University of Bahrain

College of Information Technology

Department of Computer Science



**Machine Learning Project**

Shahd Talal 202005797

**Dataset**

The dataset we’re working with is a unique compilation of three distinct collections of pictures, each representing a specific type of food: sushi, steak, and pizza. Each collection is rich in its diversity, capturing the various presentations and styles of these dishes from different cuisines around the world.

The process of splitting the dataset into training and testing subsets presented a significant challenge. The goal was to allocate 70% of the images for training the model, and the remaining 30% for testing its performance. This split is crucial to ensure that the model is well-trained and to evaluate its ability to generalize unseen data.

However, achieving this split was not straightforward due to the inherent complexities involved. These complexities could be attributed to factors such as the varying quality of images, the diversity in presentation styles of the same food item, and the need to maintain a balanced representation of each food type in both the training and testing subsets.

In addition to the training and testing sets, we reserved 15 pictures from each collection for prediction. These images serve as a separate dataset that we use to make predictions and observe how our model performs on data it has never seen before. This step is crucial in understanding the practical applicability of our model in real-world scenarios.

As we continue to refine our model and its training process, we are learning more about the nuances of working with image datasets. Despite the challenges, the potential benefits of successfully training our model - from aiding in food recognition apps to contributing to nutritional analysis tools - make our efforts worthwhile.

**Preprocessing**

In this phase of the project, two key techniques were employed to reprocess the images: Normalization and Image Encoding.

Normalization is a preprocessing technique used in machine learning to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the ranges of values or losing information. When dealing with images, normalization often involves adjusting the pixel values so that they fall within a specific range. This is typically between 0 and 1, or -1 and 1. The process involves subtracting the minimum pixel value and dividing it by the range of pixel values. Normalizing images can help the performance of your model, as calculations are often faster on smaller numbers, and it can help the model learn the weights of different features more effectively.

On the other hand, Image Encoding is the process of converting an image from one format to another while ensuring the image retains its visual information. In the context of machine learning, image encoding often involves converting images into a format that the machine learning model can understand. This could mean converting colour images to grayscale, resizing images to a certain size, or even flattening a 2D image into a 1D array for models such as a simple neural network. Additionally, if the images are being used for a classification task, the labels might also need to be encoded. This could involve one-hot encoding categorical labels into a binary vector.

These techniques are crucial in the field of computer vision, and their correct application can often be the difference between a model that performs well and one that doesn’t. By using these techniques, the project is set up to handle the images in a way that is optimal for the performance of the subsequent machine-learning model. As the project progresses, these techniques will likely be refined and adjusted to better suit the needs of the model and the data.

**Machine Learning Models:**

In this part of the project, we started with algorithms that performed poorly with pictures even though they are good with different data such as K nearest neighbour and K mean clustering that is why we switched to different algorithms however, all these algorithms performed almost the same. This is why I attempted to use (PCA) for feature selection with the assumption that the performance would be improved yet, it was not the case as the accuracy did not improve.

1. **Random Forest**

The first model used is the Random Forest classifier which performed with 60% accuracy which is not bad but also not high performance.

Accuracy: 61.69%

Precision: 63.73%

Recall: 61.69%

F1 Score: 61.61%

1. **Artificial Neural Networks and Conventional Neural Networks**

The Artificial Neural Network had a very low performance.

Accuracy: 52.37%

Average Precision: 54.1%

Average Recall: 52.37%

Average F1 Score: 51.67%

As my research showed that Artificial Neural Networks are the best for classification, I dug deeper into why the previous Neural Network model was not performing well so, I ended up with this modified model of ANN using Pytorch to work well with a dataset made of pictures however, the performance did not dramatically evaluate.

Test Accuracy of the model on the 162 test images: 62.96296296296296%

Classification Report:

precision recall f1-score support

pizza 0.61 0.74 0.67 54

steak 0.74 0.80 0.77 54

sushi 0.50 0.35 0.41 54

accuracy 0.63 162

macro avg 0.62 0.63 0.62 162

weighted avg 0.62 0.63 0.62 162

Confusion Matrix:

[[40 1 13]

[ 5 43 6]

[21 14 19]]

**Conclusion**

In conclusion both models did not have many differences in terms of performance however, the Random Forest was simpler and easier to implement compared to the NN and did not require me any modification to improve the performance.