL	91.6	91.2	91.8	8.4	1.6
ResNet-18 + IICL + MCN	91.4	90.9	92.9	4.5	1.4
ResNet-18 + IICL + MCN + KD (Our Proposed Model)	94.2	93.7	94.3	4.5	1.4

2) Ablation analysis: To illustrate the impact of various modifications and modules on the performance of our architecture, an ablation study was conducted comparing different variants of the ResNet-18 model on the validation dataset. These variants include the original ResNet-18 model, ResNet-18 with an improved initial convolutional layer (IICL), ResNet-18 with IICL and modified channel numbers (MCN), and finally, the complete proposed model which also incorporates knowledge distillation (KD). The comparison results are shown in Table VI. Starting with the original

ResNet-18, we observe a solid baseline with an accuracy of 89.1%, precision of 88.6%, and recall of 89.2%. This model, despite its relatively high computational cost (11.6 million parameters and 1.8 billion FLOPs), sets a foundation for further optimization. The introduction of an IICL, which includes increasing the stride and decreasing the kernel size, along with the addition of a Ghost Convolution layer, significantly boosts performance. This first modification enhances the model's efficiency in feature extraction and reduces computational requirements, resulting in improved accuracy (91.6%), precision (91.2%), and recall (91.8%), with a notable reduction in parameters (8.4 million) and FLOPs (1.6 billion). Interestingly, the third variant, which combines IICL with MCN, shows a slight decrease in accuracy and precision but a notable increase in recall (92.9%). This suggests that adjusting the number of channels effectively improves the model's sensitivity in detecting abnormal bags, a critical aspect of the classification task. This modification also significantly lowers the computational cost, halving the parameters to 4.5 million and reducing FLOPs to 1.4 billion, indicating a substantial increase in efficiency. Our proposed model, which further incorporates KD alongside IICL and MCN, achieves the best performance across all metrics: accuracy (94.2%), precision (93.7%), and recall (94.3%). This impressive improvement is achieved with the lowest complexity, featuring only 4.5 million parameters and 1.4 billion FLOPs. The addition of KD allows the model to learn more refined representations and decision boundaries, which is evident in its superior performance metrics. This indicates that knowledge distillation is highly effective in enhancing model performance, especially in tasks requiring high precision and recall.

Overall, the ablation study demonstrates the effectiveness of our targeted modifications in not only improving the model's accuracy, precision, and recall but also in significantly enhancing its computational efficiency. The final proposed model stands out as highly optimized for the specific task of classifying bags, making it ideal for deployment in resource-constrained environments such as the Raspberry Pi 4, where efficiency and performance are paramount.

3) Heatmap visualization: Fig. 7 visualizes heatmap results from the last convolution layer of the last block of the proposed model. The heatmaps offer valuable insights into the regions within the images that the proposed model focuses on when making classifications. These heatmaps are derived from the last convolutional layer of the last block of the model, highlighting the areas with the highest activations, typically the regions most significant for the model's decision-making process.

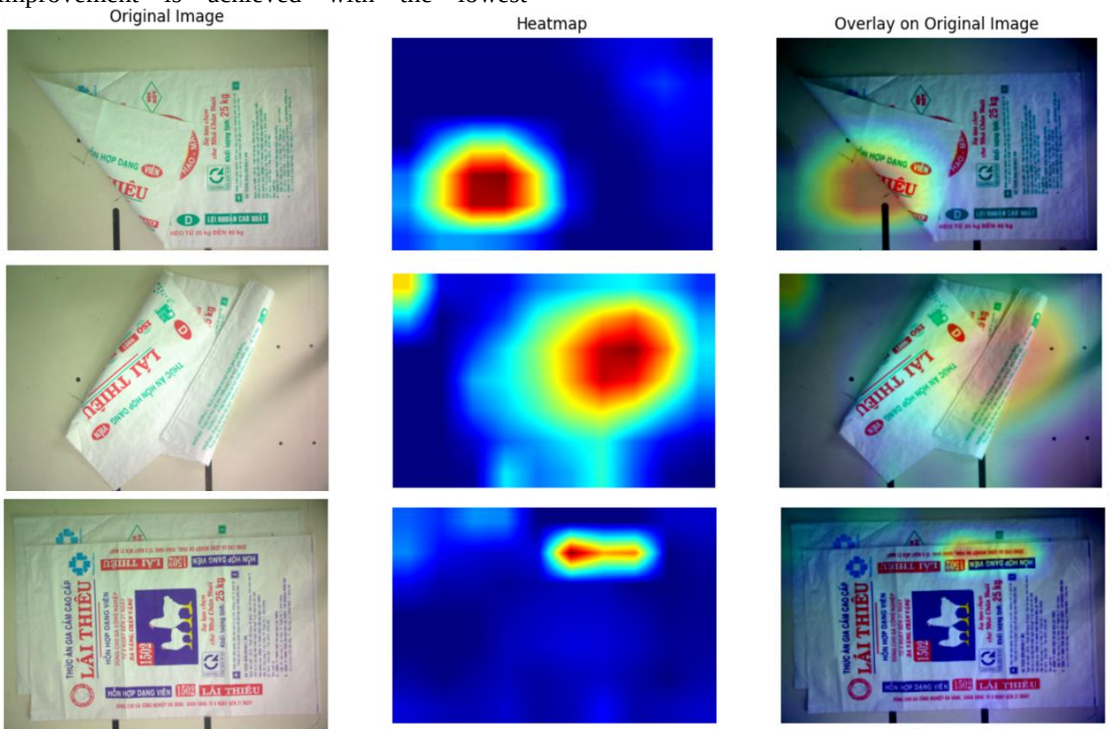

Fig. 7. Heatmap visualization of the proposed model.

In the first row, the heatmap is intensely focused on the area where part of the bag is missing, indicating that the model identifies this as the most informative region for classifying the bag as normal or abnormal. The high-intensity area represents a strong response, which suggests a distinctive feature or pattern that the model has learned to recognize as indicative of the bag's status. In the second row, we see a similar pattern of focus. The model's attention is concentrated around the center of the bag, but with a more elongated spread along the vertical axis. This reflects the model's detection of abnormalities that have a more linear orientation or a change in the bag's texture or structure in that specific region. The third row presents a

narrower focus in the heatmap. The activation is concentrated in a smaller region, which suggests that the model has detected a very specific feature of interest that is highly relevant to the classification task. The tight clustering of high activation corresponds to a localized abnormality.

In summary, these heatmaps provide a clear representation of where the model is looking and what it considers critical for its classification decisions. The results show that the model is not distracted by the periphery of the images and consistently focuses on the central parts of the bags, where textual and logo features are most prominent. This consistent focus across different bags suggests that the model has learned a generalized understanding of where relevant features are likely to appear. Crucially, these heatmaps can be used not only to understand the model's behavior but also to validate whether the model is considering the right features when making a classification. If the model were focusing on irrelevant areas, it might indicate an overfitting to noise or a misalignment in the learned features. However, the heatmaps in this figure confirm the model's appropriate focus areas, thus supporting its reliability and robustness in identifying normal and abnormal bags.

4) Deployment: For deploying the proposed model from a powerful training environment to a resource-constrained platform in the Raspberry Pi 4, we perform several steps to ensure that the model retains its accuracy and efficiency in a production setting. The first step is to convert the model into a TensorFlow Lite model. TensorFlow Lite is a set of tools that enables on-device machine learning by optimizing the TensorFlow model for performance on lightweight devices. By converting the model to TensorFlow Lite model, we substantially reduce its size while maintaining critical aspects of its performance. The converted model is further optimized using post-training quantization technique, which not only decreases the model's size but also potentially speeds up inference times by using lower-precision calculations. This is particularly important for the Raspberry Pi 4, as it has less computational power and memory compared to the original training environment.

Following optimization, the TensorFlow Lite model is deployed onto the Raspberry Pi 4, which involves transferring the model file to the device and setting up the necessary inference libraries. Once in place, the model is integrated into the automatic bagging machine's control software, where it will process input images captured by the machine's cameras in real-time. The software preprocesses the input data according to the model's requirements, then feeds it into the model for inference. The inference libraries, optimized for the Raspberry Pi 4's ARM architecture, facilitate the execution of the model efficiently, ensuring that classification decisions are made swiftly to keep pace with the operational speed of the bagging machine. The lightweight nature of the TensorFlow Lite model allows for rapid inference, which is essential for the model to be practical in a production environment.

Finally, we conduct extensive testing to confirm that the model's performance on the Raspberry Pi 4 aligns with the results observed during its initial development and validation. This involves monitoring accuracy, speed of inference, and resource utilization under real-world conditions. Table VII provides results of the deployment of the proposed model on the Raspberry Pi 4. The results in Table VII indicate that the proposed model delivers a solid performance in an operational environment. An inference time of 500 ms per bag suggests that the model is performing real-time analysis at a viable speed for the automatic bagging machine, considering the balance between speed and the complexity of the task. Power consumption stands at 4 W, which is a testament to the Raspberry Pi's energy efficiency and the lightweight nature of the optimized TensorFlow Lite model. Such low power draw is ideal for continuous, long-term operation in industrial settings. The CPU utilization of 48% indicates that the model is utilizing less than half of the CPU's capabilities, which is significant as it leaves room for the Raspberry Pi 4 to handle other tasks concurrently, if necessary, without overloading the system. The memory usage is moderate at 350 MB, which falls well within the Raspberry Pi 4's RAM capabilities, ensuring that the model runs smoothly without memory bottlenecks. This level of resource usage supports the notion that the model is indeed lightweight and suitable for embedded systems. The model's accuracy, precision, and recall rates are exceptionally high at 94.2%, 93.7%, and 94.3%, respectively. These metrics almost mirror the performance during the training phase, which indicates a successful model optimization and conversion process with negligible loss in model efficacy. Such high values suggest that the model is highly reliable, making correct decisions most of the time, and is able to identify the majority of abnormal bags correctly. An inference throughput of two bags per second may seem modest but is generally sufficient for automatic bagging operations, suitable for the speed of the conveyor and the number of bags processed in a given timeframe.

TABLE VII. RESULTS OF THE DEPLOYMENT OF THE PROPOSED MODEL ON THE RASPBERRY PI 4

Metric	Value	Comments
Inference time	500 ms	Time taken for a single inference
Power consumption	4 W	Average power during model inference
CPU utilization	48%	CPU usage during model inference
Memory usage	350 MB	RAM used by the model during operation
Model accuracy	94.2%	Percentage of correctly classified bags
Model precision	93.7%	Proportion of true positives over total positives
Model recall	94.3%	Proportion of true positives over actual positives
Inference throughput	2 bags/sec	Number of bags classified per second

V. CONCLUSION

In conclusion, our research has successfully demonstrated the efficacy of a deep learning and computer vision-based system designed for the classification of normal and abnormal bags within an automatic bagging machine. Through the strategic integration of a sophisticated data collection system directly on the production line, we have created a rich dataset that accurately reflects the variability inherent in real-world manufacturing processes. The implementation of both offline

and online data augmentation methods has significantly enhanced the robustness of our dataset, preparing our model to handle diverse operational scenarios. Our modifications of the ResNet-18 architecture into a lightweight deep learning model have proven to be particularly well-suited for deployment on the resource-limited Raspberry Pi 4, maintaining high accuracy and efficiency in bag classification tasks. The extensive comparative analysis with other CNN models and the thorough ablation studies have underscored the advantages of our proposed model. Overall, the contributions of this work not only lie in the novel application of a deep learning-based approach to a specific industrial challenge but also in the advancement of deploying complex models to edge devices. The success of this project opens avenues for future research into similar applications across different sectors, fostering the integration of AI in industrial automation.

ACKNOWLEDGMENT

This research is funded by Industrial University of Ho Chi Minh City under grant number 23.1CND01 (Contract number 135/HĐ-ĐHCN).

REFERENCES

[1] Jung, Byeonggil, Heegon You, and Sangkyun Lee. "Anomaly Candidate Extraction and Detection for automatic quality inspection of metal casting products using high-resolution images." *Journal of Manufacturing Systems* 67 (2023): 229-241.

[2] Chen, Bingsheng, Huijie Chen, and Mengshan Li. "Automatic quality inspection system for discrete manufacturing based on the Internet of Things." *Computers and Electrical Engineering* 95 (2021): 107435.

[3] Hakami, Aisha, and Mohammed Arif. "Automatic inspection of the external quality of the date fruit." *Procedia Computer Science* 163 (2019): 70-77.

[4] Kawamura, Shuso, Motoyasu Natsuga, Kazuhiro Takekura, and Kazuhiko Itoh. "Development of an automatic rice-quality inspection system." *Computers and electronics in agriculture* 40, no. 1-3 (2003): 115-126.

[5] Xia, Jiaping, YuHyeong Jeong, and Jonghun Yoon. "An automatic machine vision-based algorithm for inspection of hardwood flooring defects during manufacturing." *Engineering Applications of Artificial Intelligence* 123 (2023): 106268.

[6] LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86, no. 11 (1998): 2278-2324.

[7] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012).

[8] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[9] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1-9. 2015.

[10] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

[11] Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).

[12] Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. "Mobilenetv2: Inverted residuals and linear bottlenecks." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510-4520. 2018.

[13] Howard, Andrew, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang et al. "Searching for mobilenetv3." In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314-1324. 2019.

[14] Zhang, Xiangyu, Xinyu Zhou, Mengxiao Lin, and Jian Sun. "Shufflenet: An extremely efficient convolutional neural network for mobile devices." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848-6856. 2018.

[15] Ma, Ningning, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. "Shufflenet v2: Practical guidelines for efficient cnn architecture design." In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116-131. 2018.

[16] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." In *International conference on machine learning*, pp. 6105-6114. PMLR, 2019.

[17] Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).

[18] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

[19] Fayyaz, Mohsen, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. "Adaptive token sampling for efficient vision transformers." In *European Conference on Computer Vision*, pp. 396-414. Cham: Springer Nature Switzerland, 2022.

[20] Pan, Bowen, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogerio Feris, and Aude Oliva. "IA-RED $^ 2$: Interpretability-Aware Redundancy Reduction for Vision Transformers." *Advances in Neural Information Processing Systems* 34 (2021): 24898-24911.

[21] Rao, Yongming, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. "Dynamicvit: Efficient vision transformers with dynamic token sparsification." *Advances in neural information processing systems* 34 (2021): 13937-13949.

[22] Xu, Yifan, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. "Evo-vit: Slow-fast token evolution for dynamic vision transformer." In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, pp. 2964-2972. 2022.

[23] Zhang, Jinnian, Houwen Peng, Kan Wu, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. "Minivit: Compressing vision transformers with weight multiplexing." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12145-12154. 2022.

[24] Touvron, Hugo, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. "Training data-efficient image transformers & distillation through attention." In *International conference on machine learning*, pp. 10347-10357. PMLR, 2021.

[25] Child, Rewon, Scott Gray, Alec Radford, and Ilya Sutskever. "Generating long sequences with sparse transformers." *arXiv preprint arXiv:1904.10509* (2019).

[26] Katharopoulos, Angelos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. "Transformers are rnns: Fast autoregressive transformers with linear attention." In *International conference on machine learning*, pp. 5156-5165. PMLR, 2020.

[27] Kitaev, Nikita, Łukasz Kaiser, and Anselm Levskaya. "Reformer: The efficient transformer." *arXiv preprint arXiv:2001.04451* (2020).

[28] Chen, Chun-Fu, Rameswar Panda, and Quanfu Fan. "Regionvit: Regional-to-local attention for vision transformers." *arXiv preprint arXiv:2106.02689* (2021).

[29] Choromanski, Krzysztof, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins et al. "Rethinking attention with performers." *arXiv preprint arXiv:2009.14794* (2020).

[30] Chu, Xiangxiang, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. "Twins: Revisiting the

design of spatial attention in vision transformers." *Advances in neural information processing systems* 34 (2021): 9355-9366.

[31] Xu, Wenguang, Pengcheng Xiao, Liguang Zhu, Yan Zhang, Jinbao Chang, Rong Zhu, and Yunfeng Xu. "Classification and rating of steel scrap using deep learning." *Engineering Applications of Artificial Intelligence* 123 (2023): 106241.

[32] Nath, Vikanksh, Chiranjoy Chattopadhyay, and K. A. Desai. "NSLNet: An improved deep learning model for steel surface defect classification utilizing small training datasets." *Manufacturing Letters* 35 (2023): 39-42.

[33] Juncker, Mathieu, Ismaïl Khriss, Jean Brousseau, Steven Pigeon, Alexis Darisse, and Billy Lapointe. "A Deep Learning-Based Approach for Quality Control and Defect Detection for Industrial Bagging Systems." In *2020 IEEE 19th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pp. 60-67. IEEE, 2020.

[34] Padmapriya, J., and T. Sasilatha. "Deep learning based multi-labelled soil classification and empirical estimation toward sustainable agriculture." *Engineering Applications of Artificial Intelligence* 119 (2023): 105690.

[35] Gill, Harmandeep Singh, G. Murugesan, Abolfazi Mehbodniya, Guna Sekhar Sajja, Gaurav Gupta, and Abhishek Bhatt. "Fruit type classification using deep learning and feature fusion." *Computers and Electronics in Agriculture* 211 (2023): 107990.

[36] Shewale, Mitali V., and Rohin D. Daruwala. "High performance deep learning architecture for early detection and classification of plant leaf disease." *Journal of Agriculture and Food Research* 14 (2023): 100675.

[37] Han, Yifei, Zhaojing Liu, Kourosh Khoshelham, and Shahla Hosseini Bai. "Quality estimation of nuts using deep learning classification of hyperspectral imagery." *Computers and Electronics in Agriculture* 180 (2021): 105868.

[38] Kazi, Aafreen, and Siba Prasada Panda. "Determining the freshness of fruits in the food industry by image classification using transfer learning." *Multimedia Tools and Applications* 81, no. 6 (2022): 7611-7624.

[39] Huang, Yanqing, Junfeng Jing, and Zhen Wang. "Fabric defect segmentation method based on deep learning." *IEEE Transactions on Instrumentation and Measurement* 70 (2021): 1-15.

[40] Wei, Bing, Kuangrong Hao, Lei Gao, and Xue-Song Tang. "Bioinspired visual-integrated model for multilabel classification of textile defect images." *IEEE Transactions on Cognitive and Developmental Systems* 13, no. 3 (2020): 503-513.

[41] Han, Kai, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. "Ghostnet: More features from cheap operations." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1580-1589. 2020.