MEAT QUALITY PREDICTION USING MACHINE LEARNING

Rohit Buddiga

MEAT QUALITY PREDICTION USING MACHINE LEARNING

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Information Systems and Technology

by

Rohit Buddiga

May 2023

MEAT QUALITY PREDICTION USING MACHINE LEARNING

A Project

Presented to the

Faculty of

California State University,

San Bernardino

by

Rohit Buddiga

May 2023

Approved by:

Dr. Conrad Shayo, Committee Chair

Dr. Kamvar Farahbod, Committee Co-Chair

Dr. Conrad Shayo, Department Chair, Information and Decision Sciences

ABSTRACT

Meat quality is an essential aspect of the food industry. However, traditional methods of meat quality prediction have limitations in terms of accuracy, cost, and time efficiency. This project focused on utilizing advanced Deep learning and Machine learning algorithms to develop- machine learning models that could predict the freshness (or spoilage) of meat with a 100% accuracy, based on image data. In addition to accuracy, this study emphasizes the significance of speed and time in selecting the optimal machine learning model. The research questions are: Q1. What hybrid neural networks should be used to predict freshness? Q2. How do hybrid neural networks determine the freshness of the meat based on the image? Q3. How can accuracy and performance speed be improved? A dataset from the Kaggle repository was used to explore various machine learning algorithms such as Support Vector Machines, Decision Trees, and Random Forests with a combination of Convolutional Neural Network, a deep learning network. The findings are: Q1. A combination of Support Vector Machines-Convolutional Neural Network, Decision Trees-Convolutional Neural Network, and Random Forests-Convolutional Neural Network were used to predict freshness. 2) The hybrid neural networks were trained using the tensorflow.keras.models, a high-level neural networks API of the TensorFlow library, which allowed the creation and training of complex machine learning models in a simple and straightforward manner. 3) The accuracy and performance speed of the model can be improved by utilizing a distributed computing environment for training, which involves the collaboration of multiple machines to carry out computations. The conclusion from our project is that Utilizing the hybrid neural networks developed, it is possible to classify meat products as either fresh or spoiled using image analysis. This approach not only reduces the reliance on human input for meat classification but also decreases the time taken to complete the classification process. Furthermore, emerging areas for future research that emerged from this study is to develop machine learning models that can integrate and fuse multi-modal data such as genetics, feeding and processing techniques to make more accurate predictions of meat quality.

DEDICATION

Dedicated to my beloved mother.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

Meat is one of the most important sources of protein for human nutrition, and its consumption has been increasing over the years due to population growth, economic development, and changes in dietary patterns (FAO, 2021). As demand increases, concerns have emerged around the industry's ability to effectively manage meat production and ensure quality control. Therefore, it becomes imperative to monitor these aspects strictly to ensure the sustainability and safety of the meat industry.

Most of the operations are carried out by machines in the meat industry. Machines must show a level of perfection that precludes errors in any given circumstance. This is of paramount importance, as any form of spoilage in meat products presents a significant risk to the health and wellbeing of consumers (Lorenzo J.M et al., 2017).

Consumers have high expectations for the quality and safety of meat products, and manufacturers and retailers are under pressure to provide superior-quality items. Freshness is a critical factor that influences a consumer's decision to purchase meat and is essential in assessing its quality and safety. Therefore, it is crucial to develop effective techniques for evaluating meat freshness to meet customer demands and increase the commercial value of livestock. Microbial deterioration and metabolic processes cause meat to lose freshness over time, making it necessary to assess its freshness promptly.

There are numerous ways to measure freshness decline. To determine the freshness of meat, traditional analytical techniques employ sensory evaluation, chemical analysis, and microbial population evaluation (Amin Taheri-Garavand et al.,2019). These techniques can harm the meat samples throughout the pre-treatment procedure and take a lot of time to implement. Also, the experimenter's expertise significantly impacts the accuracy of the measurements and outcomes.

Recent advances in machine learning techniques have provided an opportunity to develop an objective and automated approach for meat quality assessment (B.W. Penning et al., 2020). A subset of artificial intelligence referred to as machine learning allows systems to improve their performance by learning from data. By training machine learning algorithms to identify patterns in data, predictions can be made based on that information. Machine learning has been successfully applied in various fields, including image and speech recognition, natural language processing, and medical diagnosis (LeCun et al., 2015).

Problem Statement

Meat quality assessment is a key factor in ensuring consumer satisfaction and health. However, traditional methods of meat quality assessment are often subjective and time-consuming (Amin Ta et al., 2019). As global demand for meat products continues to rise, the need for objective and automated meat quality assessment becomes increasingly pressing and machine learning presents a promising solution to address this challenge. Convolutional Neural Networks (CNNs) have long been considered a promising solution for developing meat quality rating systems, as evidenced by numerous studies. CNNs have been widely applied to evaluate meat quality attributes, including tenderness, color, and mosaic structure. The automatic assessment of meat quality with image processing and computer vision techniques is an active research area that contains many practical issues. This study aims to identify the most relevant features for meat quality assessment, develop an accurate machine learning model for meat quality assessment using hybrid neural networks and evaluate the performance of different machine learning algorithms for meat quality assessment. The findings of this research can provide guidance for the food industry in improving meat quality and ensuring consumer satisfaction and health. The study will address the following major questions:

1. What hybrid neural networks should be used to predict freshness?
2. How do hybrid neural networks determine the freshness of the meat based on the image?
3. How to improve accuracy and increase the performance speed of the proposed model?

Objectives

The objective of this project is to utilize advanced AI and ML algorithms to address the challenges revolving around the quality control aspect of the meat production process and to contribute to the areas of further research related to enhancing meat quality with optimal precision and efficiency using neural networks.

Organization of the Project

This project will be organized in the following way:

Chapter 2 reviews the literature and related work.

Chapter 3 introduces the types of machine learning algorithms used to build the model.

Chapter 4 is a discussion of the data collection and analysis, and

Chapter 5 includes the discussion, conclusion, and areas for further study.

CHAPTER TWO

LITERATURE REVIEW

This review aims to identify gaps in current knowledge and highlight the most relevant and significant findings related to the research question: Q1. What hybrid neural networks should be used to predict freshness? By examining and analyzing previous studies, the literature review helps to contextualize the research and establish the significance and originality of the proposed study.

A study by O. Oulcan et al., (2019) discussed the negative perception of red meat that could occur under unsuitable sales conditions, such as exposure to heat or poor storage, resulting in a negative impact on consumer perception. Additionally, it highlighted the loss of nutrients and formation of harmful microorganisms that could occur under such conditions, which posed a risk to human health. The study focused on monitoring the quality of a tray of meat cubes, which is a commonly sold product in the retail sector. To accomplish this, the authors used a stable camera to capture RGB images of the tray of meat cubes at regular intervals of two minutes, which allowed the researchers to track any changes in quality over time. Expert data was also collected and used as reference labels to help train the deep convolutional neural network architecture. The images acquired by the camera were pre-processed, and the deep learning model was trained to classify them as either "fresh" or "spoiled." The study found out that deep learning methods are successful in this research field, as evidenced by the experimental results and comparisons conducted. Their research is more focused on deep learning, which is a part of our project, but they do not answer the main purpose of our project which is to find out what hybrid neural networks are used to assess the quality of meat.

Vinda Setya Kartika et al., (2018) proposed a novel approach for detecting spoiled meat utilizing semiconductor gas sensors and image processing methods. The traditional method of detecting spoiled meat involved using human senses of sight and smell, which can be subjective and potentially dangerous due to bacterial contamination. To get over these constraints, the researchers employed a camera with image processing utilizing a Grey Level Co-Occurrence Matrix and a gas sensor array to identify gases released by decaying meat. Artificial Neural Networks (ANN) were used to analyze the responses from the gas sensor array and the Grey Level Co-occurrence Matrix to classify the degree of spoilage. The proposed method achieved a high percentage of success, with an accuracy of up to 82%. This meant the ANN could accurately classify the level of spoilage in the meat based on the gas sensor and image data. This method had the potential to replace the role of human senses in meat classification, making the process faster, more objective, and safer.

A study by Calvin, Ghiri Basuki Putra, and Esa Prakasa (2020) discusses the quality of chicken meat in Indonesia, as it is one of the most consumed meats in the country. To classify the meat, the authors employed convolutional neural networks and utilized images captured from a smartphone in RGB format, which were then converted into binary format for analysis. The data set consisted of two classes: fresh and rotten. Prior to cropping, the Otsu technique with thresholding and RGB-to-binary image conversion was used to choose the appropriate region of the images for the chicken meat image. After being cropped into three different sizes, the images were utilized as a dataset for the research. The chicken meat image dataset was trained using a self-created, straightforward architecture called Ayam6Net, while the AlexNet, VGGNet, and GoogLeNet architectures were also utilized for comparison. The highest accuracy was attained by Ayam6Net, with 92.9%. The experiment's findings led the authors to the conclusion that, in comparison to other architectures and image dataset sizes, the Ayam6Net architecture and a 400x400 pixel dataset produced greater accuracy results.

The objective of M. Guermazi et al. (2013) was to create a classification system for a novel technique of meat analysis that could furnish details on the condition of vacuum-packed meat being sold in supermarkets. The researchers used a supervised training approach with neural networks based on the back error propagation method. During the training, the primary goal was to attain accurate classification, enabling the correct categorization of inputs as well as any erroneous inputs not present in the database without introducing new categories. The classification methodology involved grouping the model parameters of the meat's physical model based on a database that incorporated the model parameters for distinct beef muscles on different days. By using this approach, the researchers aimed to develop a method that could accurately classify the state of vacuum-packed meat based on its physical characteristics.

To answer the research questions: Q2. How do hybrid neural networks determine the freshness of the meat based on the image? Q3. How to improve accuracy and increase the performance speed of the proposed model? We've done the following literature review.

The work by Erika Carlos Medeiros et al. (2018) provides an approach for using digital images to inspect the meat of tuna and salmon. The proposed method consists of hardware as well as a series of steps for pre-processing the images and obtaining relevant parameters from multiple color spaces. The datasets that were created were then used in studies with machine learning classification techniques to ascertain the levels of freshness in the fish samples. The accuracy, receiver operating characteristic curve, precision, recall, f1-score, and confusion matrix were some of the metrics used by the authors to evaluate the effectiveness of AutoML models in categorizing the degree of freshness of tuna and salmon samples. The outcomes demonstrated that the AutoML-generated ensembles for salmon and tuna both attained 100% in all metrics. The authors emphasized the ease of use of their proposed solution in various contexts. The authors were able to demonstrate the feasibility of detecting the external quality of tuna and salmon meat products through the utilization of non-destructive methods, employing computer vision and machine learning techniques. The high accuracy of the experiments highlighted the efficiency, objectiveness, consistency, and reliability of this method.

In a study conducted by B.W. Penning, W.M. Snelling, et al., (2020), two different artificial intelligence methods were compared for automating carcass quality grading. Image analysis was used in the first method, while the cutting-edge Rapid Evaporative Ionization Mass Spectrometry was used in the second. Both approaches made use of machine learning (ML) to improve the efficiency and precision of carcass quality assessment. For all meat quality measurements other than marbling, it was discovered that the image analysis approach was quicker and more precise than human meat inspectors. On the other hand, the mass spectrometry technique was put to the test using eight ML algorithms and was able to predict carcass quality attributes with an accuracy range of 81.5% to 99%. I t was discovered that the accuracy of the ML algorithms depended on the particular feature being examined, indicating that this technique may not be appropriate for all traits. Overall, the study found that both image analysis and mass spectrometry methods, utilizing machine learning algorithms, could enhance the speed and accuracy of carcass quality grading. However, the effectiveness of these methods varied depending on the specific trait being examined. These findings suggest that a combination of different AI-based methods may be necessary to achieve accurate and efficient carcass quality grading. Their study provides us with the area of further research for our project which is How do hybrid neural networks determine the freshness of the meat based on the image.

The literature review presents studies conducted to assess the quality of meat products using deep learning and image processing techniques. The meat industry is constantly seeking innovative methods to assess the quality of their products and ensure consumer safety. Deep learning and image processing techniques have emerged as promising tools in this endeavor, and this literature review is replete with studies that validate this claim. From detecting spoilage to enhancing the overall quality of meat products, these methods have been employed to significant effect. The research outcomes show the reliability of various machine learning algorithms proposed for meat quality assessment. As a result, this study aims to leverage the insights gained from these findings to develop a more optimized and effective model which is a hybrid neural network.

CHAPTER THREE

RESEARCH METHODS

To answer our first research question, Q1. What hybrid neural networks should be used to predict freshness? We are using hybrid neural networks for meat classification based on images. The idea of using a hybrid neural network comes from previous studies which we have covered in our literature review. These studies have demonstrated the effectiveness of neural networks in meat classification.

Furthermore, we are using a hybrid neural network because it combines the strengths of different neural network architectures (Zhao R et al., 2022). For instance, we would be using convolutional neural networks (CNNs) for feature extraction. Additionally, we would be using SVM, decision trees, Random Forest for classification. The use of a hybrid neural network approach allows us to create a more robust and accurate model for meat classification (Zhao R et al., 2022). We introduce the following Hybrid Neural Networks algorithms used in classification of images; SVM+CNN, Decision Trees + CNN and Random Forest + CNN. We also list some of the advantages and disadvantages of using these algorithms.

Hybrid Neural Networks for Meat Quality Prediction

SVM-CNN: Vapnik et al. (1995) developed the support vector machine (SVM) for binary classification. “Its objective is to find the optimal hyperplane $f(w, x) = w \cdot x$

+ b to separate two classes in a given dataset, with features x ∈ R m.” Vapnik et al., (1995). “Convolutional Neural Network (CNN) is a class of deep feed-forward artificial neural networks which is commonly used in computer vision problems such as image classification”, (Abien et al., 2017). A Support Vector Machine (SVM) + Convolutional Neural Network (CNN) hybrid neural network is a combination of two powerful machine learning techniques used for classification tasks (Duggani Keerthana et al., 2022). CNN performs the task of extracting features from unprocessed data, such as images, whereas the SVM is utilized for the purpose of classifying the extracted features. CNN first learns the features from the input images, then the extracted features are passed on to the SVM for classification. Szarvas et al., (2005) studied the automatically optimized CNN features and demonstrated that the highest testing accuracy could be produced by combining CNN with SVM.

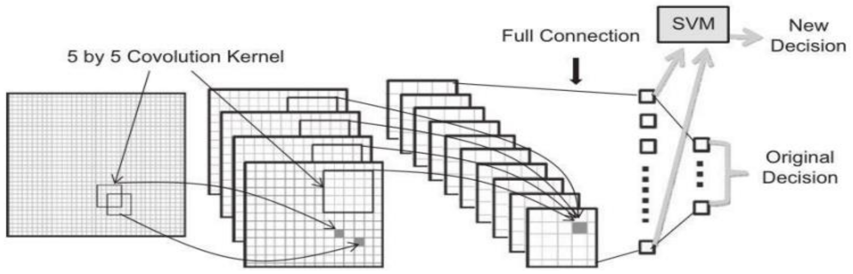


Figure 1: Architecture of CNN - SVM model (Haotian 2018)

Advantage of SVM-CNN: Combines the advantages of both SVM and CNN by using SVM for classification and CNN for feature extraction. Can achieve high accuracy and generalize well.

Disadvantage of SVM-CNN: Requires a lot of computational power and can be slow to train.

DT-CNN: The Decision Tree (DT) algorithm was first proposed by Ross Quinlan in 1986. “A simple decision tree model includes a single binary target variable Y (0 or 1) and two continuous variables, x1 and x2, that range from 0 to 1”, (Song YY et al.,2015). Nodes and branches constitute the majority of a decision tree model, while splitting, stopping, and pruning are the major building procedures. (Song YY et al.,2015). This model combines the ability of Decision Trees (DTs) to classify data with the ability of CNNs to extract features from images to achieve better accuracy in image classification tasks (Xu S et al., 2021).

This study uses CNN to extract features from the images, which are then fed into the DT for classification This helps to identify the most significant factors that affect meat quality and create a decision-making framework for meat quality control in the industry.

Advantage of DT-CNN: The model is easy to interpret, making it useful for gaining insights into the data and understanding the decision-making process.

Disadvantage of DT-CNN: The performance of the model can be affected by the quality and quantity of data used for training.

Using decision trees as a foundation, a more potent algorithm was formed: Random Forest

RF-CNN: Leo Breiman and Adele Cutler developed the Random Forest (RF) algorithm in 2001. This model combines the ability of Random Forests (RFs) to classify data with the ability of CNNs to extract features from images to achieve better accuracy in image classification tasks (Boston T et al., 2022).” It is possible to input all the samples from the training dataset {(X1 !, y1), (X2 !, y2),...,(Xn !, yn)} into the network and obtain the corresponding output of F4 layer Fx 1 ,Fx 2 ,Fx 3 , ... Fx n along with its corresponding label y 1 ,y 2 ,...,y n. These outputs and labels can be used to create a new training dataset {(Fx 1 ,y1), (Fx 2 ,y2), (Fx 3 ,y 3) ...... (Fx n ,y n)}. This new dataset can be efficiently used to train a random forest model”, (Li T et al., 2019).

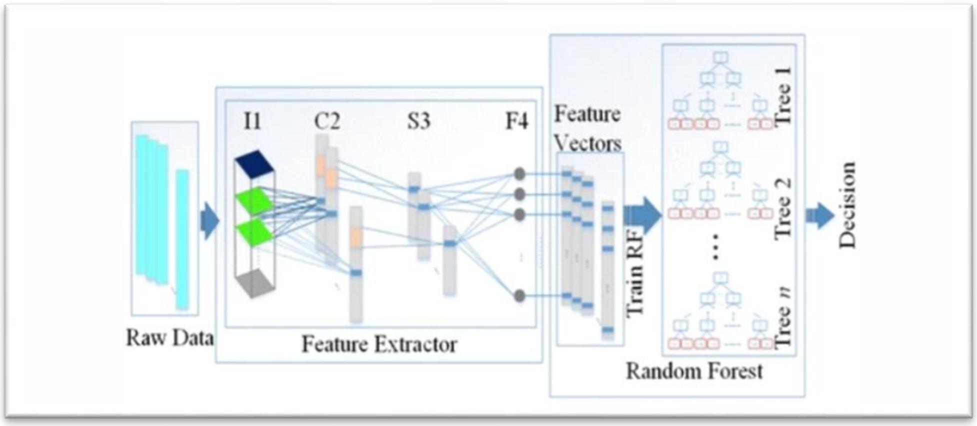


Figure 2: Architecture of CNN – Random Forest model (Li T 2019)

By analyzing many meat quality attributes and their relationships, RF can identify the most important factors that affect meat quality (Kircali Ata S et al., 2023). To leverage this benefit, the RF algorithm has been deployed in this study.

Advantages of RF-CNN: Random forests can handle non-linear and non-monotonic relationships in data and can be more accurate than decision trees. Can achieve high accuracy and generalize well.
Disadvantage of RF-CNN: The model may not generalize well to new or unseen data if overfitting occurs during training.

To answer our questions: Q2. How do hybrid neural networks determine the freshness of the meat based on the image? Q3. How to improve accuracy and increase the performance speed of the proposed model? We use Python and Supervised learning which is introduced below.

Python Libraries

Python and Libraries: Python has emerged as a highly popular high-level programming language for scientific computing purposes, owing to its interactive nature and the availability of powerful libraries. These libraries include Scikit-learn, NumPy, Matplotlib, TensorFlow and Pandas, all of which have profoundly impacted Data Science (Raschka S et al., 2020).

An open-source machine learning toolkit called Scikit-learn offers a variety of effective machine learning techniques. These methods include,” data transformation, supervised and unsupervised learning, feature selection, and model evaluation, all of which are fundamental topics related to machine learning”, (Hao & Ho et al., 2019). NumPy is an essential library for machine learning in Python, providing functions for data representation, data preprocessing, linear algebra, random number generation, and evaluation metrics. Its efficient array operations and mathematical functions make it a vital tool for machine learning applications. Matplotlib is an essential library for data visualization in machine learning, providing functions for data visualization, model evaluation, hyperparameter tuning, debugging, and reporting. An open-source library for numerical computing and extensive machine learning is named TensorFlow. Pandas is a library for data manipulation in machine learning, providing functions for data manipulation, cleaning, merging, and joining, time series analysis, and feature engineering.

Supervised learning is a type of machine learning that involves training a model using a set of labeled examples or datasets (Badillo S et al., 2020). The labeled dataset comprises both input features and their corresponding output targets, known as labels. During the training phase, the machine learning algorithm learns from the labeled data to identify patterns or relationships between the input features and their corresponding output targets. Once the algorithm has been trained on the labeled dataset, it is evaluated on a new dataset, which is referred to as the testing dataset, to measure its accuracy. The testing dataset is an unlabeled dataset that contains input features without their corresponding labels. The model predicts the output labels for the testing dataset, and its accuracy is measured by comparing its predictions with the actual labels in the dataset. If the model's accuracy is satisfactory, it can be used for real-world applications. However, if its accuracy is low, the model can be fine-tuned to improve its performance. Fine-tuning the model involves adjusting its hyperparameters, modifying the features, or using a different algorithm altogether to improve its accuracy on the testing dataset.

We decided to move forward using the following Hybrid Neural Networks algorithms used in classification of images; SVM+CNN, Decision Trees + CNN, Random Forest + CNN. After searching various repositories such as UCI, Kaggle, OpenML, Data.world, and Data.gov, We got the Meat Quality Prediction dataset from the Kaggle repository (O. Oulcan et al., 2019). We are using this dataset as the images are very clear and the dataset provides us with a rich source of information that can be used to assess meat quality accurately and efficiently. The dataset available comprises two distinct categories - fresh and spoiled red meat samples. These samples were collected from a supermarket located in Izmir, Turkey. The dataset comprises 948 fresh images and 948 spoiled images that are accessible through the code. Two distinct features exist within the dataset: the first being the image in .jpg format, while the other feature pertains to the target category. The target feature contains two categories: Fresh and Spoiled.

The process of building the models can be broken down into four fundamental steps. Initially, the data is read into the system. Subsequently, appropriate models are chosen based on the data type and desired outcome. Following this, the model is fitted and utilized to make predictions on the data. Finally, the model's accuracy is evaluated to ensure that it satisfies the desired level of performance. The following chapter will cover the data analysis part of the project.

CHAPTER FOUR

DATA ANALYSIS

In this chapter, we are going to analyze the data to answer our research questions: 1. What hybrid neural networks should be used to predict freshness? 2. How do hybrid neural networks determine the freshness of the meat based on the image? 3. How to improve accuracy and increase the performance speed of the proposed model?

Dataset Analysis & Results

To answer our first question, what hybrid neural networks should be used to predict freshness? The following analysis was done.

The project involved using Google Colab to create machine learning models. Colab is a cloud-based platform that allows users to write and execute code in a Jupyter Notebook-style environment.

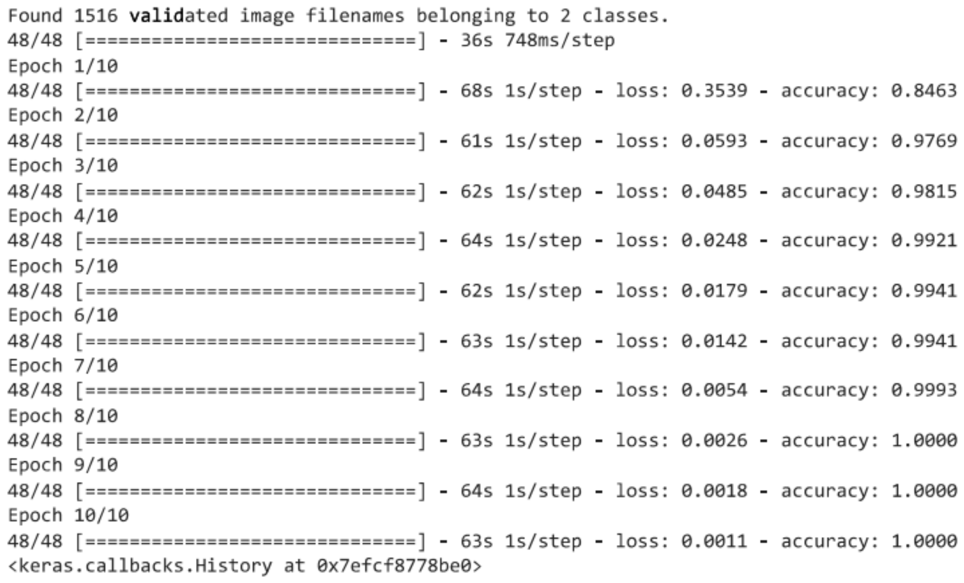
The first step in the project was to import essential modules such as Pandas, NumPy, Scikit-learn SVM, Decision Tree, Random Forest, and Tensorflow. These modules provide a range of functionalities for data analysis, visualization, and modeling.

The next step involved reading the paths of the images stored in a folder named "Meat_data" and storing them in a list variable called "JPG_Path". To convert the list into a Pandas data frame, the Pandas module was used. Then Image resizing was done. In computer vision, resizing images is a crucial pre-processing stage that significantly impacts the training of deep learning models. This is primarily because smaller images enable faster training of the models. After the images were properly resized, they were trained using the tensorflow.keras.models.

To split the data into training and validation data frames with an 80:20 ratio, the "train_test_split" function was employed. This function randomly splits the data into training and validation sets based on a user-specified percentage. In this case, the data was split into training and validation data frames with a ratio of 80:20, respectively.

Finally, a hybrid SVM+CNN model was built with an activation function of ReLU and sigmoid which enable nonlinear expression in neural networks, allowing them to suit the outcomes more accurately. An epoch of 10 was used to train the model. For the SVM part of model, we have used linear kernel for the implementation of SVM classifier because using this classifier, it will be prone on robustness to outliers, and they are good for large datasets as they are computationally efficient which altogether increases the generalization process.

The SVM+CNN hybrid model combines the strengths of SVM (Support Vector Machine) and CNN (Convolutional Neural Network) to achieve high accuracy in image classification tasks. Mathematical equations are employed as activation functions to calculate the output of a neural network. The epoch refers to the number of times the complete dataset is utilized to train the neural network.

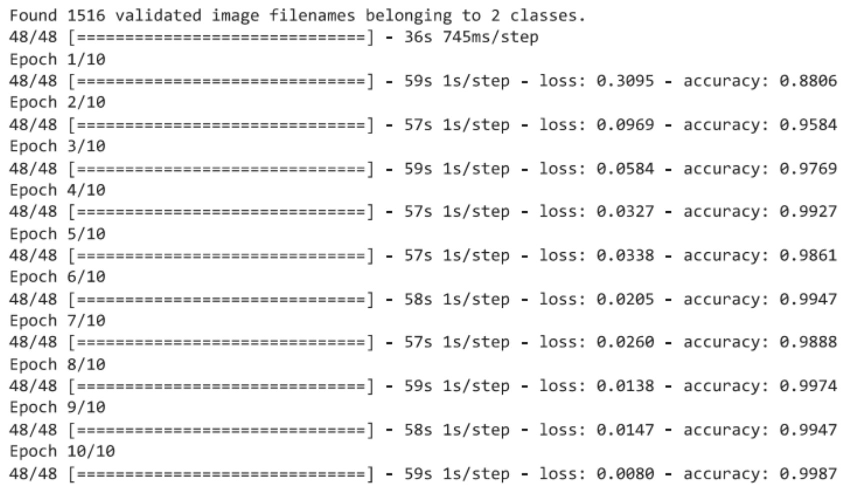


```
Found 1516 validated image filenames belonging to 2 classes.
48/48 [==============================] - 36s 748ms/step
Epoch 1/10
48/48 [==============================] - 68s 1s/step - loss: 0.3539 - accuracy: 0.8463
Epoch 2/10
48/48 [==============================] - 61s 1s/step - loss: 0.0593 - accuracy: 0.9769
Epoch 3/10
48/48 [==============================] - 62s 1s/step - loss: 0.0485 - accuracy: 0.9815
Epoch 4/10
48/48 [==============================] - 64s 1s/step - loss: 0.0248 - accuracy: 0.9921
Epoch 5/10
48/48 [==============================] - 62s 1s/step - loss: 0.0179 - accuracy: 0.9941
Epoch 6/10
48/48 [==============================] - 63s 1s/step - loss: 0.0142 - accuracy: 0.9941
Epoch 7/10
48/48 [==============================] - 64s 1s/step - loss: 0.0054 - accuracy: 0.9993
Epoch 8/10
48/48 [==============================] - 63s 1s/step - loss: 0.0026 - accuracy: 1.0000
Epoch 9/10
48/48 [==============================] - 64s 1s/step - loss: 0.0018 - accuracy: 1.0000
Epoch 10/10
48/48 [==============================] - 63s 1s/step - loss: 0.0011 - accuracy: 1.0000
<keras.callbacks.History at 0x7efcf8778be0>
```

Figure 3: Training model of SVM + CNN Hybrid Neural Network

The model was trained using hyperparameter tuning, which involved optimizing parameters such as kernel function, number of iterations, and number of epochs to improve accuracy. During training, the accuracy of the model improved with each epoch. After the final iteration, the model was tested using a randomized test dataset, resulting in a final accuracy score of 100%.

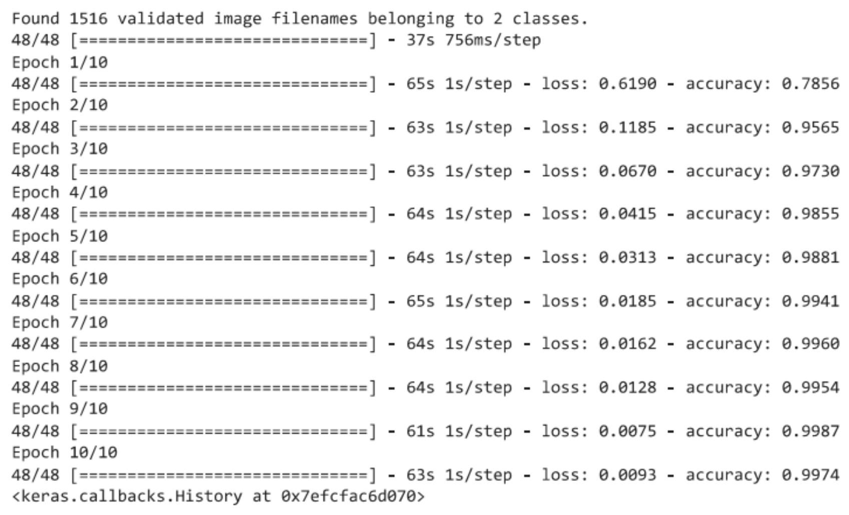
The Second Hybrid model is a combination of Decision Tree + CNN. We developed a CNN architecture using the Keras library in Python. The architecture consisted of several convolutional and pooling layers, followed by fully connected layers and SoftMax as a output layer. We used categorical binary cross-entropy as the loss function, and Adam as the optimizer. For activation we used reLU (ReLU is a non-linear activation function that introduces non-linearity into the CNN model, allowing it to learn more complex features and improve accuracy).



```
Found 1516 validated image filenames belonging to 2 classes.
48/48 [==============================] - 36s 745ms/step
Epoch 1/10
48/48 [==============================] - 59s 1s/step - loss: 0.3095 - accuracy: 0.8806
Epoch 2/10
48/48 [==============================] - 57s 1s/step - loss: 0.0969 - accuracy: 0.9584
Epoch 3/10
48/48 [==============================] - 59s 1s/step - loss: 0.0584 - accuracy: 0.9769
Epoch 4/10
48/48 [==============================] - 57s 1s/step - loss: 0.0327 - accuracy: 0.9927
Epoch 5/10
48/48 [==============================] - 57s 1s/step - loss: 0.0338 - accuracy: 0.9861
Epoch 6/10
48/48 [==============================] - 58s 1s/step - loss: 0.0205 - accuracy: 0.9947
Epoch 7/10
48/48 [==============================] - 57s 1s/step - loss: 0.0260 - accuracy: 0.9888
Epoch 8/10
48/48 [==============================] - 59s 1s/step - loss: 0.0138 - accuracy: 0.9974
Epoch 9/10
48/48 [==============================] - 58s 1s/step - loss: 0.0147 - accuracy: 0.9947
Epoch 10/10
48/48 [==============================] - 59s 1s/step - loss: 0.0080 - accuracy: 0.9987
```

Figure 4: Training model of Decision Tree + CNN Hybrid Neural Network

The Third model is a combination of Random Forest + CNN where We developed CNN architecture based on above CNN architecture and fitted that into random forest classifier where the N_estimators was given as 100 which defines the number of trees, that the subsets of various training dataset get trained,



```
Found 1516 validated image filenames belonging to 2 classes.
48/48 [==============================] - 37s 756ms/step
Epoch 1/10
48/48 [==============================] - 65s 1s/step - loss: 0.6190 - accuracy: 0.7856
Epoch 2/10
48/48 [==============================] - 63s 1s/step - loss: 0.1185 - accuracy: 0.9565
Epoch 3/10
48/48 [==============================] - 63s 1s/step - loss: 0.0670 - accuracy: 0.9730
Epoch 4/10
48/48 [==============================] - 64s 1s/step - loss: 0.0415 - accuracy: 0.9855
Epoch 5/10
48/48 [==============================] - 64s 1s/step - loss: 0.0313 - accuracy: 0.9881
Epoch 6/10
48/48 [==============================] - 65s 1s/step - loss: 0.0185 - accuracy: 0.9941
Epoch 7/10
48/48 [==============================] - 64s 1s/step - loss: 0.0162 - accuracy: 0.9960
Epoch 8/10
48/48 [==============================] - 64s 1s/step - loss: 0.0128 - accuracy: 0.9954
Epoch 9/10
48/48 [==============================] - 61s 1s/step - loss: 0.0075 - accuracy: 0.9987
Epoch 10/10
48/48 [==============================] - 63s 1s/step - loss: 0.0093 - accuracy: 0.9974
<keras.callbacks.History at 0x7efcfac6d070>
```

Figure 5: Training model of Random Forest + CNN Hybrid Neural Network.

Q2. How do hybrid neural networks determine the freshness of the meat based on the image? Was analyzed in the following manner.

As we distributed the images into train and test data, we now verified the accuracy of the predictions made by our model. We have taken 20% of the dataset as test_Dataset and regenerated it by rescaling them to feed it into the model. While the predictions were taking place, the difference between the predicted output and actual output is considered 'Loss', which minimizes upon increase in the model's accuracy.

Based on the below statistics, the question: Q3. How to improve accuracy and increase the performance speed of the proposed model? Was analyzed.

Table 1: Time taken by each model and its accuracy.

Hybrid Neural Network	Time Taken	Test Accuracy
SVM + CNN	63 seconds	100%
Decision Tree + CNN	59 seconds	100%
Random Forest + CNN	63 seconds	99. 21 %

Although all three models have the same level of accuracy, upon examining the time taken by each model, A technique called early stopping was used which stopped the training process when the validation loss stop kept decreasing. This prevented overfitting and improved the generalization of the model. It can be concluded that the Decision Tree +CNN model is the most efficient option for classifying meat into the categories of fresh and spoiled. The following chapter covers the discussion of the findings and conclusion, followed by suggestions for areas for further study.

CHAPTER FIVE

DISCUSSION, CONCLUSION, AND AREAS FOR FURTHER STUDY

DISCUSSION

The research questions are:

1. What hybrid neural networks should be used to predict freshness?
2. How do hybrid neural networks determine the freshness of the meat based on the image?
3. How to improve accuracy and increase the performance speed of the proposed model?

What follows is the discussion of the findings and conclusion, followed by suggestions for areas for further study

Findings

1) Various machine learning algorithms were utilized to predict the freshness or spoilage of meat based on image data. Specifically, Support Vector Machines, Decision Trees, and Random Forests were used as traditional machine learning algorithms, while Convolutional Neural Networks were used as a deep learning algorithm.

2) Image rescaling was done to ensure that the images were of consistent size and quality, which is essential for machine learning models to function effectively. After the images were properly rescaled, they were trained using the

tensorflow.keras.models, a high-level neural networks API of the TensorFlow library, which allowed the creation and training of complex machine learning models in a simple and straightforward manner.

The images were then passed through the neural networks to extract relevant features and patterns, which are crucial in the prediction process. Once the neural networks had been trained with the images, the machine learning models were subsequently trained, allowing them to make accurate predictions based on the extracted features.

The training of the machine learning models required the use of the previously obtained features to create an algorithm capable of predicting the freshness or spoilage of meat accurately.

3) In this study, it was found that the general models resulted in a 100% accuracy rate when utilized for the given task. However, it is important to consider other factors when selecting the optimal model. One such factor is the time taken by each model to classify. In many real-world scenarios, speed is a crucial factor, and a model that can produce accurate results in a shorter amount of time may be preferable to a more accurate model that takes longer to classify the data. Therefore, while the general models utilized in this study may have resulted in a 100% accuracy rate, the time taken by each model to classify must also be considered when selecting the optimal model for a given task. One way to enhance the speed of a machine learning model's performance is by utilizing a distributed computing environment for training, which involves the collaboration

of multiple machines to carry out computations. This technique can effectively accelerate the training process and enhance the model's scalability (M. Beraka et al., 2011).

.

Conclusion

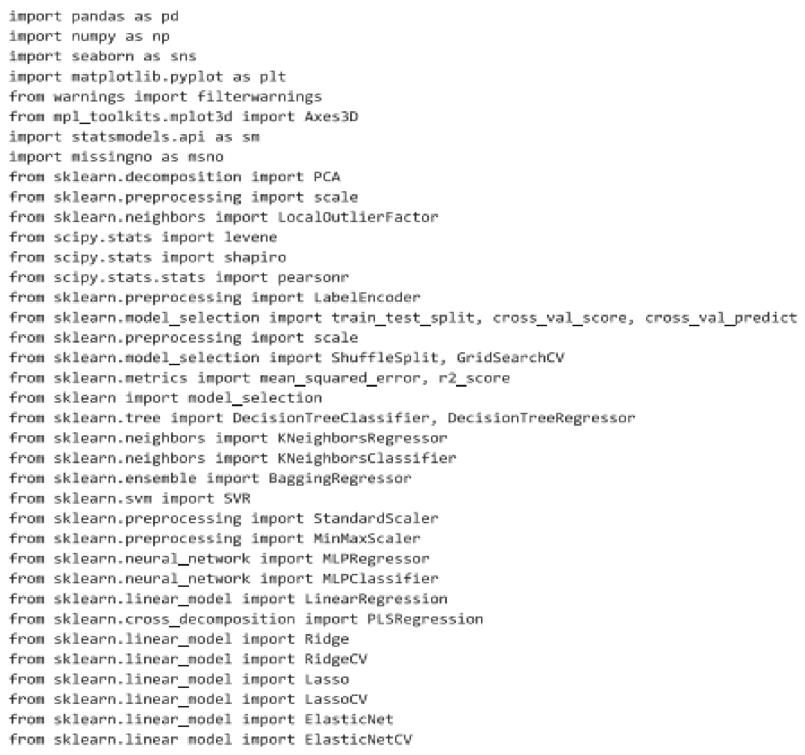
Utilizing the models developed, it is possible to classify meat products as either fresh or spoiled using image analysis. This approach not only reduces the reliance on human input for meat classification but also decreases the time taken to complete the classification process. By leveraging image analysis, the models can detect visual characteristics of fresh and spoiled meat, such as color and texture, to accurately classify the products. This streamlines the classification process and reduces the potential for human error arising from subjective judgments. In addition to enhancing efficiency and accuracy, automating the meat classification process using image analysis can reduce labor costs and improve productivity. This approach could be particularly beneficial for the meat industry, where high volumes of product need to be processed efficiently and accurately. Overall, the use of models for meat classification via image analysis represents a promising approach that can increase efficiency, accuracy, and productivity in the meat industry.

Areas of further study

As technology advances, it is possible that alongside visual analysis, there may be other sources of data such as genetics, feeding, and processing techniques that can impact meat quality. Research can be done to develop machine learning models that can integrate and fuse multi-modal data to make more accurate predictions of meat quality.

APPENDIX

CODES



```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from warnings import filterwarnings
from mpl_toolkits.mplot3d import Axes3D
import statsmodels.api as sm
import missingno as msno
from sklearn.decomposition import PCA
from sklearn.preprocessing import scale
from sklearn.neighbors import LocalOutlierFactor
from scipy.stats import levene
from scipy.stats import shapiro
from scipy.stats.stats import pearsonr
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split, cross_val_score, cross_val_predict
from sklearn.preprocessing import scale
from sklearn.model_selection import ShuffleSplit, GridSearchCV
from sklearn.metrics import mean_squared_error, r2_score
from sklearn import model_selection
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import BaggingRegressor
from sklearn.svm import SVR
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.neural_network import MLPRegressor
from sklearn.neural_network import MLPClassifier
from sklearn.linear_model import LinearRegression
from sklearn.cross_decomposition import PLSRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import RidgeCV
from sklearn.linear_model import Lasso
from sklearn.linear_model import LassoCV
from sklearn.linear_model import ElasticNet
from sklearn.linear model import ElasticNetCV
from sklearn import linear_model
from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier
from sklearn.ensemble import GradientBoostingRegressor, GradientBoostingClassifier
import xgboost as xgb
from xgboost import XGBRegressor, XGBClassifier
from lightgbm import LGBMRegressor, LGBMClassifier
from catboost import CatBoostRegressor, CatBoostClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn import tree
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report, roc_auc_score, roc_curve
from tensorflow.keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D, BatchNormalization,MaxPooling2D
from keras import models
from keras import layers
import tensorflow as tf
import os
import os.path
from pathlib import Path
import cv2
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras.utils.np_utils import to_categorical
```

```
<ipython-input-21-e46c423af031>:14: DeprecationWarning: Please use `pearsonr` from the `scipy.stats` namespace, th
  from scipy.stats.stats import pearsonr
```

```
Meat_Data_Data = Path("/content/Meat_data")
```

```
Meat_Data_Data
```

```
PosixPath('/content/Meat_data')
```

```
JPG_Path = list(Meat_Data_Data.glob(r"*/*.jpg"))
```

```
JPG_Path
```

```
       PosixPath('/content/Meat_data/Spolied/test_20171018_115121D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_145721D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_220721D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_092921D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171019_012321D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_113321D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171019_011721D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_200521D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171017_233121D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_091121D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_164721D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_115721D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_210321D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171019_024921D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_220921D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_015921D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171017_223521D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_101521D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_234321D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_141121D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_150521D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171017_220321D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171017_231721D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_060921D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171017_193321D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_180321D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_151321D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171017_220121D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171019_013321D.jpg'),
       PosixPath('/content/Meat_data/Spolied/test_20171018_044521D.jpg'),
       ...]
```

```
Labels = list(map(lambda x: os.path.split(os.path.split(x)[0])[1],JPG_Path))
```

```
print(Labels.count("Fresh"))
```

```
     948
```

```
print(Labels.count("Spolied"))
```

```
     948
```

```
File_Path = pd.Series(JPG_Path,name="JPG").astype(str)
```

```
Label_Name = pd.Series(Labels,name="CATEGORY")
```

```
print(Label_Name.value_counts())
```

```
     Fresh      948
     Spolied    948
     Name: CATEGORY, dtype: int64
```

```
Main_Data = pd.concat([File_Path,Label_Name],axis=1)
```

```
Main_Data = Main_Data.sample(frac=1).reset_index(drop=True)
```

```
print(Main_Data.head())
```

```
                                                 JPG CATEGORY
     0  /content/Meat_data/Spolied/test_20171018_14432...  Spolied
     1  /content/Meat_data/Spolied/test_20171018_18432...  Spolied
     2  /content/Meat_data/Spolied/test_20171018_18592...  Spolied
     3  /content/Meat_data/Fresh/test_20171017_030921D...    Fresh
     4  /content/Meat_data/Fresh/test_20171017_075721D...    Fresh
```

```
print(Main_Data["CATEGORY"].value_counts())
```

```
Spolied    948
Fresh      948
Name: CATEGORY, dtype: int64
```

```
Fresh_Meat = Main_Data[Main_Data["CATEGORY"] == "Fresh"]
Spoiled_Meat = Main_Data[Main_Data["CATEGORY"] == "Spolied"]
```

```
Main_Data
```

```
                                                   JPG  CATEGORY
   0   /content/Meat_data/Spolied/test_20171018_08212...   Spolied
   1  /content/Meat_data/Fresh/test_20171017_163521D...     Fresh
   2  /content/Meat_data/Fresh/test_20171017_091321D...     Fresh
   3  /content/Meat_data/Fresh/test_20171017_092121D...     Fresh
   4   /content/Meat_data/Spolied/test_20171017_20092...   Spolied
 ...                                                 ...       ...
1891  /content/Meat_data/Fresh/test_20171017_090921D...     Fresh
1892  /content/Meat_data/Fresh/test_20171016_182921D...     Fresh
1893  /content/Meat_data/Fresh/test_20171017_081121D...     Fresh
1894  /content/Meat_data/Fresh/test_20171017_122521D...     Fresh
1895  /content/Meat_data/Fresh/test_20171017_032321D...     Fresh
1896 rows × 2 columns
```

```
Main_Data['CATEGORY'] = Main_Data['JPG'].apply(lambda x: 0 if 'Fresh' in x else 1)
Main_Data.head()
```

```
                                                JPG  CATEGORY
0   /content/Meat_data/Spolied/test_20171018_08212...         1
1  /content/Meat_data/Fresh/test_20171017_163521D...          0
2  /content/Meat_data/Fresh/test_20171017_091321D...          0
3  /content/Meat_data/Fresh/test_20171017_092121D...          0
4   /content/Meat_data/Spolied/test_20171017_20092...         1
```

```
print(Fresh_Meat.head())
```

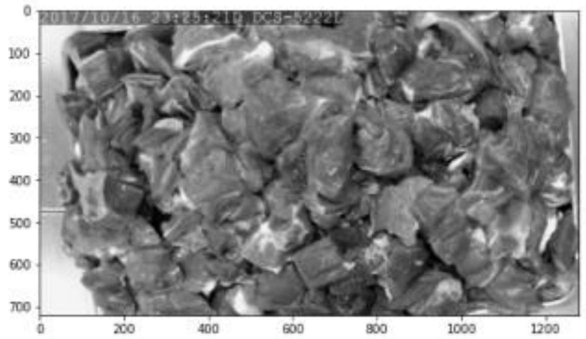
```
                                                JPG CATEGORY
0  /content/Meat_data/Fresh/test_20171017_173921D...    Fresh
1  /content/Meat_data/Fresh/test_20171017_071721D...    Fresh
3  /content/Meat_data/Fresh/test_20171017_011121D...    Fresh
6  /content/Meat_data/Fresh/test_20171017_111921D...    Fresh
7  /content/Meat_data/Fresh/test_20171017_020121D...    Fresh
```

```
print(Spoiled_Meat.head())
```

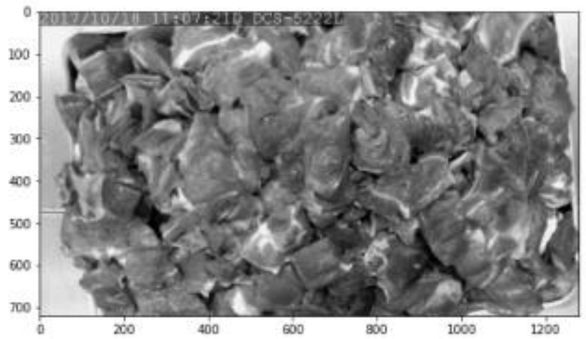
```
                                                JPG CATEGORY
2   /content/Meat_data/Spolied/test_20171018_11072...  Spolied
4   /content/Meat_data/Spolied/test_20171018_04332...  Spolied
```

```
5    /content/Meat_data/Spolied/test_20171018_06132...  Spolied
8    /content/Meat_data/Spolied/test_20171018_17332...  Spolied
13   /content/Meat_data/Spolied/test_20171018_18532...  Spolied
```

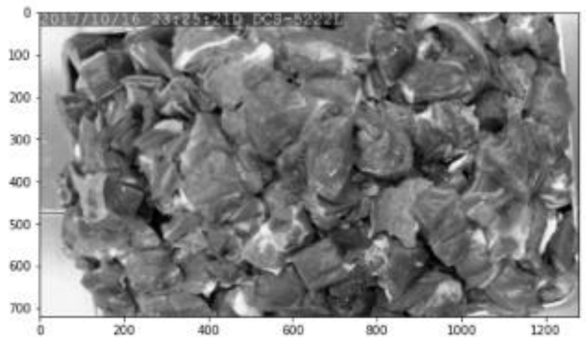
```
figure = plt.figure(figsize=(8,8))
plt.imshow(plt.imread(Main_Data["JPG"][10]))
plt.show()
```



```
figure = plt.figure(figsize=(8,8))
plt.imshow(plt.imread(Main_Data["JPG"][2]))
plt.show()
```



```
figure = plt.figure(figsize=(8,8))
plt.imshow(plt.imread(Fresh_Meat["JPG"][10]))
plt.show()
```



```
Fresh_Meat
```

	JPG	CATEGORY
0	/content/Meat_data/Fresh/test_20171017_173921D...	Fresh
1	/content/Meat_data/Fresh/test_20171017_071721D...	Fresh
3	/content/Meat_data/Fresh/test_20171017_011121D...	Fresh
6	/content/Meat_data/Fresh/test_20171017_111921D...	Fresh
7	/content/Meat_data/Fresh/test_20171017_020121D...	Fresh
...	...	...
1772	/content/Meat_data/Fresh/test_20171017_135521D...	Fresh
1773	/content/Meat_data/Fresh/test_20171016_200921D...	Fresh
1774	/content/Meat_data/Fresh/test_20171017_055721D...	Fresh
1775	/content/Meat_data/Fresh/test_20171017_024721D...	Fresh
1776	/content/Meat_data/Fresh/test_20171017_165721D...	Fresh

948 rows × 2 columns

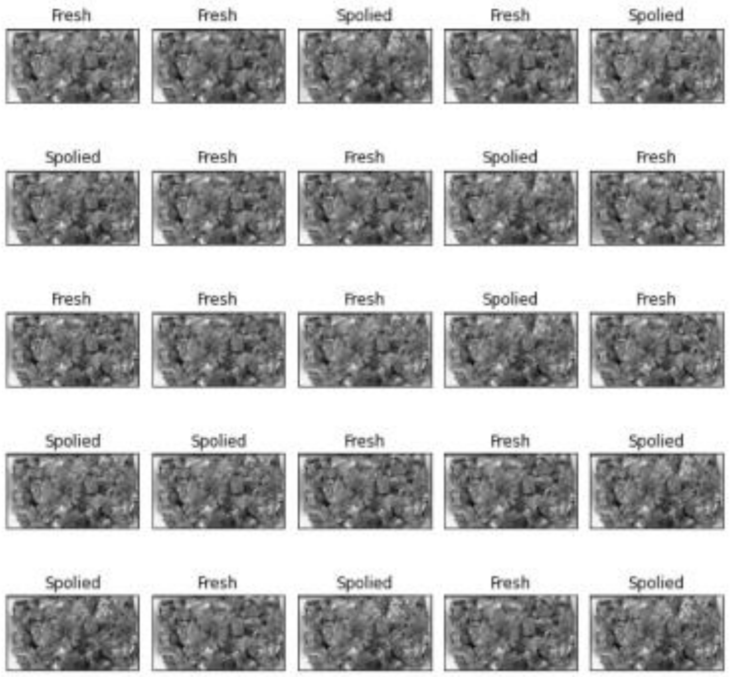
```
Spoiled_Meat
```

	JPG	CATEGORY
2	/content/Meat_data/Spolied/test_20171018_11072...	Spolied
4	/content/Meat_data/Spolied/test_20171018_04332...	Spolied
5	/content/Meat_data/Spolied/test_20171018_06132...	Spolied
8	/content/Meat_data/Spolied/test_20171018_17332...	Spolied
13	/content/Meat_data/Spolied/test_20171018_18532...	Spolied
...	...	...
1759	/content/Meat_data/Spolied/test_20171019_02192...	Spolied
1762	/content/Meat_data/Spolied/test_20171018_07252...	Spolied
1763	/content/Meat_data/Spolied/test_20171018_17232...	Spolied
1766	/content/Meat_data/Spolied/test_20171018_09472...	Spolied
1770	/content/Meat_data/Spolied/test_20171018_09012...	Spolied

829 rows × 2 columns

```
figure = plt.figure(figsize=(8,8))
plt.imshow(plt.imread(Spoiled_Meat["JPG"][2]))
plt.show()
```

```
fig, axes = plt.subplots(nrows=5, ncols=5, figsize=(8, 8),
                        subplot_kw={'xticks': [], 'yticks': []})

for i, ax in enumerate(axes.flat):
    ax.imshow(plt.imread(Main_Data["JPG"][i]))
    ax.set_title(Main_Data["CATEGORY"][i])
plt.tight_layout()
plt.show()
```



```
Train_Data, Test_Data = train_test_split(Main_Data, train_size=0.8, shuffle=True, random_state=42)
```

```
print(Train_Data.shape)
```

```
    (1421, 2)
```

```
print(Test_Data.shape)
```

```
    (356, 2)
```

```
Train_Data
```

	JPG	CATEGORY
438	/content/Meat_data/Fresh/test_20171017_064121D...	Fresh
198	/content/Meat_data/Spolied/test_20171019_01292...	Spolied
15	/content/Meat_data/Spolied/test_20171017_21472...	Spolied
265	/content/Meat_data/Spolied/test_20171018_18092...	Spolied
543	/content/Meat_data/Spolied/test_20171017_22392...	Spolied
...	...	...
1130	/content/Meat_data/Spolied/test_20171018_00252...	Spolied

```
train_df, valid_df = train_test_split(Main_Data, test_size = 0.2, random_state = 42)
print(train_df.shape, valid_df.shape)
```

```
(1516, 2) (380, 2)
```

```
train_df.head()
```

	JPG	CATEGORY
1387	/content/Meat_data/Spolied/test_20171018_09312...	1
544	/content/Meat_data/Spolied/test_20171018_09172...	1
937	/content/Meat_data/Fresh/test_20171017_183521D...	0
741	/content/Meat_data/Fresh/test_20171016_162721D...	0
1832	/content/Meat_data/Spolied/test_20171018_22472...	1

```
from sklearn import svm
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np

# Load the data
#train_df = ... # shape (1516, 2)
#image_dir = ... # path to directory containing image files

# Define the CNN architecture
input_shape = (128, 128, 3)
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=input_shape))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))

# Define the SVM classifier
svm_classifier = svm.SVC(kernel='linear')

# Define the hybrid model
svm_input = model.output
svm_output = Dense(1, activation='sigmoid')(svm_input)
hybrid_model = Model(inputs=model.input, outputs=svm_output)

# Compile the hybrid model
hybrid_model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Prepare the image data for training the CNN
train_datagen = ImageDataGenerator(rescale=1./255)
```

```
train_generator = train_datagen.flow_from_dataframe(
    dataframe=train_df,
    x_col="JPG",
    y_col="CATEGORY",
    target_size=(128, 128),
    batch_size=32,
    class_mode='binary')

# Train the CNN and extract features
cnn_features = model.predict(train_generator)

# Train the SVM on the extracted features
svm_classifier.fit(cnn_features, train_generator.labels)

# Train the hybrid model end-to-end
hybrid_model.fit(train_generator, epochs=10)
```

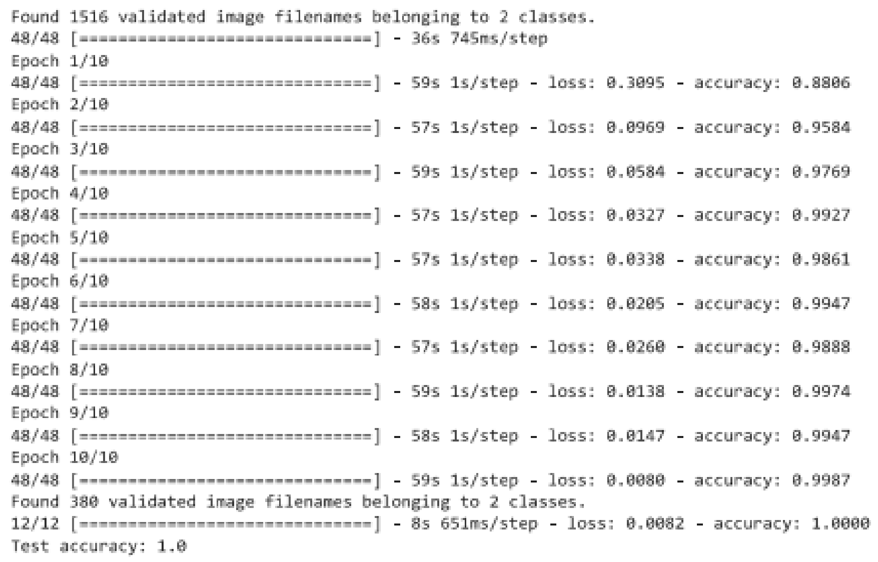
```
Found 1516 validated image filenames belonging to 2 classes.
48/48 [==============================] - 36s 748ms/step
Epoch 1/10
48/48 [==============================] - 68s 1s/step - loss: 0.3539 - accuracy: 0.8463
Epoch 2/10
48/48 [==============================] - 61s 1s/step - loss: 0.0593 - accuracy: 0.9769
Epoch 3/10
48/48 [==============================] - 62s 1s/step - loss: 0.0485 - accuracy: 0.9815
Epoch 4/10
48/48 [==============================] - 64s 1s/step - loss: 0.0248 - accuracy: 0.9921
Epoch 5/10
48/48 [==============================] - 62s 1s/step - loss: 0.0179 - accuracy: 0.9941
Epoch 6/10
48/48 [==============================] - 63s 1s/step - loss: 0.0142 - accuracy: 0.9941
Epoch 7/10
48/48 [==============================] - 64s 1s/step - loss: 0.0054 - accuracy: 0.9993
Epoch 8/10
48/48 [==============================] - 63s 1s/step - loss: 0.0026 - accuracy: 1.0000
Epoch 9/10
48/48 [==============================] - 64s 1s/step - loss: 0.0018 - accuracy: 1.0000
Epoch 10/10
48/48 [==============================] - 63s 1s/step - loss: 0.0011 - accuracy: 1.0000
<keras.callbacks.History at 0x7efcf8778be0>
```

```
test_datagen = ImageDataGenerator(rescale=1./255)
test_generator = test_datagen.flow_from_dataframe(
    dataframe=valid_df,
    x_col="JPG",
    y_col="CATEGORY",
    target_size=(128, 128),
    batch_size=32,
    class_mode='binary')

# Generate CNN features for the test data
cnn_test_features = model.predict(test_generator)

# Evaluate the SVM classifier on the CNN features
svm_test_preds = svm_classifier.predict(cnn_test_features)

# Evaluate the hybrid model on the test data
loss, accuracy = hybrid_model.evaluate(test_generator)
print("Test accuracy:", accuracy)
```

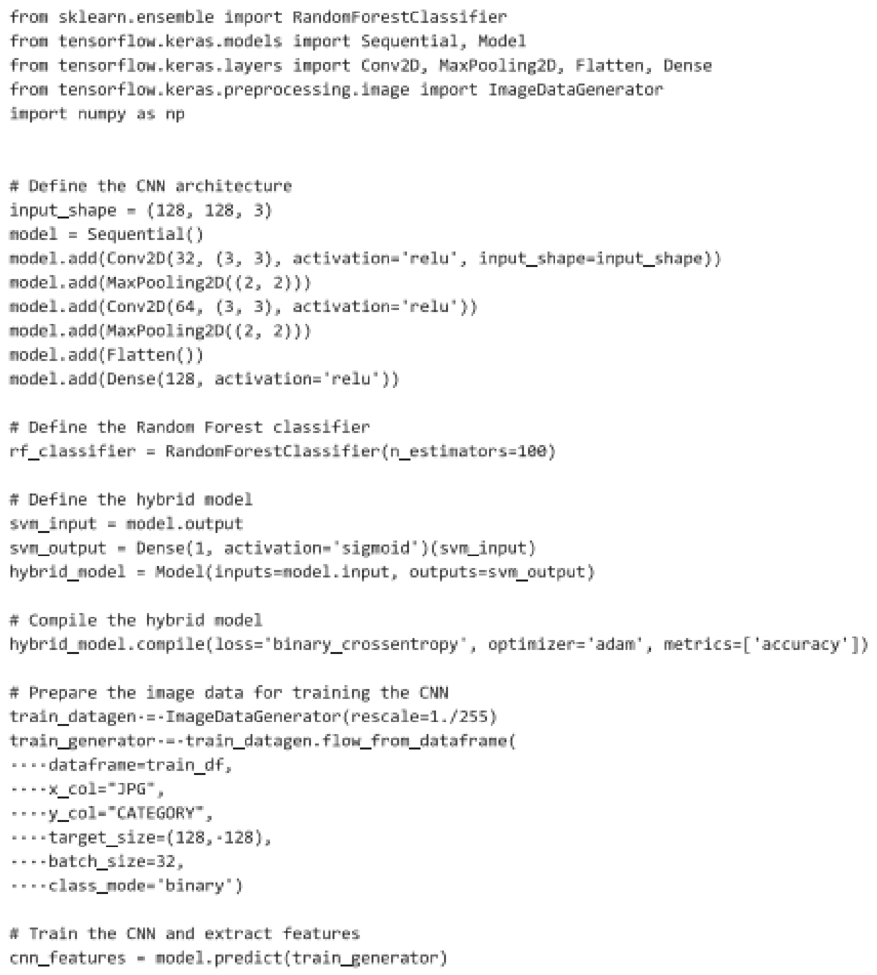
```
Found 380 validated image filenames belonging to 2 classes.
12/12 [==============================] - 9s 772ms/step
```

```
     12/12 [==============================] - 8s 640ms/step - loss: 0.0014 - accuracy: 1.0000
     Test accuracy: 1.0
```

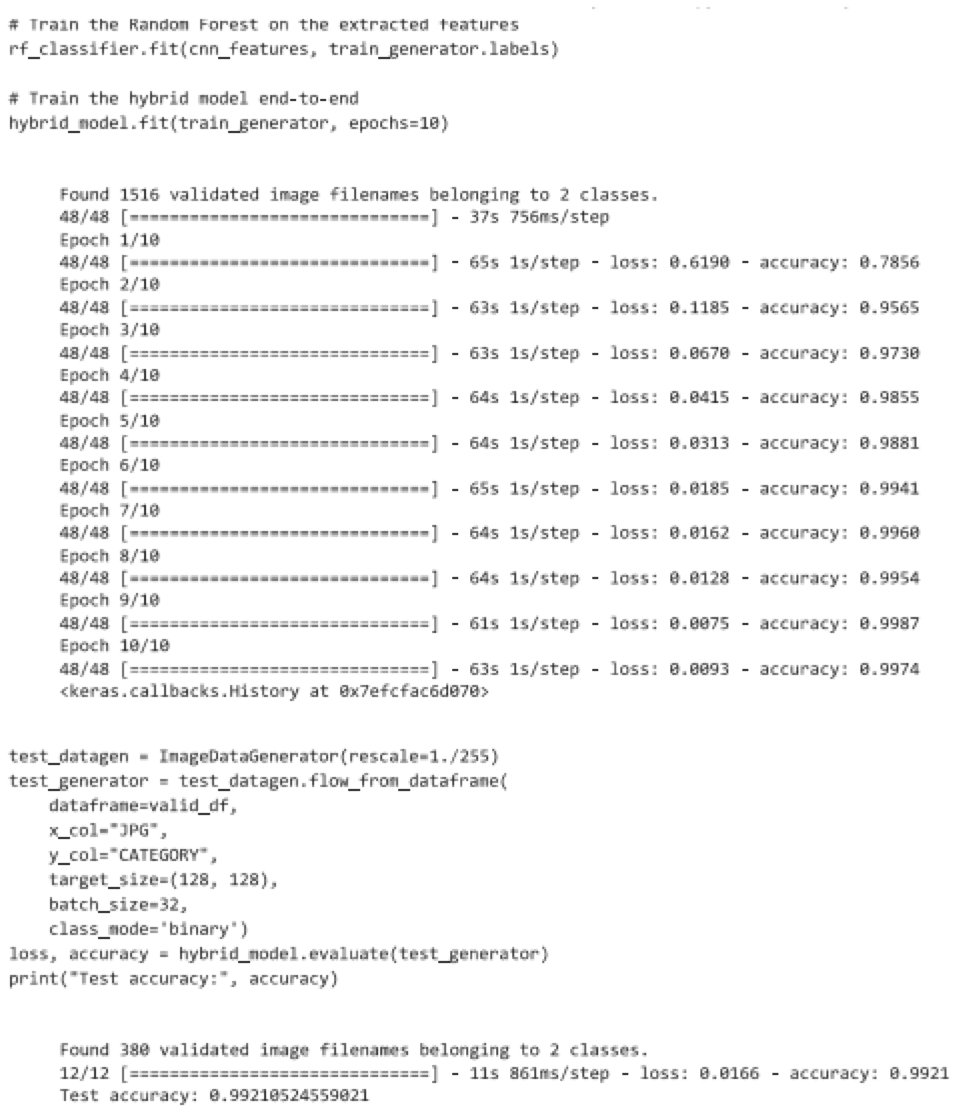
```
from sklearn.tree import DecisionTreeClassifier
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np

# Define the CNN architecture
input_shape = (128, 128, 3)
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=input_shape))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))

# Define the decision tree classifier
dt_classifier = DecisionTreeClassifier()

# Define the hybrid model
cnn_input = model.input
cnn_output = model.layers[-2].output
dt_input = cnn_output
dt_output = Dense(1, activation='sigmoid')(dt_input)
hybrid_model = Model(inputs=cnn_input, outputs=dt_output)

# Compile the hybrid model
hybrid_model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Prepare the image data for training the CNN
train_datagen = ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_dataframe(
    dataframe=train_df,
    x_col="JPG",
    y_col="CATEGORY",
    target_size=(128, 128),
    batch_size=32,
    class_mode='binary')

# Train the CNN and extract features
cnn_features = model.predict(train_generator)

# Train the decision tree on the extracted features
dt_classifier.fit(cnn_features, train_generator.labels)

# Train the hybrid model end-to-end
hybrid_model.fit(train_generator, epochs=10)

# Evaluate the hybrid model on the test set
test_datagen·=·ImageDataGenerator(rescale=1./255)
test_generator·=·test_datagen.flow_from_dataframe(
····dataframe=valid_df,
····x_col="JPG",
····y_col="CATEGORY",
····target_size=(128,·128),
····batch_size=32,
····class_mode='binary')
loss,·accuracy·=·hybrid_model.evaluate(test_generator)
print("Test·accuracy:",·accuracy)
```



```
Found 1516 validated image filenames belonging to 2 classes.
48/48 [==============================] - 36s 745ms/step
Epoch 1/10
48/48 [==============================] - 59s 1s/step - loss: 0.3095 - accuracy: 0.8806
Epoch 2/10
48/48 [==============================] - 57s 1s/step - loss: 0.0969 - accuracy: 0.9584
Epoch 3/10
48/48 [==============================] - 59s 1s/step - loss: 0.0584 - accuracy: 0.9769
Epoch 4/10
48/48 [==============================] - 57s 1s/step - loss: 0.0327 - accuracy: 0.9927
Epoch 5/10
48/48 [==============================] - 57s 1s/step - loss: 0.0338 - accuracy: 0.9861
Epoch 6/10
48/48 [==============================] - 58s 1s/step - loss: 0.0205 - accuracy: 0.9947
Epoch 7/10
48/48 [==============================] - 57s 1s/step - loss: 0.0260 - accuracy: 0.9888
Epoch 8/10
48/48 [==============================] - 59s 1s/step - loss: 0.0138 - accuracy: 0.9974
Epoch 9/10
48/48 [==============================] - 58s 1s/step - loss: 0.0147 - accuracy: 0.9947
Epoch 10/10
48/48 [==============================] - 59s 1s/step - loss: 0.0080 - accuracy: 0.9987
Found 380 validated image filenames belonging to 2 classes.
12/12 [==============================] - 8s 651ms/step - loss: 0.0082 - accuracy: 1.0000
Test accuracy: 1.0
```



```
from sklearn.ensemble import RandomForestClassifier
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np


# Define the CNN architecture
input_shape = (128, 128, 3)
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=input_shape))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))

# Define the Random Forest classifier
rf_classifier = RandomForestClassifier(n_estimators=100)

# Define the hybrid model
svm_input = model.output
svm_output = Dense(1, activation='sigmoid')(svm_input)
hybrid_model = Model(inputs=model.input, outputs=svm_output)

# Compile the hybrid model
hybrid_model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Prepare the image data for training the CNN
train_datagen·=·ImageDataGenerator(rescale=1./255)
train_generator·=·train_datagen.flow_from_dataframe(
····dataframe=train_df,
····x_col="JPG",
····y_col="CATEGORY",
····target_size=(128,·128),
····batch_size=32,
····class_mode='binary')

# Train the CNN and extract features
cnn_features = model.predict(train_generator)
```



```
# Train the Random Forest on the extracted features
rf_classifier.fit(cnn_features, train_generator.labels)

# Train the hybrid model end-to-end
hybrid_model.fit(train_generator, epochs=10)
```

```
Found 1516 validated image filenames belonging to 2 classes.
48/48 [==============================] - 37s 756ms/step
Epoch 1/10
48/48 [==============================] - 65s 1s/step - loss: 0.6190 - accuracy: 0.7856
Epoch 2/10
48/48 [==============================] - 63s 1s/step - loss: 0.1185 - accuracy: 0.9565
Epoch 3/10
48/48 [==============================] - 63s 1s/step - loss: 0.0670 - accuracy: 0.9730
Epoch 4/10
48/48 [==============================] - 64s 1s/step - loss: 0.0415 - accuracy: 0.9855
Epoch 5/10
48/48 [==============================] - 64s 1s/step - loss: 0.0313 - accuracy: 0.9881
Epoch 6/10
48/48 [==============================] - 65s 1s/step - loss: 0.0185 - accuracy: 0.9941
Epoch 7/10
48/48 [==============================] - 64s 1s/step - loss: 0.0162 - accuracy: 0.9960
Epoch 8/10
48/48 [==============================] - 64s 1s/step - loss: 0.0128 - accuracy: 0.9954
Epoch 9/10
48/48 [==============================] - 61s 1s/step - loss: 0.0075 - accuracy: 0.9987
Epoch 10/10
48/48 [==============================] - 63s 1s/step - loss: 0.0093 - accuracy: 0.9974
<keras.callbacks.History at 0x7efcfac6d070>
```

```
test_datagen = ImageDataGenerator(rescale=1./255)
test_generator = test_datagen.flow_from_dataframe(
    dataframe=valid_df,
    x_col="JPG",
    y_col="CATEGORY",
    target_size=(128, 128),
    batch_size=32,
    class_mode='binary')
loss, accuracy = hybrid_model.evaluate(test_generator)
print("Test accuracy:", accuracy)
```

```
Found 380 validated image filenames belonging to 2 classes.
12/12 [==============================] - 11s 861ms/step - loss: 0.0166 - accuracy: 0.9921
Test accuracy: 0.99210524559021
```

REFERENCES

Calvin, & Putra, Ghiri & Prakasa, Esa. (2020). Classification of Chicken Meat Freshness using Convolutional Neural Network Algorithms. 16.10.1109/3ICT51146.2020.9312018.https://www.researchgate.net/publication/348366725_Classification_of_Chicken_Meat_Freshness_using_Convolutional_Neural_Network_Algorithms

M. Guermazi, O. Kanoun and N. Derbel, Method of model's parameters classification using neural network for meat characterization, 10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13), Hammamet, Tunisia, 2013, pp. 1-5, doi: 10.1109/SSD.2013.6564163. https://www.researchgate.net/publication/261379650_Method_of_model's_parameters_classification_using_neural_network_for_meat_characterization

V. S. Kartika, M. Rivai and D. Purwanto, "Spoiled meat classification using semiconductor gas sensors, image processing and neural network," 2018 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2018, pp. 418-423, doi:

10.1109/ICOIACT.2018.8350678.
https://scholar.its.ac.id/en/publications/spoiled-meat-classification-using-semiconductor-gas-sensors-image

Medeiros, E.C.; Almeida, L.M.; Filho, J.G.d.A.T. Computer Vision and Machine Learning for Tuna and Salmon Meat Classification. Informatics 2021, 8, 70. https://doi.org/10.3390/informatics8040070

B. W. Penning, W. M. Snelling, and M. J. Woodward-Greene, Machine Learning in the Assessment of Meat Quality, in IT Professional, vol. 22, no. 3, pp. 39-41, 1 May-June 2020, doi: 10.1109/MITP.2020.2986123. https://www.ars.usda.gov/research/publications/publication/?seqNo115=371112

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The Caltech-UCSD Birds-200-2011 Dataset,” California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011. https://authors.library.caltech.edu/27452

D. Singh, N. D. Shah, and J. J. Wang, “Content-based image retrieval using deep learning and decision tree,” in 2015 IEEE International Conference on Big Data (Big Data), 2015, pp. 786–793 https://www.researchgate.net/publication/322990964_Content_based_image_retrieval_using_deep_learning_process

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014. https://jmlr.org/papers/v15/srivastava14a.html

Lorenzo JM, Munekata PE, Dominguez R, Pateiro M, Saraiva JA, Franco D. Main Groups of Microorganisms of Relevance for Food Safety and Stability: General Aspects and Overall Description. Innovative Technologies for Food Preservation. 2018:53–107. doi: 10.1016/B978-0-12-811031-7.00003-0. Epub 2017 Sep 29. PMCID: PMC7150063. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7150063/

Amin Taheri-Garavand, Soodabeh Fatahi, Mahmoud Omid, Yoshio Makino,Meat quality evaluation based on computer vision technique: A review,Meat Science, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7150063/

O.Ulucan , D.Karakaya and M.Turkan. (2019) Meat quality assessment based on deep learning. In Conf. Innovations Intell. Syst. Appli. (ASYU) https://www.kaggle.com/datasets/crowww/meat-quality-assessment-based-on-deep-learning

Zhao, R., Yang, Z., Zheng, H. et al. A framework for the general design and computation of hybrid neural networks. Nat Commun 13, 3427 (2022). https://doi.org/10.1038/s41467-022-30964-7

LeCun, Yann & Bengio, Y. & Hinton, Geoffrey. (2015). Deep Learning. Nature. 521. 436-44. 10.1038/nature14539. https://pubmed.ncbi.nlm.nih.gov/26017442/

M. Beraka, H**.** Mathkour and S. Gannouni, "Data sharing in distributed computing environment," 2011 International Conference on Electrical and Control Engineering, Yichang, China, 2011, pp. 4760-4763, doi: 10.1109/ICECENG.2011.6057414.https://www.researchgate.net/publication/252054422_Data_sharing_in_distributed_computing_environment

Duggani Keerthana, Vipin Venugopal, Malaya Kumar Nath, Madhusudhan Mishra, Hybrid convolutional neural networks with SVM classifier for classification of skin cancer, https://doi.org/10.1016/j.bea.2022.100069.

Xu S, Liu S, Wang H, Chen W, Zhang F, Xiao Z. A Hyperspectral Image Classification Approach Based on Feature Fusion and Multi-Layered Gradient Boosting Decision Trees. *Entropy (Basel)*. 2020;23(1):20. Published 2020 Dec 25. doi:10.3390/e23010020 https://www.mdpi.com/1099-4300/23/1/20

Raschka S, Patterson J, Nolet C. Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. *Information*. 2020; 11(4):193. https://doi.org/10.3390/info11040193

Badillo S, Banfai B, Birzele F, et al. An Introduction to Machine Learning. *Clin Pharmacol Ther*. 2020;107(4):871-885. doi:10.1002/cpt.1796 https://pubmed.ncbi.nlm.nih.gov/32128792/

Szarvas, M., Yoshizawa, A., Yamamoto, M., & Ogata, J. (2005). Pedestrian detection with convolutional neural networks. Proceedings of the IEEE on Intelligent Vehicles Symposium, pp. 224–229. https://www.semanticscholar.org/paper/Pedestrian-detection-with-convolutional-neural-Szarvas-Yoshizawa/47caac4cf1169cd46f38c8ecb30a9906be0efc88

Haotian Shi, An Architecture Combining Convolutional Neural Network and Support Vector Machine for Image Recognition (2018). Cortes, C., Vapnik, V. Support-Vector Networks. *Machine Learning* **20**, 273–297 (1995). https://doi.org/10.1023/A:1022627411411

Song YY, Lu Y. Decision tree methods: applications for classification and prediction. *Shanghai Arch Psychiatry. 2015;27(2):130-135. doi: 10.11919/j.issn*.1002-0829.215044 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4466856/

Kircali Ata S, Shi JK, Yao X, et al. Predicting the Textural Properties of Plant-Based Meat Analogs with Machine Learning. *Foods*. 2023;12(2):344. Published 2023 Jan 11. doi:10.3390/foods12020344 https://www.mdpi.com/2304-8158/12/2/344

Cortes, C., Vapnik, V. Support-Vector Networks. Machine learning 20, 273–297 (1995). https://doi.org/10.1023/A:1022627411411

Abien Fred M. Agarap. An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification.

https://doi.org/10.48550/arXiv.1712.03541

Li, T., Leng, J., Kong, L. *et al.* DCNR: deep cube CNN with random forest for hyperspectral image classification. *Multimed Tools Appl* 78, 3411–3433 (2019). https://doi.org/10.1007/s11042-018-5986-5 https://doi.org/10.3390/informatics8040070https://doi.org/10.48550/arXiv.1712.03541

Hao, J., & Ho, T. K. (2019). Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language. Journal of Educational and Behavioral Statistics, 44(3), 348–361. https://doi.org/10.3102/1076998619832248