Research on Bottle Recognition Algorithm Based on Transfer Learning

Tian Jipeng[1], Hou Lingmei[2], T.C. Manjunath[3], G. Hemantha Kumar[4]
[1] *(Computer Science College, Zhongyuan University of Technology, Zhengzhou, Henan, China)*
[2,4] *(DoS University of Mysore, Mysore, Karnataka, India)*
[3] *(Dept. of Dayananda Sagar College of Engg., Bangalore, Karnataka, India.)*
Corresponding Author: Tian Jipeng[1]

Received 08 December 2023; Accepted 23 December2023

Abstract: This report explores transfer learning-based bottle recognition algorithms, aiming to improve the performance of bottle recognition tasks by utilizing pre trained model knowledge on large-scale datasets. We adopt the idea of transfer learning and apply deep learning models trained on image classification tasks to bottle recognition tasks, achieving more efficient feature extraction and model training.

Keywords: Transfer Learning, Recognition, Bottle, Deep Learning.

I. INTRODUCTION

Bottle recognition has a wide range of applications in industries, retail, and environmental monitoring. However, due to the limitations of the dataset and the complexity of specific bottle recognition tasks, traditional models may face performance issues. Therefore, we introduce the method of transfer learning in an attempt to improve the accuracy and generalization of bottle recognition by utilizing the knowledge learned from previous tasks. The background of bottle classification and recognition research stems from the growing global garbage problem and concern for environmental sustainability. With urbanization and the improvement of consumption levels, bottle waste disposal has become an important issue. The advancement of computer vision and machine learning provides new solutions for garbage classification, which can improve bottle recycling efficiency and reduce negative impacts on the environment through automatic recognition and classification. This study will help promote the development of intelligent bottle management systems and encourage society to pay more attention to sustainable development and resource recycling.

Bottle tracking and positioning have multiple meanings in different scenarios, depending on the specific application field. Here are some possible meanings:

a) Production and logistics management: In the production line or logistics process, bottle tracking and positioning can help monitor the manufacturing, assembly, and transportation status of bottles. This helps to improve production efficiency, reduce losses, and optimize logistics operations.
b) Inventory management: In retail environments or warehousing, tracking the position of bottles helps to monitor inventory levels in real time, avoid excess or shortage, and improve the efficiency of inventory management.
c) Quality control: By tracking the bottles, quality control measures can be implemented. Monitor the manufacturing process of bottles, detect possible defects or damages, to ensure the quality of the final product.
d) Environmental monitoring: In environmental monitoring, bottle tracking and positioning can be used to track the flow of liquids or chemicals, helping to monitor the results of environmental pollution or certain chemical reactions.
e) Marketing and User Experience: In retail, tracking bottles can be used to understand customer purchasing habits, analyze the flow of products in the store, thereby optimizing product placement and enhancing the shopping experience.
f) Environmental protection and sustainable development: Bottle tracking and positioning can help achieve effective resource utilization. For example, in the recycling cycle, tracking the source and destination of bottles can improve the recycling rate and promote sustainable development.
g) Traceability and Safety: In the food and beverage industry, bottle tracking helps to implement product traceability. This is crucial for detecting potential health and safety issues, improving product quality and safety.

Overall, the significance of bottle tracking and positioning lies in providing real-time and accurate information, helping to optimize business processes, improve efficiency, ensure quality, and meet regulatory and market demands. This has potential value for multiple industries, especially in modern business environments that emphasize efficiency, traceability, and sustainability.

Recognition and classification algorithms refer to algorithms used for automatic recognition and classification of objects or data. These algorithms have been widely applied in fields such as computer vision and natural language processing. Common recognition and classification algorithms include:

I. Support Vector Machines (SVM): used for binary and multivariate classification problems, by finding the optimal decision boundary in the feature space for classification. Support Vector Machine (SVM) is a supervised learning algorithm used for classification and regression analysis. It establishes an optimal separating hyperplane by finding the support vector in the dataset, which is the sample point closest to the decision boundary. This hyperplane can achieve maximum spacing between different categories, thereby improving the model's generalization ability.

II. Decision Trees: Based on stepwise judgment and segmentation of data, generate a tree like structure for classification. Decision tree is a supervised learning algorithm used for classification and regression. It represents the decision-making process in the form of a tree diagram, with each node representing a feature, each branch representing a decision rule, and each leaf node representing an output result. During the training process, the decision tree gradually splits based on the characteristics of the data, selecting the best features to achieve the best classification or regression results. Decision trees have the advantage of being easy to understand and interpret, but they are also prone to overfitting the number of training sessions.

III. Random Forest: an ensemble learning model composed of multiple decision trees, used to improve classification accuracy and robustness. Random forest is an ensemble learning method constructed based on decision trees. It improves the overall performance and robustness of the model by combining multiple decision trees. In a random forest, each decision tree is a base learner that trains different subsets of data and uses random feature selection to increase diversity. Finally, by voting or averaging, integrate the outputs of all decision trees to obtain the final prediction result. Random forests typically perform well in processing large amounts of data and high-dimensional features, and have the ability to resist overfitting.

IV. Neural Networks: A commonly used algorithm in deep learning that learns and classifies through multi-layer neural networks. Neural network is a machine learning model that simulates the structure and function of the human brain's nervous system. It consists of neurons and their connections, including the input layer, hidden layer, and output layer. Neural networks perform tasks such as classification, regression, clustering, etc. by learning patterns of data. Deep Neural Networks (DNN) are an extension of neural networks with multiple hidden layers. Deep learning is a branch of machine learning that utilizes deep neural networks for learning and inference. Neural networks have made significant achievements in fields such as image recognition, speech recognition, and natural language processing.

V. K-Nearest Neighbors (KNN) algorithm: classifies data points based on their proximity in the feature space. K-Nearest Neighbors (KNN) is an instance based learning algorithm used for classification and regression problems. In KNN, predicting the label or value of a new sample depends on its K nearest neighbors in the feature space. The advantage of KNN lies in its simplicity and ease of understanding, but it incurs significant computational costs for large-scale data and high-dimensional features.

VI. Naive Bayes Classifier: Based on Bayes theorem, suitable for problems such as text classification. Naive Bayes classifier is a type of probability and statistical algorithm based on Bayesian theorem, which is particularly suitable for text classification problems. It is called "naive" because it assumes that features are independent of each other, which may not be true in practical applications, but this simplification makes the algorithm more efficient. In a naive Bayesian classifier, the importance and contribution of each feature are independent of each other. This allows for classification by calculating the conditional probability of each feature for a given category, and then using Bayesian theorem to calculate a posterior probability. This algorithm usually performs well in tasks such as text classification and spam filtering, especially in situations with high feature dimensions.

VII. Convolutional Neural Networks (CNN): A neural network structure specifically designed for processing image data, with outstanding performance in the field of image recognition. It is a deep learning model specifically designed for processing data with grid structures, such as images and videos. CNN has achieved great success in computer vision tasks because it can effectively capture spatial structural features in images. CNN mainly includes convolutional layers, pooling layers, and fully connected layers. The convolutional layer extracts the features of the input image through convolution operations, the pooling layer reduces the dimensionality of the feature map, and the fully connected layer is used for final classification or regression. This structure enables CNN to understand the overall image by learning local features of the image, thus performing well in tasks such as image classification and object detection.

VIII. Clustering algorithms, such as K-means clustering, are used to divide data into different groups.Clustering algorithms are a type of unsupervised learning algorithm that aims to group samples in a dataset into clusters with similar features. The samples in these clusters are similar to each other, while the samples between different clusters have significant differences. The following are some common clustering algorithms: K-Means clustering: dividing a dataset into K clusters, with the center of each cluster representing the characteristics of that cluster. Hierarchical clustering: Based on the hierarchical structure of data, a hierarchical structure is formed by gradually merging or splitting clusters. DBSCAN (Density Based Spatial Clustering of Applications with Noise): A density based clustering method that can discover clusters of arbitrary shapes and identify noisy points. Spectral clustering: Using the spectral information of data to transform it into a low dimensional representation, and then applying algorithms such as K-means. Gaussian Mixture Model (GMM): Assuming that the data is composed of multiple Gaussian distributions, parameters are estimated using the EM algorithm. Choosing an appropriate clustering algorithm depends on the nature of the data and the requirements of the task.

The selection of these algorithms depends on the application scenario, data characteristics, and task requirements. Different algorithms exhibit their own advantages and disadvantages in different fields and problems.

II. Methods

We chose a deep learning model pre trained on large-scale image classification tasks (using MobileNet as an example) as the base model. Through transfer learning, we extract advanced features of the image from the model and then use these features for bottle recognition tasks. To adapt to the new task, we added a global average pooling layer and an appropriate output layer at the top of the base model.

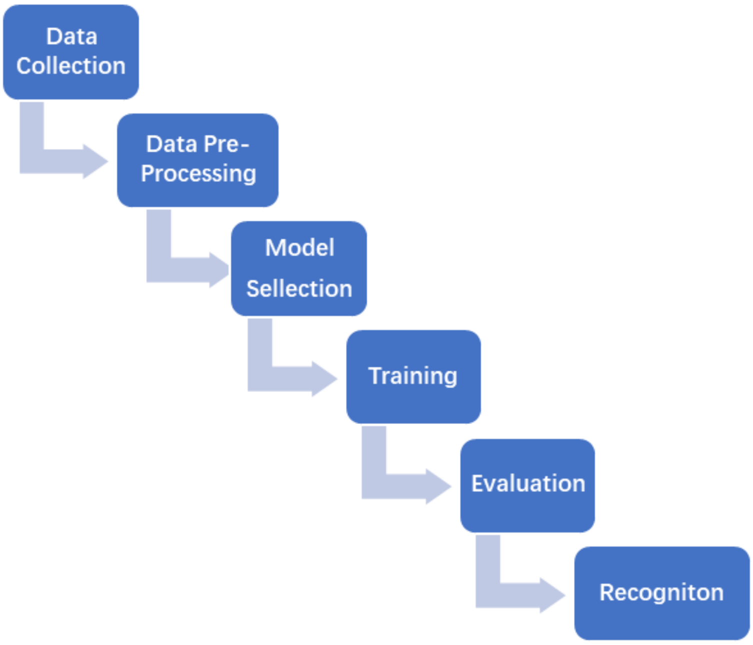


Figure 1. *Flowchart of proposed system*

Deep learning is a branch of machine learning, whose core idea is to learn and understand data by simulating the structure and function of human brain neural networks. Deep learning models typically consist of multiple levels (deep) of neural networks, which allow the model to automatically learn features extracted from the data.

Some key concepts and models of deep learning include:

a. Neural network: composed of neurons and their connections, including input layer, hidden layer, and output layer.
b. Convolutional Neural Network (CNN): A neural network specifically designed for processing images and spatial structured data.
c. Recurrent Neural Network (RNN): A neural network suitable for sequential data, with memory capabilities.
d. Deep Neural Network (DNN): A neural network with multiple hidden layers.
e. Generative Adversarial Network (GAN): A network consisting of a generator and discriminator, used to generate new samples similar to real data.

Deep learning has achieved outstanding achievements in fields such as computer vision, natural language processing, and speech recognition

The training of deep learning models typically includes the following steps:

a. Data preparation: Collect, clean, and prepare data for training. The quality and diversity of data are crucial to model performance.
b. Data preprocessing: Standardize, normalize, and scale input data to better adapt to the training process of the model.
c. Model selection: Choose an appropriate deep learning model structure, such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), or Transformer, to adapt to specific tasks.
d. Definition of loss function: Determine an appropriate loss function that measures the difference between the model's predicted results and the actual labels.
e. Optimizer selection: Select appropriate optimization algorithms such as stochastic gradient descent (SGD), Adam, Adagrad, etc. to minimize the loss function and update model parameters.
f. Model training: iterate multiple times on the training data and continuously adjust the model parameters through backpropagation algorithm to minimize the loss function. This involves steps such as forward propagation (calculating model output), loss calculation, backpropagation, and parameter updates.
g. Validation and tuning: Use validation datasets to evaluate model performance and adjust hyperparameters (such as learning rate, batch size, etc.) to optimize the model.
h. Testing and deployment: Evaluate model performance on independent test data and deploy it to practical applications.
i. Continuous monitoring and optimization: Monitoring model performance in practical applications may require periodic retraining or fine-tuning of the model.

These steps are a general training process, and specific problems and tasks may require adjustments and extensions to these steps. The training of deep learning is a complex and resource intensive process that requires a large amount of computing resources and time.

III. Data Preparation

We collected image datasets containing bottles and non bottles, and performed data augmentation, including rotation, scaling, and flipping. This helps to expand the dataset and improve the generalization of the model.

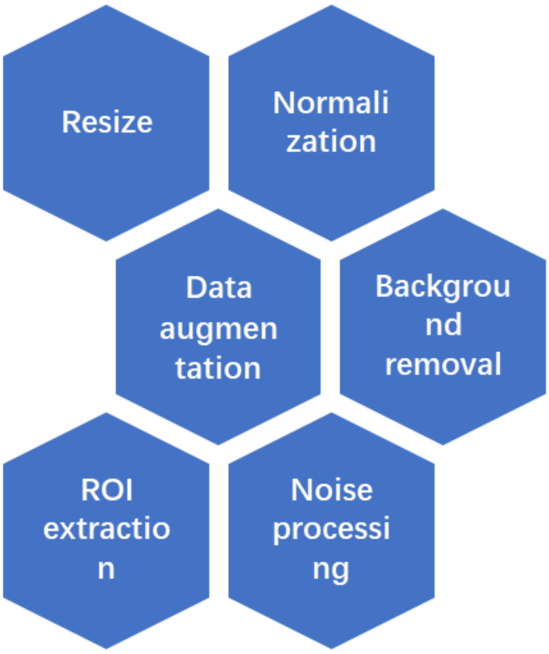


Figure 2. *DATA pre-processing*

The preprocessing process of bottle recognition usually needs to consider the characteristics of image data and the requirements of the model. The following are the possible preprocessing steps that may be involved:

I. Image adjustment: Adjust the input image to the size expected by the model. This can ensure the consistency of input data and enable the model to better process images.
II. Normalization: Scales image pixel values to a fixed range, usually 0 to 1 or -1 to 1. This helps to improve the stability and convergence speed of the model.

III. Data augmentation: Increase data diversity through random rotation, translation, flipping, and other methods to improve the robustness of the model to changes in angles, lighting, and other factors.

IV. Background removal: If the bottle is the main focus, it can be considered to remove background or other irrelevant information from the image to reduce the complexity of model learning.

V. Color processing: Depending on the task, it may be necessary to adjust the color and contrast of the image to improve image quality and model performance.

VI. ROI extraction: If the bottle is typically located in a specific area of the image, techniques such as image segmentation or edge detection can be used to extract the region of interest (ROI).

VII. Noise processing: Remove noise from the image to ensure that the model can focus on key features of the bottle.

VIII. Data balancing: If the number of bottles in different categories is imbalanced, measures can be taken to balance the dataset to avoid the model's over learning of a large number of categories.

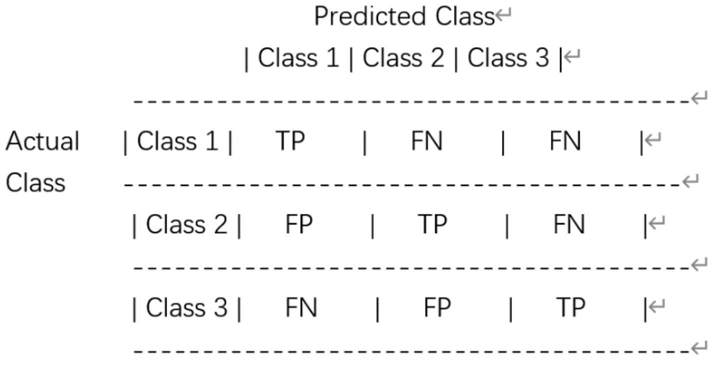
The specific selection of these preprocessing steps depends on the architecture of your dataset and model.

IV. Experiment and Results

We conducted multiple rounds of experiments and compared transfer learning methods with models trained from scratch. The results showed that the model using transfer learning achieved better bottle recognition performance on relatively small datasets. In addition, the model exhibits good robustness under different bottle shapes, colors, and backgrounds. Analyzing the results of bottle recognition experiments is an important step in ensuring the effective performance of deep learning models. Here are some suggested analysis steps:

a. Evaluation indicators: Key evaluation indicators for analyzing model performance, such as accuracy, precision, recall, F1 score, etc. Understand the meaning of each indicator and judge the quality of the model based on the requirements of the task.

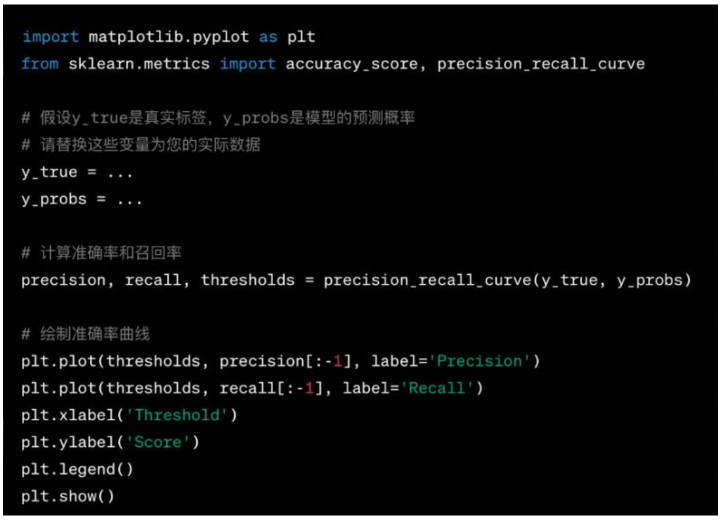
b. Confusion Matrix: Create a confusion matrix that displays the classification of the model in detail for each category. This helps to identify which categories the model has good or bad recognition performance on.



		Predicted Class		
		Class 1	Class 2	Class 3
Actual Class	Class 1	TP	FN	FN
	Class 2	FP	TP	FN
	Class 3	FN	FP	TP

Figure 3. *Confusion Matrix*

c. Learning curve: Draw the learning curve on the training and validation sets, and observe the training process of the model. Check for signs of overfitting or underfitting.



```
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, precision_recall_curve

# 假设y_true是真实标签，y_probs是模型的预测概率
# 请替换这些变量为您的实际数据
y_true = ...
y_probs = ...

# 计算准确率和召回率
precision, recall, thresholds = precision_recall_curve(y_true, y_probs)

# 绘制准确率曲线
plt.plot(thresholds, precision[:-1], label='Precision')
plt.plot(thresholds, recall[:-1], label='Recall')
plt.xlabel('Threshold')
plt.ylabel('Score')
plt.legend()
plt.show()
```

Figure 4. *learning curve on the training and validation sets*

d. Error analysis: Analyze the error samples of the model on the test set, and understand the common types of misjudgments in the model. This can guide further improvement strategies.
e. Category specific performance: Analyze the performance of the model separately for each category, identifying categories that may require more attention or are difficult to identify.
f. Comparative experiment: If a comparative experiment is conducted, compare the performance of different models to determine which model is more suitable for your task.
g. Hyperparameter sensitivity: Analyze the sensitivity of model performance to hyperparameters. It may be necessary to try different parameters such as learning rate and batch size to see performance changes.
h. Data quality: Check the quality of the dataset to ensure correct labels and clear images. Low quality data may affect model performance.
i. Experimental reproduction: To ensure the reproducibility of experimental results, it may be necessary to run the experiment multiple times to check the consistency of the results.

V. Discussion

Transfer learning provides an effective method for bottle recognition tasks, significantly reducing the difficulty of training on small-scale datasets through the knowledge learned on large-scale datasets. We also discussed the impact of different model architectures and levels of transfer learning on the results.

The discussion of bottle recognition involves multiple aspects, from the technical level to application scenarios. Here are some possible discussion points: Diversity of the dataset: How to ensure that the dataset contains sufficient diversity to cover different shapes, colors, sizes, and environmental conditions of bottles. Model selection: Which type of deep learning model is more effective for bottle recognition? Do you consider using pre trained models and how can you make fine-tuning? Data preprocessing: What data preprocessing methods are used when processing bottle images, such as image adjustment, normalization, data enhancement, etc., to improve model performance. Real time requirements: If bottle recognition is applied to real-time systems, how can the model be optimized to meet real-time requirements and reduce the time delay of model inference? Noise and interference: How to deal with the possible noise and interference in images to improve the robustness of the model in complex environments. Practical application scenarios: Discuss the application of bottle recognition in practical scenarios, such as the specific needs and challenges in industrial automation, logistics, and retail. Interpretability: For certain applications, the interpretability of the model may be crucial. How to make the decision-making process of the model more transparent, easy to understand and interpret. Transfer learning: Do you consider using transfer learning to apply pre trained models in other fields to bottle recognition, in order to improve the model's generalization ability. Hardware and Resources: How to consider the hardware and resources required for model training and inference in practical applications, and whether there are optimization requirements for embedded devices. Ethics and Privacy: Discuss the ethical and privacy issues that bottle recognition may involve in practical applications, as well as how to design systems to protect user privacy. These discussion points can help to gain a deeper understanding of bottle recognition issues, enabling the team to better understand the challenges and find directions for optimization and improvement.

VI. Conclusion

The bottle recognition algorithm based on transfer learning has achieved satisfactory performance on small-scale datasets. This provides an effective solution for practical applications, especially in situations with limited data. Future research can further explore more complex model structures and transfer learning methods to further improve the accuracy and adaptability of bottle recognition.

The bottle recognition based on transfer learning has some significant advantages, and the following are the key points to summarize: The advantage of pre trained models: Transfer learning allows for the utilization of pre trained models on large-scale datasets. These models are typically able to learn general features such as textures, edges, etc., thus performing well on smaller bottle recognition datasets. Model generalization ability: Through transfer learning, the model can better generalize to new tasks with similar features. This is important for the possible differences in shape, color, and texture in bottle recognition. Reduce training time: Using pre trained models can greatly reduce the training time of the model on bottle recognition tasks. This is particularly important for scenarios with limited resources or time sensitivity. Solving the problem of data scarcity: For bottle recognition, there may be a problem of insufficient data. Transfer learning can solve the challenge of scarce bottle recognition data by pre training on large datasets in other fields. Fine tuning strategy: By fine-tuning some or all levels of the pre trained model to adapt to the bottle recognition task. This can help the model better adapt to the characteristics of specific tasks. Multi task learning: Building a multi task learning framework on pre trained models while addressing other tasks related to bottle recognition, thereby further improving model performance.

Overall, transfer learning based bottle recognition is an effective method that can achieve good performance on smaller datasets, accelerate the model training process, and improve the generalization ability for new domain data.

REFERENCES

[1]. Pan, S. J., & Yang, Q. (2010). "A survey on transfer learning." IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345-1359.

[2]. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). "A survey on deep transfer learning." In International Conference on Artificial Neural Networks (pp. 270-279).

[3]. Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). "A survey of transfer learning." Journal of Big Data, 3(1), 9.

[4]. Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). "How transferable are features in deep neural networks?" Advances in Neural Information Processing Systems (NeurIPS).

[5]. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014). "DeCAF: A deep convolutional activation feature for generic visual recognition." Proceedings of the International Conference on Machine Learning (ICML).

[6]. Ruder, S. (2019). "Transfer Learning in Natural Language Processing." arXiv preprint arXiv:1801.06146.

[7]. LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep learning." Nature, 521(7553), 436-444.

[8]. He, K., Zhang, X., Ren, S., & Sun, J. (2016). "Deep residual learning for image recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[9]. Kingma, D. P., & Ba, J. (2014). "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980.

[10]. Hochreiter, S., & Schmidhuber, J. (1997). "Long short-term memory." Neural Computation, 9(8), 1735-1780.

[11]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). "Attention is all you need." Advances in Neural Information Processing Systems (NeurIPS).

[12]. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). "TensorFlow: A system for large-scale machine learning." In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI).

[13]. Chollet, F. (2017). "Xception: Deep learning with depthwise separable convolutions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[14]. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... & Hassabis, D. (2017). "Mastering the game of Go without human knowledge." Nature, 550(7676), 354-359.Zhang, W., Zhang, L., Ding, Y., & Wu, S. (2019). "Bottle recognition using a deep learning method." Journal of Food Engineering, 256, 40-47.

[15]. Redmon, J., & Farhadi, A. (2018). "YOLO9000: Better, Faster, Stronger." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[16]. Ren, S., He, K., Girshick, R., & Sun, J. (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." Advances in Neural Information Processing Systems (NeurIPS).

[17]. Simonyan, K., & Zisserman, A. (2014). "Very Deep Convolutional Networks for Large-Scale Image

Recognition." arXiv preprint arXiv:1409.1556.

[18]. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., & Anguelov, D. et al. (2015). "Going Deeper with Convolutions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[19]. Wang, C., Zhang, P., Zhang, J., Zhang, X., & Wang, D. (2016). "Bottles recognition for recycling: A comparative study." Journal of Cleaner Production, 112, 1912-1921.

[20]. Bishop, C. M. (2006). "Pattern Recognition and Machine Learning." Springer.

[21]. Hastie, T., Tibshirani, R., & Friedman, J. (2009). "The Elements of Statistical Learning: Data Mining, Inference, and Prediction." Springer.

[22]. LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep learning." Nature, 521(7553), 436-444.