# Fruit Growth Pattern Classification using Pretrained Neural Networks: A Study on Tree-based Growth

Lucia Korbeľova
Comenius University, Faculty of mathematics, physics and informatics
Bratislava, Slovakia

***Abstract*:**

**This machine learning project explores the application of machine learning techniques to classify various types of fruits based on their growth patterns. Fruits are categorized into three main growth patterns: trees, bushes, and vines. The primary objective is to develop a binary classification model that distinguishes between fruits growing on trees and those adopting alternative patterns. Leveraging the Fruits-360 dataset from Kaggle, we utilize a neural network with a pretrained visual recognition model VGG16 as its foundation. The project unfolds in three phases: hyperparameter tuning, preprocessing technique selection, and model evaluation.**

**In the hyperparameter tuning phase, a random search methodology is employed to identify the optimal configuration for layers above the pretrained base layer. The subsequent preprocessing phase involves testing multiple techniques on the dataset, culminating in the selection of two promising approaches: Gaussian blur and histogram equalization. The final phase assesses the performance of two models—one employing each preprocessing technique—on testing data. The results indicate the Gaussian blur model's superior accuracy, particularly excelling in generalizing to fruit types previously unseen during training.**

**However, challenges emerge when evaluating the models on realistic photographs containing multiple fruits and those captured in natural growth settings. Both models exhibit reduced accuracy, emphasizing the need for further refinement. Additionally, the equalize histogram model struggles with previously unseen fruit types.**

**The study contributes insights into fruit growth pattern classification using neural networks, offering potential applications in automated fruit recognition systems. The project codebase and datasets are made available on GitHub for transparency and reproducibility.**

***Keywords—neural network, fruit, growth pattern, dataset, Gaussian blur, histogram equalization***

## I. INTRODUCTION

The agricultural industry stands to benefit significantly from advancements in machine learning, particularly in tasks related to automated fruit classification. This study delves into the realm of fruit growth pattern classification, aiming to develop a robust model capable of distinguishing between fruits growing on trees and those adopting alternative growth patterns—on bushes or vines. Fruit classification has traditionally revolved around the identification of specific fruit types such as apples, bananas, apricots and so on. However, a novel exploration emerges in this project, focusing on fruit classification based on their growth patterns—a domain less explored in current machine learning literature. Unlike conventional tasks, this research aims to investigate whether a neural network can recognize common features among fruits that share their pattern of growth and is able to categorize them according to their growing patterns.

The dataset employed for this investigation, the Fruits-360 dataset from Kaggle, captures fruits in 100x100 pixel images with a white background, encompassing various angles and settings. While conventional fruit classification tasks organize fruits based on their types, this study adopts a unique approach, categorizing them into two classes: "tree" and "not tree." In the pursuit of fruit classification, this project strategically narrows its focus to binary classification, prompted by the inherent imbalance in fruit types—where the tree growth pattern dominates the dataset. This deliberate simplification addresses concerns about the adequacy of training data for recognizing fruits growing on vines and bushes. Given the limited number of fruit types within these categories and the subsequent lack of shape variability, a binary classification approach is deemed more appropriate.

## II. METHODOLOGY

### A. *Dataset*

The Fruits-360 dataset from Kaggle, originally curated for fruit recognition based on names, encompasses a diverse array of fruits and vegetables. Each item is captured in 100x100 pixel images set against a white background, providing multiple angles and rotations for each fruit and vegetable type. Notably, certain fruit types include multiple variants. For example, multiple variants of apples can be observed: red, green, golden, …

To tailor the dataset for the binary classification task—categorizing fruits based on their growth patterns—organization into three directories (tree, bush, vines) facilitated a more streamlined labeling process. This structuring allows for easy identification of a fruit's growth pattern during the loading phase, based on its directory name. In the preprocessing journey, certain adjustments were made to optimize the dataset. Images of vegetables and nuts, extraneous to the desired fruit classification, were systematically removed. For fruit types with multiple variants like apples, pears, grapes, cherries and many more specific variants were removed from the training dataset. This strategic pruning served to evaluate the model's ability to accurately categorize fruit variants in testing dataset which were previously unseen during training phase. Additionally, some fruit types, including mandarines, maracuja, and peaches, were entirely excluded from the training dataset. To assess the models ability to generalize and classify fruit types excluded from training.

A balancing act was performed by reducing the number of images depicting fruits growing on trees, addressing the inherent imbalance where tree-growing fruits significantly outnumber those of other growth patterns. This decision aimed to level the playing field in the training dataset.

As a result, the dataset underwent meticulous curation to optimize its composition for the subsequent stages of model training and evaluation.

### *B. Model Architecture:*

The choice of the pretrained VGG16 neural network as the base model in this project was driven by its proven efficacy in image-related tasks. VGG16, initially trained on the ImageNet dataset, has gained prominence for its ability to discern intricate features within images, making it a valuable foundation for image classification tasks.

VGG16 is a convolutional neural network (CNN) architecture. Its distinctive characteristic lies in its deep structure, comprising 16 convolutional and fully connected layers. The network excels at capturing hierarchical features, a quality crucial for discerning growth patterns in fruit images. By leveraging VGG16's pretrained weights, the model taps into a wealth of knowledge acquired during the ImageNet training phase, enhancing its feature extraction capabilities.

*Random Search for Hyperparameters:*

The random search approach aimed to optimize the layers added atop the pretrained VGG16 layer. The methodology involved the sequential construction of neural networks, incorporating various configurations. Each iteration in the random search explored a diverse set of hyperparameters, encompassing choices such as flattening or using GlobalAveragePooling2D, the inclusion of BatchNormalization or Dropout layers, varying dropout rates, neural network layer sizes, and the decision to add a secondary layer. Here is a concise breakdown of the random search process:

The base model, derived from VGG16 with the exclusion of the top layer, served as the foundation for subsequent modifications. The next step involved a random choice between Flatten() or GlobalAveragePooling2D(), offering flexibility in the extraction of features from the base pretrained layer. BatchNormalization and Dropout layers were selectively incorporated based on random decisions, providing regularization mechanisms to prevent overfitting and improve model generalization. The introduction of a dense layer, with a randomly determined number of units and associated activation relu function, further enriched the model's capacity to capture complex patterns within the data. Additional Dropout layers, potentially included in the architecture, contribute to regularization by preventing co-adaptation of nodes during training, fostering model robustness. The potential addition of a second dense layer, sized at half that of the initial layer, enables the model to learn hierarchical representations of features. The output layer, consisting of a single-unit dense layer with a sigmoid activation function, facilitates binary classification by determining whether some fruit grows on a tree or not.

Lastly, the optimizer selection process involves a random choice between Adam and RMSprop, with a learning rate randomly set to 0.0001 or 0.001. These choices influence the optimization algorithm's behavior during model training, impacting convergence speed and overall training performance.

The training phase consisted of fitting these models on the training dataset, evaluating their performance on the validation set, and determining the optimal configuration based on training and validation accuracy. This iterative process played a pivotal role in identifying the most effective hyperparameter combinations for the neural network architecture.

### *C. Preprocessing Techniques:*

The second phase of the project delved into the exploration and testing of various preprocessing techniques following the completion of the random search for optimal hyperparameters and neural network architecture. The primary objective during this phase was to enhance the model's performance by assessing the impact of different preprocessing techniques on input images. Six distinct techniques were evaluated, each serving as a potential enhancement to the base preprocessing method, which involved resizing the images to meet the specifications of the VGG16 pretrained model.

The base preprocessing technique served as a baseline preprocessing technique. It is a fundamental comparison point, involving the resizing of images to establish a standard for assessing the impact of additional preprocessing methods. Grayscale conversion, the first technique explored, simplified the images by reducing color information to a single channel, potentially improving computational efficiency and model simplicity. Histogram equalization, when applied to grayscale images, enhanced contrast by adjusting the distribution of pixel intensities, proving beneficial for visibility of features in images, especially under varying lighting conditions.

The Contrast Limited Adaptive Histogram Equalization (CLAHE) technique, another variant of histogram equalization, limited contrast enhancement in localized regions, preventing over-amplification of noise. This method proved effective in preserving local details and mitigating issues associated with global histogram equalization. Gaussian blur, as the fifth technique, involved smoothing the image through convolving with a Gaussian kernel, reducing noise and enhancing edges, potentially aiding in feature extraction by emphasizing prominent structures. The final technique focused on edges preprocessing, emphasizing edges in the image using techniques like Sobel filtering, which highlighted boundary information, potentially improving the model's ability to recognize shapes and contours.

Each preprocessing technique warranted dedicated model training through cross-validation (k=4) and data augmentation. The process involved fitting the model, obtained from the random search phase, on images preprocessed with one of the six techniques for 5 epochs.

Evaluation metrics, including accuracy score, precision score, recall score, and validation accuracy, were calculated for each cross-validation fold. Following this comprehensive

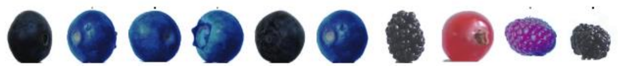

*Figure 1: base preprocessing*

*Figure 2: grayscale preprocessing*

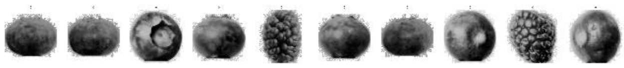

*Figure 3: histogram equalization preprocessing*

*Figure 4: clahe preprocessing*

*Figure 5: Gaussian blur preprocessing*

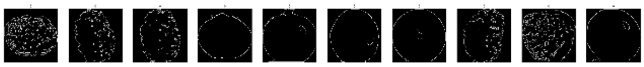

*Figure 6: edges preprocessing*

*Examples of the effect of preprocessing techniques on the images in training dataset*

evaluation, Gaussian blur and equalize histogram (applied on grayscale images) emerged as the two most effective preprocessing techniques based on their performance metrics. Subsequently, these two techniques were selected for further testing on the testing dataset to assess their generalization performance.

## III. Experiments and Results

### A. *Hyperparameter Tuning:*

In the initial phase of the experiments, hyperparameter tuning was conducted through a random search on approximately 1600 training images and around 4000 validation images. Despite the relatively small training dataset, the extensive validation set allowed for a thorough evaluation of model performance. Eight random models were trained for 10 epochs, and training and validation accuracy were collected after each epoch for analysis.

The line graphs depicting the validation accuracy evolution over epochs were instrumental in assessing model performance. The first model, ending with only 50 percent validation accuracy after the last epoch, stood out as the least successful. This result, coupled with the observation that it was the only model with a learning rate of 0.001, suggests that a smaller learning rate, specifically 0.0001, is more suitable for optimal convergence.

The third model, which exhibited a slower convergence rate and achieved around 91 percent validation accuracy after the last epoch, drew attention due to its added second layer before the final classification layer. This unique characteristic indicates that the additional layer possibly impeded the model's ability to converge efficiently.

Evaluating the accuracy curves of the remaining models revealed crucial insights into the suitability of different models. Some models exhibited irregularities and inconsistencies in their accuracy curves, even though their training accuracy values appeared consistently high. These irregularities suggested potential issues with the model architecture or hyperparameters, hindering the generalization of these models to the validation dataset. By ruling out models with erratic accuracy curves, it became apparent that a smooth and consistent convergence pattern is indicative of a well-behaved model. This process highlighted the importance of not solely relying on training accuracy but considering the overall behavior of accuracy curves to ensure robust performance on unseen data. The decision to eliminate models with irregularities contributed to selecting a model that not only excelled in training but also demonstrated strong generalization capabilities.

Models 5 and 8 emerged as strong contenders, converging rapidly and maintaining high accuracy rates. Both models utilized GlobalAveragePooling2D(), hinting at its effectiveness for the given data compared to Flatten(). Ultimately, Model 5 was selected for its simplicity, comprising GlobalAveragePooling2D(), a dense layer with 256 units and a ReLU activation function, the final classification layer with a sigmoid function and a size of 2, and the Adam optimizer with a learning rate of 0.0001. The

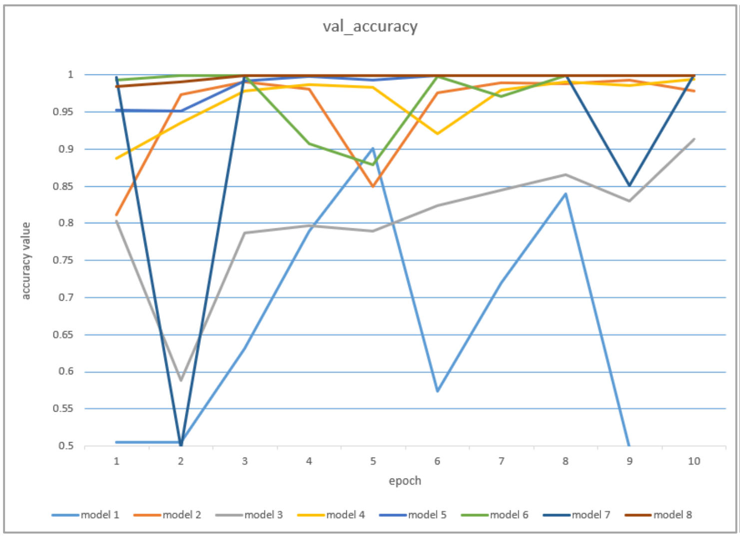


*Figure 7: Validation accuracy during 10 epochs of all models in the random search of hyperparameters*

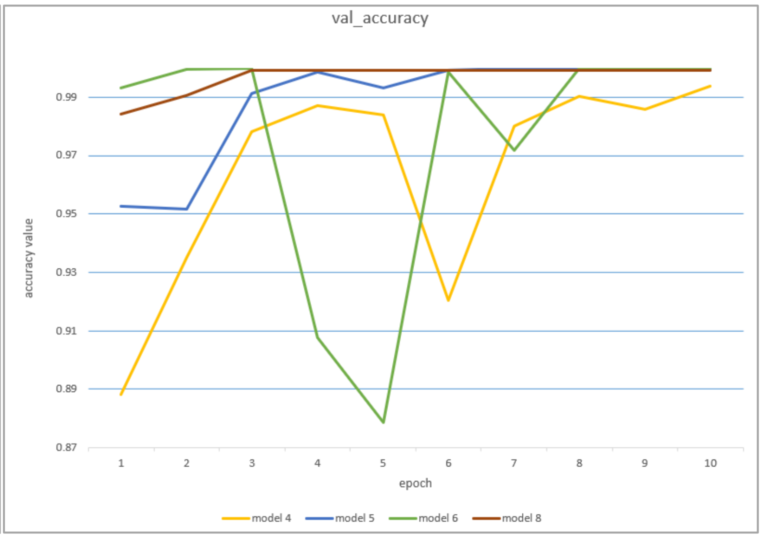


*Figure 8: Validation accuracy during 10 epochs of the four most promising models in the random search of hyperparameters*

preference for simpler models was noted, as they demonstrated greater success, and neither dropout nor batch normalization significantly contributed to the models' performance. This selection process emphasized the importance of model simplicity in achieving optimal results.

### *B. Preprocessing Techniques:*

In this phase of the project, the effects of various preprocessing techniques were investigated using cross-validation with k = 4 and training for 5 epochs. Two key tables were generated for this paper: one showcasing final validation accuracy for each of the 4 splits, and another presenting average accuracy, precision, and recall for each preprocessing technique.

The selection of these metrics was deliberate and aimed at providing a comprehensive evaluation of the model's performance. Accuracy, as a global measure, assesses the overall correctness of predictions by considering both true positives and true negatives, offering an insightful overview of the model's effectiveness across all classes. Precision, focusing on the accuracy of positive predictions, becomes crucial in scenarios where the cost of false positives is significant. In the context of fruit classification, precision aids in understanding how well the model identifies fruits that grow on trees without misclassifying other categories, with higher precision indicating fewer false positives. On the other hand, recall, or sensitivity, evaluates the model's ability to capture all relevant instances of the positive class. For fruit classification, high recall is desirable, as it signifies the model's effectiveness in recognizing all tree-grown fruits.

Notably, all preprocessing techniques exhibited high average recall, consistently above 0.92, indicating effective identification of positive instances (class tree). While this signifies proficient recognition of tree-grown fruits, potential implications of such uniformity warrant consideration.

A recurring trend observed was that models with lower performance often exhibited lower average precision, while maintaining high average recall. This suggests a tendency for the model to be less conservative in predicting positive instances, possibly leading to more false positives. This characteristic points towards a trade-off between precision and recall, suggesting that, in certain instances, the model may sacrifice specificity for sensitivity.

Edges preprocessing yielded the poorest results, with validation accuracy ranging between 0.65 and 0.8. It demonstrated the lowest average accuracy around 0.77, and its validation accuracy in individual splits and epochs did not exhibit significant improvement during fitting of the model. In some of the splits the validation accuracy had decreasing tendency after new epochs. Grayscale, the second-worst method, displayed an average validation accuracy of 0.84, with inconsistent increases and decreases in accuracy after each epoch. Notably, grayscale generated a significant amount of false positives, evidenced by its average precision of 0.78.

Similar outcomes were observed with the Clahe preprocessing technique, albeit with slightly improved average accuracy and precision.

The top three performers were base preprocessing, grayscale, and histogram equalization. However, despite the remarkable metric results, base preprocessing faced exclusion from the final preprocessing techniques due to its inconsistent validation accuracy trends. The irregularities observed in validation accuracy across epochs and splits indicated a lack of stability and consistency in the model's training, raising concerns about its ability to generalize reliably.

Gaussian blur and histogram equalization emerged as the final choices, both demonstrating consistent validation accuracy increases in cross-validation splits, indicating stable training. Gaussian blur, the clear winner, exhibited consistently high values for all three average metrics and achieved validation accuracy above 0.95 in nearly all splits. Histogram equalization, while slightly inferior, demonstrated stability. Consequently, histogram equalization was selected for comparison, allowing for an evaluation of how much better Gaussian blur performs in the final stage and assessing the significance of our results in terms of generalization.

VALIDATION ACCURACY OF PREPROCESSING TECHNIQUES IN CROSS VALIDATION

| | *Cross validation - validation accuracy after last epoch* | | | |
|---|---|---|---|---|
| *Preprocessing technique* | *1.split* | *2. split* | *3. split* | *4. split* |
| Base | 0.7309 | 0.9435 | 0.9400 | 0.9067 |
| Grayscale | 0.8771 | 0.8173 | 0.8500 | 0.8333 |
| Gaussian blur | 0.9568 | 0.9169 | 0.9533 | 0.9967 |
| Histogram equalization | 0.9169 | 0.8671 | 0.8900 | 0.8667 |
| Clahe | 0.8771 | 0.8073 | 0.8700 | 0.7667 |
| Edges | 0.7841 | 0.7342 | 0.8033 | 0.6500 |

PREPROCESSING TECHNIQUES METRICS

| *Preprocessing technique* | *Average Accuracy* | *Average Precision* | *Average Recall* |
|---|---|---|---|
| Base | 0.9317 | 0.9246 | 0.9456 |
| Grayscale | 0.8394 | 0.7856 | 0.9407 |
| Gaussian blur | 0.9559 | 0.9452 | 0.9736 |
| Histogram equalization | 0.8851 | 0.8575 | 0.9308 |
| Clahe | 0.8510 | 0.8190 | 0.9292 |
| Edges | 0.7795 | 0.7119 | 0.9489 |

### *C. Testing on Testing Dataset and Realistic Photos:*

In the final phase of the project, the two best-performing models, one utilizing Gaussian blur preprocessing and the other using histogram equalization, were subjected to rigorous training and evaluation. The models demonstrated exceptional performance on the validation datasets, achieving 0.99 and 0.98 validation accuracy, respectively. This careful training involved data augmentation, early stopping, and automatic learning rate reduction, spanning 12 epochs.

Upon testing these models on the dedicated testing dataset, both exhibited outstanding accuracy, with the Gaussian blur model reaching 99.73% and the histogram equalization model achieving 98.72%. However, the analysis of the confusion matrices and wrongly classified images (directory mislabeled in GitHub repository) revealed notable differences in mislabeling patterns.

The Gaussian blur model predominantly mislabeled fruit as tree-grown when, in fact, they were not. This was particularly evident with blueberries and strawberries, indicating potential challenges in distinguishing certain fruit shapes, especially when viewed from specific angles. These problematic angles might have made them appear similar to other fruit (plums/dark cherries, apples). The model's excellent overall accuracy on the testing dataset underscores its proficiency, despite these specific mislabeling.

On the other hand, the histogram equalization model displayed a higher number of false negatives (wrongly labeled as fruit not growing on trees). Further investigation revealed that several mislabeled instances belonged to fruit types excluded from the training set, such as specific variants of cherries, pears, and mandarins. Important thing to note is that other different looking variants of cherries and pears were included in the training. However, a salient observation arises concerning mandarins, as this particular fruit category was entirely excluded from the training data. The mislabeling instances involving numerous images of mandarins assume a heightened significance in light of this exclusion, emphasizing the model's potential limitations in generalizing to fruit types absent from its training set.

We may conclude that the model struggled more with generalization, especially when confronted with fruit types it had not encountered during training. This stands in stark contrast to the counterpart leveraging Gaussian blur preprocessing, which completely avoided mislabeling any of the unknown fruit types introduced during testing. It appears that the Gaussian blur model encountered difficulties primarily with specific angles of mislabeled fruit, yet it demonstrated proficiency in accommodating new fruit types. Consequently, the Gaussian blur model emerges as the superior choice in terms of accuracy and generalization capabilities.

CONFUSION MATRIX : GAUSSIAN BLUR MODEL WITH TESTING DATA

| | ***Predicted negative*** | ***Predicted positive*** |
|---|---|---|
| ***Actually negative*** | 2037 | 19 |
| ***Actually positive*** | 1 | 5422 |

CONFUSION MATRIX : HISTOGRAM EQUALIZER MODEL WITH TESTING DATA

| | ***Predicted negative*** | ***Predicted positive*** |
|---|---|---|
| ***Actually negative*** | 2037 | 19 |
| ***Actually positive*** | 77 | 5346 |

In the final evaluation phase, a more realistic and intricate testing dataset, comprising multiple fruit of one type in the image, often accompanied by various objects or set against a natural backdrop, was employed. The performance on this dataset was notably lower, with the Gaussian blur model achieving an accuracy of 59.09%, and the histogram equalization model registering 53.41%. While these figures may initially appear acceptable, a deeper analysis through the lens of confusion matrices and mislabeled photos unveils a concerning trend. Upon closer inspection, both models predominantly classified almost all images as negative, implying that the fruits depicted therein do not grow on trees. Specifically, the Gaussian blur model made only five accurate predictions of tree-grown fruit, while the histogram equalization model managed only two, with one being a false positive. This behavior suggests a limitation in the models' ability to generalize to more diverse and complex scenarios, potentially impacted by the absence of such instances in the training data. This highlights the importance of considering real-world scenarios and diverse datasets during model development.

CONFUSION MATRIX : GAUSSIAN BLUR MODEL WITH MORE REALISTIC AND COMPLEX TESTING DATA

| | ***Predicted negative*** | ***Predicted positive*** |
|---|---|---|
| ***Actually negative*** | 47 | 0 |
| ***Actually positive*** | 36 | 5 |

CONFUSION MATRIX : HISTOGRAM EQUALIZER MODEL WITH MORE REALISTIC AND COMPLEX TESTING DATA

| | ***Predicted negative*** | ***Predicted positive*** |
|---|---|---|
| ***Actually negative*** | 46 | 1 |
| ***Actually positive*** | 40 | 1 |

*Figure 9: Examples of images in the more complex testing dataset*

## IV. DISCUSSION

The obtained results shed light on the effectiveness and limitations of the devised fruit classification model.

Notably, the Gaussian blur preprocessing technique emerged as the superior choice, showcasing remarkable accuracy across various evaluation phases. Its robust performance on the testing dataset underscored its proficiency in recognizing unseen fruit types and complex scenarios. Conversely, the histogram equalization model exhibited slightly lower accuracy, particularly in terms of false negatives on the testing dataset. An unexpected observation was the mislabeling of previously unseen mandarins and excluded variants of cherries and pears in the histogram equalization model during the testing phase. This suggests potential weaknesses in the model's generalization ability, emphasizing the need for more comprehensive training datasets that encompass a broader array of fruit types and variants.

A consistent pattern emerged, revealing that models with inferior performance consistently displayed lower average precision, even as they maintained high average recall. This uniformity in high average recall across all preprocessing techniques, consistently surpassing 0.92, is noteworthy for its indication of the model's proficiency in identifying positive instances, specifically fruits that grow on trees. While this uniformity suggests a reliable ability to recognize tree-grown fruits regardless of the preprocessing technique, it prompts considerations regarding potential overfitting or biased learning. The model's consistent high recall may imply a strong inclination toward identifying fruits on trees, potentially at the expense of other categories. This raises concerns about the model's generalization ability to accurately classify fruits with different growth patterns. This characteristic points towards a trade-off between precision and recall, suggesting that, in certain instances, the model may sacrifice specificity for sensitivity. Therefore, while achieving high recall is crucial, striking a balance with precision becomes pivotal to ensure accurate and reliable positive predictions without introducing excessive false positives.

The model's challenges in handling more realistic and complex images, primarily labeling most instances as negative for tree-grown fruit, underscore the importance of refining the model for real-world scenarios. This behavior may stem from the absence of such diverse instances in the training data, highlighting the need for a more extensive and representative dataset that encapsulates a wider range of fruit appearances and settings.

To enhance the model's performance, future improvements could involve augmenting the training dataset with more diverse images, incorporating additional preprocessing techniques, or exploring advanced neural network architectures. Furthermore, fine-tuning the model parameters, such as adjusting the learning rate or exploring different optimization algorithms, may contribute to improved generalization.

In conclusion, while the current fruit classification model exhibits commendable accuracy, particularly with Gaussian blur preprocessing, there remain areas for refinement. Addressing these limitations and leveraging insights gained from this project will pave the way for more robust and applicable fruit classification models in real-world scenarios.

## References


[1] Dataset: https://www.kaggle.com/datasets/moltean/fruits?resource=download

[2] GitHub Repository: https://github.com/lk0103/ML_project_fruit_growth/tree/main