CNN-Based Multiclass Classification for Banana Ripeness Assessment: A Big Data Approach

*Estera Wisniewska*

*MSC Data Analytics*

*CCT College Dublin*

*Abstract*—

Keywords—

# Introduction

## Background

Bananas undergo a remarkable transformation in their nutritional composition and sugar levels as they ripen. This transformation is influenced by various factors, including the banana's inherent genetic makeup, external environmental conditions, and climatic factors. The ripening process of bananas has significant implications for their nutritional content.

As bananas ripen, their nutritional profile undergoes distinct changes. One of the most noticeable changes is the alteration in sugar levels. Unripe bananas are characterized by their high starch content, primarily in the form of resistant starch. As they ripen, this starch is gradually converted into simpler sugars, such as glucose, fructose, and sucrose. This transition from starch to sugars not only contributes to the banana's sweet taste but also makes these sugars more easily digestible and readily absorbed by the body.

Additionally, the nutritional content of ripe bananas differs from that of unripe ones. Ripe bananas tend to have a higher concentration of certain nutrients. For instance, the levels of antioxidants like vitamin C increase as the banana ripens. This means that a ripe banana may offer more antioxidant protection compared to an unripe one.

While human judgment remains a common method for assessing banana ripeness, there is an emerging technology that has the potential to revolutionize the way people select bananas at the supermarket. In the future, the ability to employ a device for immediate, on-the-spot scanning of bananas, accompanied by the instantaneous provision of comprehensive data pertaining to their nutritional composition—encompassing precise measurements of assorted nutrients and sugar content—has the potential to bestow consumers with enhanced capabilities for making judicious choices in their selection of bananas, and possibly other food as well.

## Problem Statement

The absence of readily available and accurate methods for consumers to assess the nutritional content of fresh produce, such as bananas, during the purchasing process presents a significant challenge in making informed dietary choices. This research aims to address this challenge by developing a Convolutional Neural Network (CNN) model capable of classifying banana images into three distinct ripeness categories: green, yellow, and ripe. By harnessing the power of machine learning and image analysis, this project seeks to empower consumers with a tool that not only identifies the ripeness of bananas but also considers their evolving nutritional attributes, thereby enhancing their ability to make informed and health-conscious dietary decisions.

## Research objective

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type | Sugar (g) | Protein (g) | Calories | Fiber(g) | Fat(g) |
| Green | 12 | 1.2 | 89 | 2 | 0.5 |
| Yellow | 12 | 1.1 | 100 | 2.6 | 0.3 |
| Ripen | 15 | 1 | 110 | 2.9 | 0 |

# Literature Review

## Cnn architecture for a big data

## Advanced Data Analysis

## Abbreviations and Acronyms

# Methodology

In recent years, Convolutional Neural Networks (CNNs), have achieved unprecedented results, rivaling human-level performance in various tasks. The unique feature of CNNs is their explicit assumption that input data, such as images, possesses inherent structure, allowing them to efficiently encode this property into their architecture by sharing weights across image locations and facilitating local responsiveness of neurons (Jonas Teuwen,Nikita Moriakov, 2020). CNNs have emerged as a powerful tool, particularly in the domain of computer vision, due to their ability to harness spatial relationships within structured data.

In this research, the CNN model was used for its remarkable capability to handle image data and to classify banana pictures. CNNs excel at automatically extracting pertinent features from images, allowing them to discern variations in texture, colour, and shape associated with different ripeness levels (green, yellow, ripe). Moreover, their ability to generalize from visual data makes CNNs a potent tool for precise and efficient banana classification.

## Colour channels

CNNs process images through colour channels, typically the red (R), green (G), and blue (B) channels in RGB images. Each channel represents different colour information. For instance, the R channel primarily carries red and yellow information, the G channel emphasizes green and yellow, and the B channel highlights blue and yellow. By analyzing these channels, a CNN can capture critical colour cues specific to banana ripeness, ensuring that colour-related features are considered during the classification process. This capability is essential in accurately distinguishing between the varying stages of banana ripeness based on their colour characteristics.

In the process of converting colour channels to grayscale, it becomes evident that high values within these channels signify the predominance of the respective colour, while low values indicate its attenuation. Grayscale images, in essence, serve as representations of colour intensity, with white serving as a symbol of heightened intensity and black as a representation of reduced intensity.

A close up of a ring

Description automatically generatedA banana with a black background

Description automatically generated with medium confidenceA banana with a banana in the middle

Description automatically generated with medium confidence

An examination of these grayscale representations of individual colour channels permits a comprehensive exploration of each colour's contribution to the overall visual composition of an image. Instances, where regions appear predominantly white in the grayscale renderings, signify the predominance of that specific colour in those particular areas, while darker regions imply a decreased presence of that colour. This analytical approach proves highly beneficial for image assessment and aids in understanding the distribution of colours within the image's composition.

## Data collection

## Data Preparation

To initiate the CNN project, data loading and preprocessing are essential. The initial step employs the function load\_and\_preprocess\_image, which accomplishes several pivotal tasks. Firstly, it employs the OpenCV library (cv2) to read image files, a robust choice given its efficiency and wide acceptance in the computer vision community. Secondly, the function standardizes the images by resizing them to a fixed dimension of 960x540 pixels. This resizing operation ensures uniformity in the dimensions of all input images, which is indispensable for training a CNN model. Images of varying sizes can introduce complexities in terms of model architecture and hinder convergence during training.

Thirdly, the function scales the pixel values to fall within the range of 0 to 1. This scaling process transforms the pixel intensities from the original 8-bit integer representation to a floating-point representation between 0 and 1. This common practice offers several advantages. It prevents numerical instability during training, facilitates faster convergence, and enables the model to generalize better to new, unseen data.

The second part of data preparation was categorizing the dataset into distinct classes and assigning appropriate labels. The code accomplishes this task by systematically navigating through files (Green, Yellow, Ripe) and labeling images based on their names. The labels are assigned numerically, with 'Green' corresponding to 0, 'Yellow' to 1, and 'Ripe' to 2.

This labeling scheme is a product of careful consideration, aligning with best practices in supervised classification tasks. By assigning numerical labels, the model can effectively learn to associate image features with specific banana categories. Moreover, this numerical representation simplifies the loss calculation and model evaluation processes, as many machine learning frameworks expect numerical labels.

## Creating CNN model

Effective dataset management is paramount in machine learning to ensure unbiased model evaluation and to guard against overfitting. The code aptly divides the dataset into training, validation, and test subsets using the train\_test\_split function.

The 70-15-15 split ratio is a well-established convention in machine learning experiments. Here, 70% of the data is allocated to the training set, 15% to the validation set, and 15% to the test set. The reasoning behind this division is twofold. First, the training set provides the model with sufficient diversity to learn meaningful patterns, while the validation set is used for fine-tuning hyperparameters and monitoring training progress. Finally, the test set serves as an independent benchmark to evaluate the model's performance. This separation of data ensures that the model's accuracy is a true reflection of its generalization capability.

The central component of this CNN project is the meticulously designed model architecture. The initial convolutional layer with 32 filters, each employing a 3x3 kernel, is instrumental in detecting local patterns within the images. The choice of 32 filters is based on empirical observations and experimentation, striking a balance between model complexity and efficiency.

Subsequent to the convolutional layer, a max-pooling layer with a 2x2 pooling size is employed. Max-pooling plays a vital role in dimensionality reduction, which is pivotal in mitigating computational burden and introducing translation invariance.

The subsequent flattening layer reshapes the data, transforming the two-dimensional feature maps into a one-dimensional vector. This step is fundamental before feeding the data into fully connected layers.

The two fully connected dense layers further process the feature vector. The first dense layer with 64 units uses the Rectified Linear Unit (ReLU) activation function, which introduces non-linearity and facilitates feature transformation. The final dense output layer consists of three units, each corresponding to a fruit category. The softmax activation function ensures that the output represents a probability distribution over the three classes, making it suitable for multi-class classification.

The architecture adheres to well-established CNN design principles for image classification. Convolutional layers are adept at capturing hierarchical features, and fully connected layers integrate these features to make classification decisions.

The successful construction of a CNN model culminates in model compilation. This step involves specifying critical elements such as the optimizer, loss function, and evaluation metric.

The Adam optimizer was chosen sue to its renowned for its efficiency and effectiveness in optimizing deep neural networks. Its adaptive learning rate ensures convergence even in complex optimization landscapes, which is invaluable in training deep models.

The sparse categorical cross-entropy as the loss function aligns with the multi-class classification nature of the problem. This loss quantifies the dissimilarity between predicted probabilities and true labels. It encourages the model to make predictions that are close to the true class labels, facilitating accurate classification.

## Training the model and the Evaluation

The final phase of the project involves training the CNN model. This process includes feeding the model with the preprocessed training data, specifying batch size and epochs, and validating the model's performance on the validation set.

The training data consists of images after preprocessing, ensuring they are in a format compatible with the model's architecture. The use of batch processing, with a batch size of 32, enhances training efficiency. This approach allows the model to update its parameters using mini-batches of data, which often results in smoother convergence and efficient use of computational resources.

The choice of ten epochs for training reflects a balance between model convergence and training time. Empirical observations and experimentation determined this value, but it can be adjusted based on specific requirements.

# Big Data

## 

## Interpretation

## Limitations

## Suggestions for Future Reaserch

# Conclusion

##### References