Journal of Soft Computing Paradigm (ISSN: 2582-2640)
www.irojournals.com/jscp/

Enhancing Fruit Maturity Detection using Convolutional Neural Networks Algorithm Compared with Naive Bayes Algorithm

Frank Alwin S.[1], Manikandan G.[2]

[1]Research scholar, [2]Research Guide, Corresponding Author, Department of Department of Applied Electronics, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamil Nadu, India.

E-mail: [1]192110528.sse@saveetha.com, [2]manikandang.sse@saveetha.com

Abstract

This study aims to compare the accuracy of the fruit maturity detection enhancement using Convolutional Neural Networks (CNNs) and Naive Bayes Algorithm, with a specific focus on various methods. This research also evaluates their effectiveness in Enhancing Fruit Maturity Detection. Using G*Power parameters of 0.8 for each group, 0.07 for alpha, and 0.2 for beta, the total sample size is calculated as 10,000 (5,000 samples in group 1 and 5,000 in group 2). To improve results, synthetic datasets were created. The Convolutional Neural Networks was implemented, and configured with Naive Bayes in deep learning. The selection of the most suitable approach is based on the outcomes derived from the SPSS statistical analysis. After evaluating both algorithms, it became evident that CNN outperformed Naïve Bayes, exhibiting a performance accuracy of 81.56% versus 54.79%. The sample T-test indicated no significant difference between CNN and Naïve Bayes, with a p-value of 0.048 ($p < 0.05$). This suggests that Convolutional Neural Networks can handle datasets of varying sizes effectively, while Naïve Bayes performs reasonably well with smaller datasets and can be trained quickly.

Keywords: CNN(Convolutional Neural Networks), Naive Bayes, Fruit Maturity Detection, Accuracy, deep learning, image classification, Augmentation.

DOI: https://doi.org/10.36548/jscp.2024.4.004
Received: 13.11.2024, received in revised form: 02.12.2024, accepted: 21.01.2025, published: 04.02.2025

1. Introduction

Convolutional Neural Networks (CNNs) are deep learning methods which is mainly crafted for tasks centered around image recognition. The application of CNNs in the recognition of various image datasets is demonstrated in the existing research [2,9]. Baker et al. [1] presented the AlexNet architecture, which significantly surpassed existing methods in the ImageNet large scale visual recognition challenges. This work played a pivotal role in establishing the Image classification and CNNs as a foundation in modern computer vision research. The Naive Bayes explains and also employs the Bayes theorem and makes a “naive” assumption of independence between features. Research on the Naive Bayes Algorithm has highlighted its strengths in situations. McCallum et al. [8] stated a comparison of event models for Naive Bayes text classification in AAAI-98 Workshop on Learning for Text Categorization. The application for enhancing fruit maturity using CNNs and Naive Bayes includes agriculture and farming, the food industry, and supply chain management. Enhancing fruit maturity primarily focuses on product quality and customer satisfaction [5]. The databases like IEEE Xplore and Web of Science directly used for receiving the most up-to-date information often allow filters based on specific keywords, authors, and time periods. Kautish et al. [5], and Bridewell et al. [3] Bayesian Classification helps in estimating the continuous distribution of various criteria. In Journal of Machine Learning Research, a total of 826 articles focussed on enhancing fruit maturity detection using the Convolutional Neural Networks (CNN) algorithm compared with the Naïve Bayes (NB) Algorithm.Common research topics related to CNN and NB include image classification, object detection, transfer learning, feature extraction, text classification, and the assumption of independence. Some notable publications in this field include:"Fruit Recognition Using Color and Texture Features with Support Vector Machine" by Zhang et al. [12]"A Comprehensive Study on Fruit Classification and Grading Techniques" in [10]"FruitNet: A Deep Learning Architecture for Real-Time Fruit Detection" in. [7]

The gaps in enhancing fruit maturity detection highlight the advantages and limitations of different machine learning models. Deep learning models, such as Convolutional Neural Networks (CNNs), are capable of learning hierarchical features directly from raw data. They automatically identify patterns and representations that may be challenging for traditional methods. Naïve Bayes operates under the assumption that features are conditionally independent given the class label. However, this assumption can be limiting in cases where features are dependent on each other [8,9]. A key advantage of Naïve Bayes is that training is generally computationally less intensive compared to deep learning models. Additionally, Naïve Bayes can perform reasonably well with smaller datasets and can be trained quickly. On the other hand, CNNs, like many deep learning models, typically require large amounts of training data to perform well. They may struggle with small datasets but excel at learning intricate patterns and spatial relationships, particularly in image-processing tasks. Given these differences, we are motivated to study enhanced fruit maturity detection by comparing CNNs and Naïve Bayes. CNNs can adapt automatically to detect subtle changes in fruit maturity based on visual cues, making them well-suited for this task. Meanwhile, Naïve Bayes relies on the assumption of feature independence, which may limit its ability to capture complex and interdependent features in image-based fruit maturity detection [4,6,12].

2. Materials and Methods

The study was conducted at the Department of Computer Science Engineering at SIMATS School of Engineering, Saveetha University. Alpha and Beta contributors have focused on safeguarding real-time assessment of fruit maturity in a production line, particularly by comparing it with the Naïve Bayes algorithm, which adds an interesting dimension to the research. Naïve Bayes is known for its simplicity and efficiency, especially in image classification. However, it is intriguing to examine how it compares to the complexity of Convolutional Neural Networks (CNNs) in the context of fruit maturity detection. Kuo et al. [6] have explored various techniques for wide range of applications in various industries, emphasizing the need for improved accuracy and efficiency. Additionally, research by Tumasyan et al. [11] highlights the importance of fruit maturity detection.

The Kaggle open-access dataset,(https://www.kaggle.com/datasets/utkarshsaxenadn/ fruits-classification) was used for acquiring images of various fruits and for training and testing machine learning models. The dataset includes five different types of fruits (apples, mangoes, bananas, grapes, and strawberries) each class contains 2000 images, resulting in 1000 images in the dataset.

The design and implementation of the proposed work utilized Python and OpenCV software, and the system operated on a 64-bit operating system. The experimental setup consisted of a computer system with a high-performance central processing unit (CPU), ample system memory (RAM), and sufficient storage capacity. Specifically, the system featured an Intel Core i7 processor with multiple cores to facilitate parallel processing and 32 gigabytes

(GB) of RAM. Python was used for code implementation. Various tools, such as Google Colab, packages, and libraries, were employed for image maturity detection. TensorFlow was used in the Convolutional Neural Networks (CNNs), while the Naïve Bayes algorithm utilized PySyft.libraries.

3. Methodology

The Convolutional Neural Networks algorithm can be used for fruit detection in images by acquiring images of the scene, pre-processing them to remove noise, and enhancing features. The algorithm is then trained to recognize the features that are characteristic of the fruit such as its shape, color, and texture. Once the fruit has been detected, relevant features that indicate the fruit's maturity such as color, size, and shape can be extracted using the CNN algorithm. These features are then used to train a machine learning model to classify fruits as mature or not mature.

The CNN model was designed, trained, and evaluated using Tensor flow with keras, the image manipulation tasks such as resizing and normalization were handled by the OpenCV.

The Fruit classification datasets are initially loaded and preprocessed. The images are resized to a consistent size of 224x224 pixels using the OpenCV library to ensure that all images have the same dimensions suitable for the CNN. Each pixel value is then normalized by dividing by 255.0 to bring the values into the range of [0, 1].

The CNN model is defined using TensorFlow and Keras, which allows for easy model building and training. The architecture starts with a series of convolutional layers (Conv2D), each followed by max-pooling layers (MaxPooling2D). These convolutional layers apply filters to the input images, capturing low-level features like edges and textures, while the pooling layers reduce spatial dimensions, retaining only the most important features. The architecture typically includes multiple convolutional blocks: the first block uses 32 filters with a kernel size of (3,3), the second block uses 64 filters, and the third block uses 128 filters to capture increasingly complex features. ReLU activation functions are used after each convolution to introduce non-linearity, allowing the network to learn more complex patterns.

The model is then compiled with the Adam optimizer, which adapts the learning rate during training, helping the model converge faster and more efficiently. The loss function used is categorical cross-entropy, as this is a multi-class classification problem. The model is trained over 10 epochs with a batch size of 32.

The data augmentation is applied during training using the ImageDataGenerator class from Keras to enhance the models ability. Augmentation techniques include random rotations up to 30 degrees, width and height shifts up to 20% of the image dimensions, zooming in and out within a range of 20%, and horizontal flipping. This helps to prevent overfitting by artificially increasing the number of the training data.

Once the model is trained, the feature extraction process takes place during the forward pass through the network. Instead of using the final classification layer (softmax), the model's output at the last pooling layer is flattened using the Flatten() layer to generate a feature vector representing the image. These feature vectors contain high-level information about the fruit images, such as the shape, texture, and spatial relationships of different parts of the fruit.

These extracted features can then be used for further analysis or fed into Naïve Bayes for final classification. Figure 1 shows the workflow of the methodology in classifying the fruits

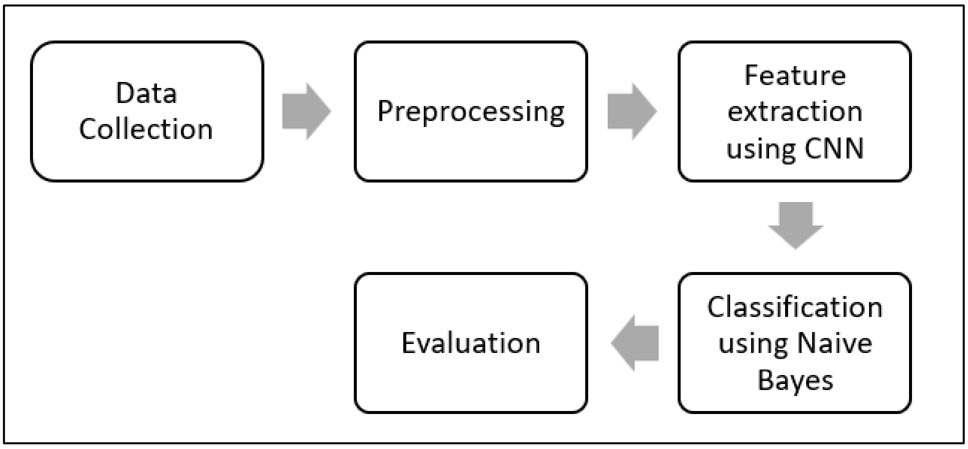

Figure 1. Workflow of the Methodology

3.1 Statistical Analysis

Statistical analysis was conducted utilizing IBM SPSS software with version 27.0 to investigate the standard error mean, and standard deviation value. The IBM SPSS Statistics Data Editor was utilized to analyze delay time forecasts using Convolutional Neural Networks (CNN) and Naïve Bayes (NB). Before proceeding with the analysis, both independent and dependent variables were defined. The accuracy of each algorithm was considered the dependent variable, influenced by image size, while the CNN and NB algorithms served as independent variables. Additionally, independent variables included temperature, humidity, ethylene exposure, and storage conditions, whereas dependent variables included color changes, fruit firmness, sugar content, and flavor development.

Finally, an independent samples t-test was performed using the iteration results to determine the statistical significance.

4. Results

Table 1 shows the Convolutional Neural Networks for different image sizes in milliseconds. So that the accuracy value increases linearly with the image size. For example, it takes 1.2 milliseconds to augment a 10 KB image and 1200 milliseconds to augment a 10000 KB image. This means that CNN processes images in fixed-size blocks. In both algorithms, a larger image size generally leads to better performance.

Table1. The Accuracy of the Convolutional Neural Networks with Respect to Image Size

Iteration	**CNN**
1	76.06
2	76.42
3	77.33
4	78.62
5	78.27
6	78.84
7	78.12
8	80.64
9	77.23
10	81.56

Table 2 shows the Naive Bayes accuracy for different image sizes in milliseconds. So that the accuracy values increase linearly with the image size. For example, it takes 1.2 milliseconds to augment a 10 KB image and 120 milliseconds to augment a 1000 KB image. This is because Naive Bayes is a distributed image system, which means that it stores the image by exchange by the users. Naive Bayes provides a structured and systematic way to solve complex problems or to perform tasks.

Table 2. The Accuracy of the Naive Bayes with Respect to Image Size

Iteration	Naive bayes
1	45.23
2	46.54
3	46.89
4	47.79
5	50.34
6	48.68
7	40.46
8	51.12
9	53.23
10	54.79

Table 3 shows the description analysis of Convolutional Neural Networks (CNN) and Naive Bayes in terms of accuracy delay. The table includes three columns: Algorithm, N, and Accuracy Delay. The Algorithm column specifies the augmented algorithm applied, N represents the number of samples, and accuracy Delay denotes the average time taken to augment an image in milliseconds. Table 3 also shows that CNN is faster than NB for augmenting the images based on standard deviation observed for the two models. The average accuracy taken to augment an image using NB is 48.507 milliseconds, while the average accuracy taken to augment an image using CNN is 78.209 milliseconds. The table additionally reveals the standard deviation of time delay for CNN is less than the standard deviation of time delay for NB. This means that the Accuracy for the CNN is more consistent than the augmentation time for NB.

Table 3. Descriptive Analysis of Convolutional Neural Networks and Naive Bayes.

Algorithms (efficiency)	N	Mean	std.Deviation	Std.Error Mean
Convolutional Neural Networks	10	78.3090	1.73741	.54942

Naive Bayes	10	48.5070	4.14315	1.31018

The results of a statistical analysis comparing the independent variables of CNN and NB are presented in Table 4. According to Levene's Test for Equality of variances, the variances of the two groups are found to be unequal. Therefore, the Welch's t-test for Equal Means was used instead of the Student's t-test. The results of the t-test indicate that there is no statistically significant difference between the means of the two groups by (Sig. (2-tailed) =0.01). The mean difference is 29.80200 milliseconds, with a standard error of 1.42071 milliseconds. The 95% confidence interval of the difference is from 26.81719 milliseconds to 32.78681 milliseconds. In other words, the table shows that CNN and NB have similar augmentation times, on average. However, there is a large amount of variation in the augmentation times within the groups.

Table 4. Compare the Results of the Convolutional Neural Networks (CNN) and Naive Bayes (NB)

Levene's Test Equality of Variances			**T-test for Equality of Means**						
Efficiency	**F**	**Sig**	**t**	**df**	**Mean Differences**	**Std.error differences**	**2-tailed test**	**Lower**	**Upper**
Equal variances assumed	4.651	.048	20.977	18	29.80200	1.42071	0.01	26.81719	32.78681
Equal variances not assumed			20.977	12.070	29.80200	1.42071	0.01	26.70853	32.89547

Figure 2 represents the graphical representation of Convolutional Neural Networks Algorithm with training accuracy and validation accuracy. The X-axis shows the number of epochs taken for the observation whereas Y-axis shows the Mean-accuracy values across the epochs.

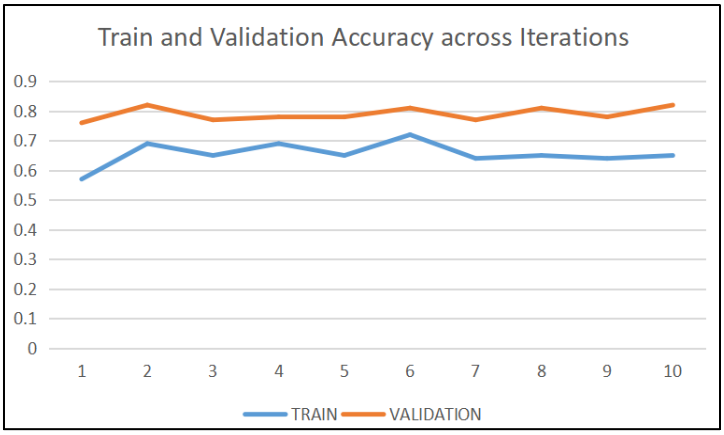

Figure 2. Training and Validation Accuracy of CNN

Figure 3 represents the graphical representation of the Naive Bayes Algorithm with training accuracy and validation accuracy. The X-axis shows the number of epochs taken for the observation whereas the Y-axis shows the Mean-accuracy values across the epochs.

CNNs, benefit from large amounts of data for training. They may not perform well with small datasets. In Naive Bayes, it can work reasonably well with smaller datasets and the model can be trained to study fruit maturity. Enhancing fruit maturity detection using Convolutional Neural Networks compared to the Naive Bayes Algorithm involves utilizing the strengths of each approach to achieve more accurate and robust results. The main aim of CNNs is to adapt automatically learning intricate features and patterns from raw data, especially in image processing tasks.

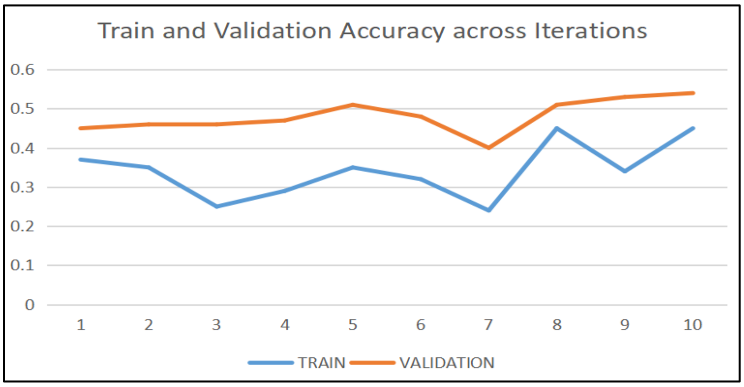

Figure 3. Training and Validation Accuracy of NB

Naive Bayes algorithms offer simplicity and effectiveness, particularly with smaller datasets, leveraging strong independence assumptions between features. This contrasts with CNNs, which excel with large datasets, automatically learning complex patterns from raw data. Combining both approaches could enhance accuracy and robustness, with Naive Bayes providing quick classifications and CNNs extracting intricate features. This hybrid strategy balances Naive Bayes efficiency with CNNs ability to learn detailed patterns, potentially yielding more precise fruit maturity detection systems.

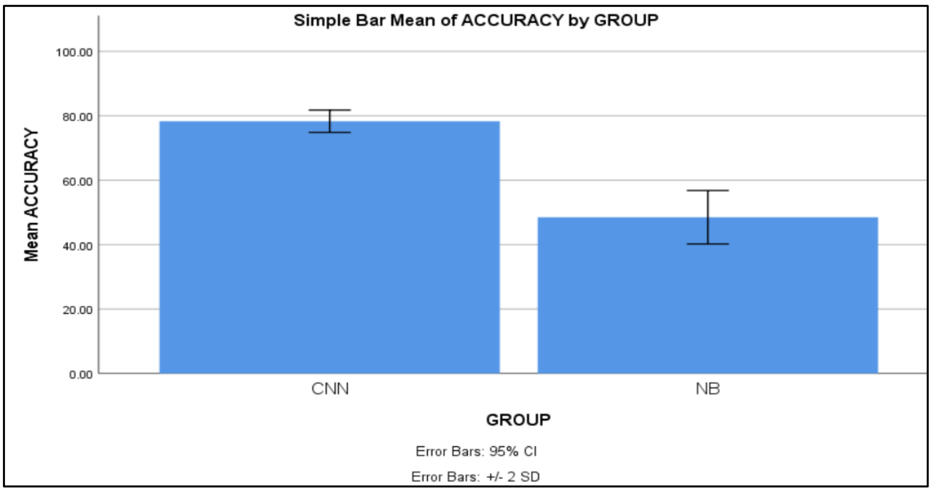

Figure 4. Mean Accuracy of CNN and NB

Figure 4 depicts a bar graph that displays the mean for efficient image sharing for Convolutional Neural Networks (CNN) and Naive Bayes (NB), obtained from t-test outputs in SPSS. The Y-axis shows the accuracy in milliseconds, while the X-axis shows the two groups. The graph contains two bars, representing the mean values of Convolutional Neural Networks and Naive Bayes obtained. The mean image for CNN was 1.2 milliseconds, while the mean for NB was 1.9 milliseconds. The mean accuracy for CNN was 2.8 milliseconds, while the mean accuracy for NB was 3.5 milliseconds. The error bars on the graph represent that the standard deviation(SD) value is of ±1SD. Based on the graph, it shows a lower average image dispersion time and augmentation time than NB. The error bars also show that CNN has a lower standard deviation for both data dispersion time and augmentation time. This means that CNN is more consistent in its performance than NB. In conclusion, CNN is a better choice for data dispersion and augmentation than the NB, especially for applications where performance is critical.

4.1 Discussion

By utilizing Convolutional neural networks (CNNs) for enhancing fruit maturity, detection achieved a notable accuracy of 81.56%, surpassing the performance of Naive Bayes, which demonstrated a lower accuracy. The significance of this difference between the two groups was established through a test, yielding a p-value of 0.048($p<0.05$).

Research on fruit detection in natural environments has shown challenges and opportunities, including variable ripening patterns, diverse fruit shapes, and environmental factors affecting sensor accuracy. To improve fruit maturity detection, advanced image processing algorithms and deep learning models trained on diverse datasets, including spectral analysis, can be integrated. Future research could focus on implementing computer vision algorithms and deep learning models, as well as exploring multispectral imaging or hyperspectral technology for a more comprehensive analysis of fruit ripeness.

5. Conclusion

This research paper highlights the potential of fruit classification, statistical features like mean, standard deviation, and significance value in the detection of fruit growth. The results contribute to the ongoing efforts to improve the accuracy of fruit maturity analysis. Enhancing fruit maturity using image classification through Convolutional Neural Networks with an improved accuracy of 81.56% shows that it is significantly better than Naive Bayes with the Zero-Knowledge proof. The Convolutional Neural Networks performed better than the Naive Bayes with accuracy. This confirms that the results achieved by CNN are superior and is statistically significant.

References

[1] Baker, Nicholas, Patrick Garrigan, Austin Phillips, and Philip J. Kellman. 2022. “Configural Relations in Humans and Deep Convolutional Neural Networks.” Frontiers in Artificial Intelligence 5: 961595.

[2] Braschler, Martin, Thilo Stadelmann, and Kurt Stockinger. 2019. Applied Data Science: Lessons Learned for the Data-Driven Business. Springer.

[3] Bridewell, Will, and Pat Langley. 2010. “Two Kinds of Knowledge in Scientific Discovery.” Topics in Cognitive Science 2 (1): 36–52.

[4] Huang, Yan, and Liang Wang. 2023. Deep Cognitive Networks: Enhance Deep Learning by Modeling Human Cognitive Mechanism. Springer Nature.

[5] Kautish, Sandeep, and Gaurav Dhiman. 2022. AI-Enabled Multiple-Criteria Decision-Making Approaches for Healthcare Management. IGI Global.

[6] Kuo, Yong-Hong, Yelin Fu, Peng-Chu Chen, Calvin Ka-Lun Or, George G. Huang, and Junwei Wang. 2022. Intelligent Engineering and Management for Industry 4.0. Springer Nature.

[7] Koirala, Anand, Keryy B. Walsh, Zhenglin Wang, and Cheryl McCarthy. "Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'MangoYOLO'." Precision Agriculture 20, no. 6 (2019): 1107-1135.

[8] McCallum, Andrew, and Kamal Nigam. "A comparison of event models for naive bayes text classification." In AAAI-98 workshop on learning for text categorization, vol. 752, no. 1, pp. 41-48. 1998.

[9] Murphy, Andrew, Matthew Skalski, and Frank Gaillard. 2018. "The Utilisation of Convolutional Neural Networks in Detecting Pulmonary Nodules: A Review." The British Journal of Radiology 91 (1090): 20180028.

[10] J. V. P, S. M. M, R. P and P. U, "A Comprehensive Study on Fruit Classification and Grading Techniques," 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2022, pp. 970-978, doi: 10.1109/ICESC54411.2022.9885492.

[11] Tumasyan, A., W. Adam, J. W. Andrejkovic, T. Bergauer, S. Chatterjee, K. Damanakis, M. Dragicevic, et al. 2022. "Search for Long-Lived Particles Decaying to Leptons with Large Impact Parameters in Proton-Proton Collisions at." European Physical Journal C: Particles and Fields 82 (2): 153.

[12] Zhang, Yudong, and Lenan Wu. "Classification of fruits using computer vision and a multiclass support vector machine." sensors 12, no. 9 (2012): 12489-12505.