International Journal of Scientific Research in Science and Technology
Available online at : www.ijsrst.com
Print ISSN: 2395-6011 | Online ISSN: 2395-602X
doi : https://doi.org/10.32628/IJSRST5241113

Deep Learning-Based Tomato Ripeness Detection : A ResNet-152 Approach

Mohammed Mutahar*, Shyamalan Kannan, Mohammed Mustafa Jafer, Maneesh Ragavendra K

Computer Science and Engineering, Presidency University, Bangalore, Karnataka, India

ARTICLEINFO

Article History:

Accepted: 15 Dec 2023

Published: 06 Jan 2024

Publication Issue :

Volume 11, Issue 1

January-February-2024

Page Number :

34-41

ABSTRACT

This research paper presents an advanced deep learning framework for tomato ripeness and maturity detection, employing Convolutional Neural Networks (CNNs) as the primary tool for automated classification. In particular, the study emphasizes the utilization of the ResNet-152 architecture, a sophisticated deep neural network known for its exceptional performance in image classification tasks. ResNet-152 addresses the challenges of fine-grained categorization by enabling the extraction of intricate visual features such as color nuances, textures, and shapes inherent to tomatoes. This eliminates the necessity for manual feature engineering, enhancing the model's ability to discern between "RIPE," "UNRIPE," and "ROTTEN" tomato classes with unprecedented accuracy.

The ResNet-152 architecture's effectiveness lies in its unique design, featuring residual blocks that facilitate the training of very deep networks. This architecture mitigates the vanishing gradient problem and enables the model to learn complex hierarchical features, contributing to its state-of-the-art performance in image classification. In the context of tomato ripeness detection, ResNet-152 acts as a powerful tool for capturing and understanding the intricate visual cues that define different ripeness states, laying the foundation for an efficient and accurate automated sorting system in the agricultural and food processing industries.

Keywords: Convolutional Neural Networks (CNNs), ResNet-152, deep neural network, fine-grained categorization, residual blocks, tomato ripeness detection, image classification, automated sorting system, agricultural, food processing industries.

I. INTRODUCTION

Product inspection is a critical facet of quality assurance, serving as a frontline defence against the delivery of defective goods to customers. However, the current inspection processes face substantial challenges that impede their effectiveness. Notably, there is a significant disjunction between customer expectations for comprehensive inspections and the practical limitations imposed by space and manpower constraints, resulting in only a meager 10% inspection rate. This incongruity not only jeopardizes the assurance of product quality but also raises concerns about meeting customer expectations. Additionally, the temporal misalignment between case processing and subsequent inspections, coupled with a delay in expert availability, hinders the implementation of real-time corrective actions, posing a substantial challenge in maintaining and enhancing product quality standards.

Moreover, the reliance on human visual inspection introduces an inherent subjectivity that adds complexity to quality control efforts. The variability in individual judgments raises questions about the consistency and objectivity of defect identification, necessitating a shift towards more objective and automated inspection methodologies. This research aims to address these multifaceted challenges by proposing an innovative solution that not only optimizes inspection efficiency but also integrates real-time corrective measures, ultimately contributing to a more reliable and customer-centric quality assurance process.

The agricultural and food processing industries are no strangers to the challenges posed by produce quality control. In these sectors, the ability to accurately assess the ripeness and maturity of fruits and vegetables holds substantial significance. This is especially true for commodities like tomatoes, where optimal ripeness is a key determinant of taste, texture, and market value. However, the manual assessment of tomato ripeness is labour-intensive, subjective, and inherently prone to errors, the consequence? A significant portion of produce maybe wrongly classified as unripe or ripe or overripe, leading to economic losses for farmers, poor produce quality for consumers and a waste of valuable resources. Beyond the farm, retailers, and consumers share this concern. Unripe or overripe tomatoes can affect the taste and nutritional value of dishes, impacting consumer satisfaction and health. The ripple effect extends to the environment, as the production and disposal of unused or spoiled produce contribute to food waste and its associated ecological footprint.

To address this challenge, we present a deep learning-based tomato ripeness and maturity detection system. Our system leverages convolutional neural networks (CNNs), a class of deep learning models specifically designed to tackle image-related tasks. The diverse nature of produce varieties, influenced by factors like soil quality, climate, and growing conditions, poses a significant challenge. Technological limitations, sustainability concerns, and the necessity for compliance with food safety regulations further add layers of complexity to our proposed solution.

II. METHODS AND MATERIAL

A. Data Collection and Preprocessing

1. Dataset Description:

The dataset to be used for this research comprises images of tomatoes categorized into three classes: "RIPE," "UNRIPE," and "ROTTEN." The dataset must be collected from various sources, which may include public repositories, and must consist at least a total of 3000 images, 1000 belonging to each class. These images must be coloured and can be resized to a standard size of say 120x120 pixels. The dataset's distribution across classes is as follows: 1000 Ripe (700 for training, 150 for testing, 150 for validation), 1000 Unripe (700 for training, 150 for testing, 150 for validation), 1000 Rotten (700 for training, 150 for testing, 150 for validation).

2. Data Preprocessing:

To prepare the dataset for model training, a series of preprocessing steps can be applied: Image Augmentation: Image augmentation is a critical step to increase the dataset's diversity and reduce overfitting. The Keras ImageDataGenerator class can be utilized to perform various data augmentations, including:

Random rotations within a specified range: Rotating images within a specified range introduces variability in the orientation of tomatoes in the images, since tomatoes are going to appear at different angles during inspection.

Horizontal and vertical flips: This augmentation technique can help in tackling issues related to the left-to-right or top-to-bottom orientation of tomatoes in images.

Zooming in/out: Zooming in/out changes the scale of the tomato within the image generalizing the model to different sizes of tomatoes that may be encountered in real-world scenarios.

Adjusting brightness and contrast: Modifying brightness and contrast simulates variations in lighting conditions, which are common in real-world settings. By introducing these variations, the model becomes more robust to lighting changes, improving generalization.

Normalization: To enhance the model's training convergence, image pixel values can be normalized. Typically, values need to be rescaled to the range [0, 1] or [-1, 1] to ensure consistency in pixel intensity.

B. Model Architecture

ResNet-152 Architecture

For tomato ripeness and maturity classification, we propose the use of the ResNet-152 architecture. ResNet, or Residual Networks, are known for their deep architecture with skip connections, which enables them to handle vanishing gradient problems and learn more complex features. The decision to use ResNet-152 was based on its ability to handle vanishing gradient problems, allowing for the training of deeper networks, and its advantage over older architectures like VGG or AlexNet in capturing intricate features essential for fine-grained classification tasks. ResNet-152 is a deep variant of the architecture, offering superior feature extraction capabilities. This advancement makes it possible to train extremely deep networks, and ResNet-152 is an example of one of the most complex variations of this architectural paradigm. The ResNet-152 layered architecture is explained briefly below:

Input Layer: ResNet-152's first input layer is made to take in images with fixed dimensions, typically 224x224 pixels, and three different colour channels: red, green, and blue.

Convolutional Layer: ResNet-152's main convolutional layer is tasked with extracting basic features from the input imagery. After this crucial layer, ResNet-152 develops into a series of residual blocks, each of which contains several convolutional layers. These layers sequentially and methodically extract from the input data increasingly complex and abstract features.

Residual Blocks: The integration of residual blocks is ResNet's defining architectural innovation. Each residual block is made up of a stack of convolutional layers and a shortcut connection, also known as a skip connection, that links the block's input and output together. This architectural element solves the infamous vanishing gradient issue, making it possible to train extraordinarily deep neural networks. An architectural depth of 152 layers is notable in ResNet-152, which attests to the abundance of these residual blocks.

Pooling Layer: ResNet-152 strategically incorporates max-pooling layers at different nodes in the network to coordinate the spatial downscaling of feature maps. This method permits the reduction of the spatial dimensions of feature maps while simultaneously retaining the most important data.

The Fully Connected Layer: The network's final stratum takes on the responsibility of determining the final classification. It begins with a dense layer that is fully connected, and is then expanded upon by a layer that pools global averages. In accordance with the cardinality of the classification task, which is typically illustrated by 1000 classes in the context of ImageNet, for example, this final fully connected layer can accommodate an equivalent number of units.

Softmax Layer: The final layer is frequently equipped with a softmax activation function, a crucial element that coordinates the conversion of the outputs from the fully connected layer before it into class probabilities.

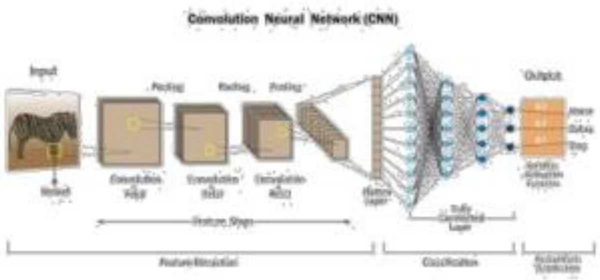

Figure 1: General Structure of a CNN

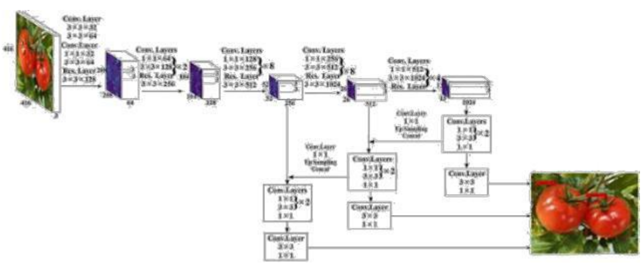

Figure 2: Sample CNN for Tomato Detection

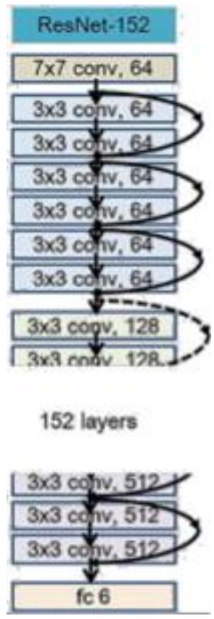

Figure 3: ResNet-152 Architecture

Compared to older architectures like VGG or AlexNet, ResNet-152 has several advantages. It allows us to build deeper networks while maintaining model International Journal of Scientific Research in Science and Technology (www.ijsrst.com) performance, thanks to residual connections. Deeper networks can capture more intricate features, crucial for fine-grained classification tasks like tomato ripeness detection. These residual connections also simplify training and mitigate the vanishing gradient problem.

C. Training and Evaluation

1. Training Process

The model must be fine-tuned on the ResNet-152 architecture using transfer learning. We start with the pre-trained ResNet-152 weights and adapted them to our specific classification task. The training process must include hyperparameter tuning, including setting appropriate learning rates, batch sizes, and the number of training epochs.

2. Evaluation Metrics

Any particular model's performance can be assessed using the following evaluation metrics:

Accuracy : Accuracy is a standard metric that measures the overall correctness of predictions.

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

Precision: Precision quantifies the model's ability to correctly classify ripe, unripe, or rotten tomatoes within a class.

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

Recall: Recall assesses the model's capacity to identify all ripe, unripe, or rotten tomatoes within the dataset.

F1-Score: The F1-Score is the harmonic mean of precision and recall and provides a balanced assessment of the model's performance.

$$F1 - Score = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)}$$

Additionally, the two loss functions that can be used during training and evaluation are:

Binary Crossentropy: Binary Crossentropy, also known as log loss or logistic loss, is commonly used when dealing with binary classification problems. It quantifies the dissimilarity between the predicted probability distribution and the actual binary labels (0 or

1) It is computed for each individual sample and then averaged across all samples. The formula for Binary Crossentropy loss for a single sample is as follows:

$$L(y, y') = -[y\, log(y') + (1 - y)\, log(1 - y')]$$

y represents the true label (0 or 1) for the sample.

y' represents the predicted probability for class 1 (typically produced by the model). It's essentially the model's confidence that the sample belongs to class 1.

The loss value is minimized during training, with lower values indicating better model predictions. This loss function encourages the model to assign a higher probability to the correct class (1 or 0) for each sample.

Categorical Crossentropy: Categorical Crossentropy loss is used in multi-class classification problems, where the target labels can take on multiple classes. This loss measures the dissimilarity between the predicted class probabilities and the actual class labels. The formula for Categorical Crossentropy loss for a single sample is as follows:

$$L(y, y') = -\sum_{i=1}^{C} y_i \log(y'_i)$$

y is a one-hot encoded vector of true labels, where yi is 1 for the correct class and 0 for all other classes.

y' is a vector of predicted class probabilities produced by the model.

The loss is computed for each sample and then averaged across all samples. In practice, this loss penalizes the model for assigning low probabilities to the true class while encouraging high probabilities for the correct class. It aims to maximize the likelihood of the true class given the input.

In the context of our tomato ripeness and maturity classification task, the selection of the Categorical Crossentropy loss function is driven by the inherent nature of the problem involving three distinct classes: "RIPE," "UNRIPE," and "ROTTEN." Categorical Crossentropy is specifically designed for multi-class classification scenarios, accommodating target labels that can take on multiple classes. Its one-hot encoded representation of true labels, with a value of 1 for the correct class and 0 for others, aligns well with our dataset structure. This loss function effectively penalizes the model for low probabilities assigned to the true class while encouraging high probabilities for the correct class, thus facilitating optimal training for accurate predictions across the diverse categories of tomato ripeness.

III. RESULTS AND DISCUSSION

The reason CNNs were suggested to formulating the ResNet152 has demonstrated state-of-the-art performance in a wide range of computer vision tasks, solution was that CNNs are specifically designed for including image classification, which is relevant to our image-related tasks. They excel at automatically problem statement. The ripeness and rot of tomatoes are learning hierarchical features from raw pixel data. In the characteristics that are primarily discernible through case of tomato ripeness detection, these features could visual cues. Changes in colour, texture, and blemishes include colour, texture, and shape information. CNNs are all indicative of the ripeness or rot of a tomato. For can capture both low-level features (e.g., edges, colour example, a ripe tomato may exhibit a vibrant red colour, gradients) and high-level features (e.g., complex patterns) through their deep convolutional layers, which may while a rotten one may display dark, mouldy spots. However, a nuance in these patterns is the variation prove to be challenging to achieve with traditional based on factors like lighting conditions and tomato machine learning models. CNNs also possess the ability varieties. A ripe tomato may appear dark under dim to capture spatial invariance, meaning they can lighting conditions and can be falsely classified as rotten recognize features regardless of their location in the since a dark mouldy patch is a characteristic of a rotten image. In tomato ripeness detection, the location of the tomato. Deep learning models, with their ability to ripe or rotten portions of a tomato can vary, even still the capture intricate and subtle features in images, can network can learn to identify these features irrespective of their position within the image. Moreover, traditional effectively handle this variability. machine learning techniques like Decision Trees and SVMs often require manual feature engineering, where domain expertise is needed to extract relevant features from the data, and although it is arguable how experienced one may need to be to recognize a rotten tomato, CNNs contrastingly can automatically learn and extract features directly from the input images, reducing the need for manual feature engineering and thereby increasing accuracy of the model. To harness the full potential of computational acceleration, the utilization of Graphics Processing Units (GPUs) or Tensor Processing Units (TPUs) is recommended. These hardware accelerators are adept at handling parallel computations efficiently, significantly accelerating the training process and alleviating computational bottlenecks.

The main reason we suggest a CNN is that CNNs benefit from the availability of pretrained models on large image datasets like ImageNet. Transfer learning allows you to leverage the knowledge and feature representations learned from these datasets and fine-tune them for your own specific problems like tomato ripeness detection. This can significantly reduce the amount of labelled data required for training and improve the model's performance, saving you effort of collecting data.

Going a step further this paper also suggests the pretrained CNN model - ResNet152 as a possible solution for the tomato ripeness detection problem. ResNet152 is exceptionally deep, consisting of 152 layers, making it a more complex and expressive architecture compared to AlexNet or VGGNet. ResNet152's key innovation is the introduction of residual connections. The residual connections in ResNet152 allow for the training of deeper networks without degradation in performance. This is in contrast to networks like VGGNet or AlexNet, which may suffer from diminishing returns as they become deeper.

ResNet152 has demonstrated state-of-the-art performance in a wide range of computer vision tasks, including image classification, which is relevant to our problem statement. The ripeness and rot of tomatoes are characteristics that are primarily discernible through visual cues. Changes in colour, texture, and blemishes are all indicative of the ripeness or rot of a tomato. For example, a ripe tomato may exhibit a vibrant red colour, while a rotten one may display dark, mouldy spots. However, a nuance in these patterns is the variation based on factors like lighting conditions and tomato varieties. A ripe tomato may appear dark under dim lighting conditions and can be falsely classified as rotten since a dark mouldy patch is a characteristic of a rotten tomato. Deep learning models, with their ability to capture intricate and subtle features in images, can effectively handle this variability.

The solution presented is scalable to food processing industries since it can be integrated into automated sorting and quality control systems, reducing the need for manual inspection and potentially improving the overall efficiency of the process.

In what ways can the scope of our study be broadened? Certainly, a lot of aspects of this study leaves areas for improvement. This study restricts its scope to visual sensors, leaving room for integration of other sensors such as the humidity sensors or gas sensors, which can help provide additional data for ripeness and freshness assessment, enhancing the overall accuracy of the system. Going beyond ripeness detection, the system may even be expanded to predict quality metrics like sugar content, firmness, and shelf life, providing more comprehensive insights. Capitalizing on the scalable and parallel processing capabilities offered by cloud platforms, such as AWS, Google Cloud, and Azure, presents a scalable solution to computational challenges. Leveraging cloud computing services facilitates the efficient distribution of computational tasks, overcoming local hardware limitations and ensuring the timely execution of complex analyses.

IV. CONCLUSION

In conclusion, the adoption of Convolutional Neural Networks (CNNs), and specifically ResNet152, for tomato ripeness detection proves to be a judicious choice owing to the innate capabilities of CNNs in automatically learning hierarchical features from raw pixel data. The spatial invariance and capacity to capture both low-level and high-level features make CNNs adept at discerning subtle variations in tomato characteristics, such as colour, texture, and shape, crucial for ripeness detection. The utilization of powerful hardware accelerators, such as GPUs or TPUs, further enhances the efficiency of the training process, ensuring the scalability of the proposed solution for real-world applications.

Moreover, the paper underscores the advantages of CNNs over traditional machine learning models, particularly in eliminating the need for manual feature engineering. CNNs can autonomously learn and extract relevant features directly from input images, reducing the dependence on domain expertise and potentially increasing the accuracy of the model. Transfer learning, facilitated by pretrained models like ResNet152 on ImageNet, emerges as a pivotal strategy to leverage existing knowledge and feature representations for tomato ripeness detection, diminishing the labelled data requirements and improving model performance.

The scalability of the proposed solution for integration into food processing industries is a significant strength, offering potential automation of sorting and quality control systems. However, the study also points towards avenues for improvement, suggesting the incorporation of additional sensors such as humidity or gas sensors to provide supplementary data for ripeness and freshness assessment. Beyond ripeness detection, the system could be expanded to predict various quality metrics, offering more comprehensive insights into tomato characteristics.

Looking ahead, the study advocates for the integration of cloud computing services to address computational challenges, providing a scalable solution through platforms like AWS, Google Cloud, and Azure. This not only overcomes local hardware limitations but also ensures the efficient distribution of computational tasks, enabling the timely execution of complex

V. REFERENCES

[1]. Agarwal, M., et al. Tomato Leaf Disease Detection using Convolutional Neural Networks. International Conference on Computational Intelligence and Data Science (ICCIDS 2019).
[2]. Afonso, M., et al. Tomato Fruit Detection and Counting in Greenhouses Using Deep Learning. Frontiers in Plant Science, 2020. DOI: 10.3389/fpls.2020.571299.[2]
[3]. Zhao, M., et al. Determination of Quality and Maturity of Processing Tomatoes Using Near-Infrared Hyperspectral Imaging with Interpretable Machine Learning Methods.
[4]. Appe, S.R.N., et al. Tomato Ripeness Detection and Classification using VGG based CNN Models. International Journal of Intelligent Systems and Applications in Engineering (IJISAE, 2023, 11(1), 296â€“302). ISSN: 2147-6799.

Cite this article as :

Mohammed Mutahar, Shyamalan Kannan, Mohammed Mustafa Jafer, Maneesh Ragavendra K, "Deep Learning-Based Tomato Ripeness Detection : A ResNet-152 Approach", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 11 Issue 1, pp. 34-41, January-February 2024. Available at doi :

https://doi.org/10.32628/IJSRST5241113

Journal URL : https://ijsrst.com/IJSRST5241113