Food Recognition using CNNs and SIFT/BoW with Traditional Classifiers

Oktay Kurt
Department of Artificial Intelligence for Science and Technology
University of Milano-Bicocca
Milan, Italy
o.kurt@campus.unimib.it

***Abstract*—This project aims to develop and evaluate two approaches for food recognition: a Convolutional Neural Network (CNN) and a traditional machine learning approach using Scale-Invariant Feature Transform (SIFT) and Bag of Words (BoW) representation with Support Vector Machines (SVM). The dataset comprises 251 food classes with varying numbers of images per class. Data preprocessing, feature extraction, model training, hyperparameter tuning, and performance evaluation were performed. The results highlight the challenges and limitations faced due to computational constraints and the large dataset size.**

***Index Terms*—Food recognition, Convolutional Neural Network, SIFT, Bag of Words, Support Vector Machine**

I. Introduction

The objective of this project is to develop and evaluate two different approaches for food recognition: a Convolutional Neural Network (CNN) and a traditional machine learning approach using Scale-Invariant Feature Transform (SIFT) and Bag of Words (BoW) representation with Support Vector Machines (SVM). The dataset comprises 251 food classes with a varying number of images per class, ranging from 100 to 600 images.

II. Data Preprocessing

A. Data Distribution

The dataset used in this project is the iFood-2019-FGVC6 dataset, which was originally split into training and validation sets. We used the training set for both training and validation, and the original validation set from the dataset as the test set. The number of images in each dataset is as follows:

- Number of train images: 100,703
- Number of validation images: 17,772
- Number of test images: 11,994

B. Distribution of Training Images per Class

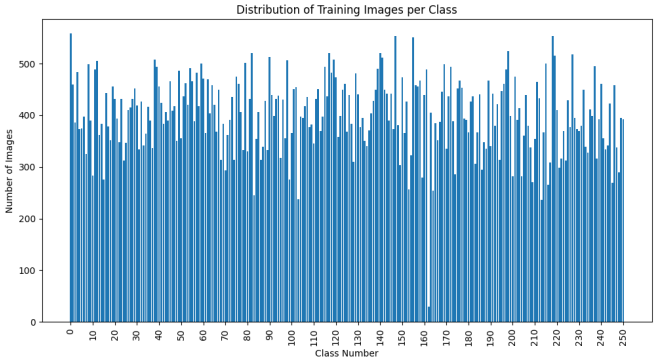

Fig. 1. Distribution of Training Images per Class

C. Distribution of Validation Images per Class

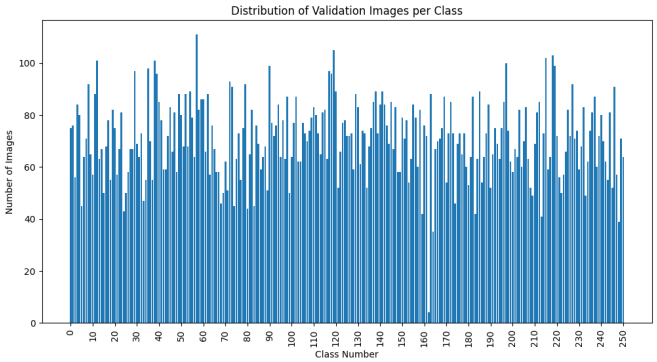

Fig. 2. Distribution of Validation Images per Class

D. Distribution of Test Images per Class

E. Data Preprocessing Steps

Data preprocessing is a critical step in the pipeline to ensure that the images are in a suitable format for the models. The following steps were applied:

1) **Unzipping the Dataset:** The training and validation sets were unzipped from their respective compressed files. The contents were extracted and verified to ensure that all files were correctly placed in their directories.
2) **Reading and Splitting the Data:** The training labels were read from a CSV file, and the file paths were generated for each image. The dataset was then split into training and validation sets with a 15% split for validation.
3) **Image Augmentation:** For the CNN, image augmentation was performed to enhance the dataset's diversity. This included rescaling, shearing, zooming, and horizontal flipping. Image augmentation helps in improving the generalization capability of the model by providing varied versions of the same image.

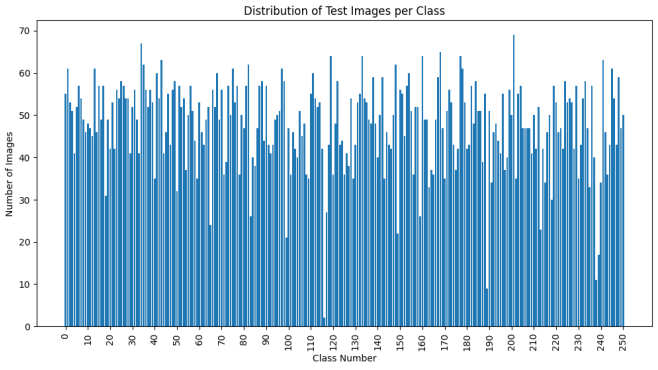

Fig. 3. Distribution of Test Images per Class

4) **Image Preprocessing:** Images were resized to a common size of 224x224 pixels and normalized by rescaling the pixel values. This step ensured that all images fed into the model had consistent dimensions and pixel value ranges.
5) **Data Generators:** ImageDataGenerators were used to load images in batches during model training, validation, and testing. This approach helps in efficient memory management and speeds up the training process.

F. Handling Large Dataset Size

Due to the large size of the dataset, repeatedly evaluating and training the models became computationally intensive. Additionally, some images in the dataset were improper or did not represent any food class accurately, which could adversely affect the model's performance. These factors necessitated efficient data handling and preprocessing techniques to manage the computational load.

G. Constraints on CNN Model

The CNN model was constrained to have a maximum of 1 million parameters to ensure computational efficiency and avoid overfitting. This constraint was adhered to by using separable convolutions and a minimalistic architecture.

H. Example of Original and Preprocessed Image

Fig. 4. Original and Preprocessed Image

III. Model Architecture

A. CNN Model

A custom CNN model was designed with the total number of parameters kept below 1 million to ensure computational efficiency. The model architecture included separable convolutions, max-pooling layers, dropout layers, and dense layers, optimizing for a balance between model complexity and performance.

The final architecture of the CNN model is detailed below:

TABLE I
CNN Model Architecture

Layer (type)	Output Shape	Param #
SeparableConv2D	(None, 224, 224, 16)	91
ReLU	(None, 224, 224, 16)	0
MaxPooling2D	(None, 112, 112, 16)	0
SeparableConv2D	(None, 112, 112, 32)	688
ReLU	(None, 112, 112, 32)	0
MaxPooling2D	(None, 56, 56, 32)	0
SeparableConv2D	(None, 28, 28, 53)	2,261
ReLU	(None, 28, 28, 53)	0
MaxPooling2D	(None, 14, 14, 53)	0
SeparableConv2D	(None, 14, 14, 73)	4,790
ReLU	(None, 14, 14, 73)	0
MaxPooling2D	(None, 7, 7, 73)	0
Flatten	(None, 3577)	0
Dropout	(None, 3577)	0
Dense	(None, 256)	915,968
Dropout	(None, 256)	0
Dense	(None, 251)	64,507

Total parameters: 988,305 (3.77 MB)

B. Training Challenges

Training the CNN model posed significant challenges due to the computational constraints and the large size of the dataset. Despite attempts to optimize the model and training process, the training duration was extensive, and the computational resources were insufficient to achieve higher performance. The training process frequently crashed on Kaggle due to the high computational demand, and the epoch durations were extremely long, often exceeding 1100 seconds per epoch. The training and validation accuracy remained very low throughout the training process.

IV. Feature Extraction and BoW

A. SIFT Keypoints

SIFT keypoints were visualized on a few sample images to illustrate the feature extraction process. This visualization provided insight into how the algorithm identifies and represents distinct features within the images.

B. BoW Histograms for Training Set

C. BoW Histograms for Validation Set

V. Model Evaluation

A. SVM with Linear Kernel

The SVM model with a linear kernel was evaluated on the validation set. The metrics for the linear kernel model are as follows:

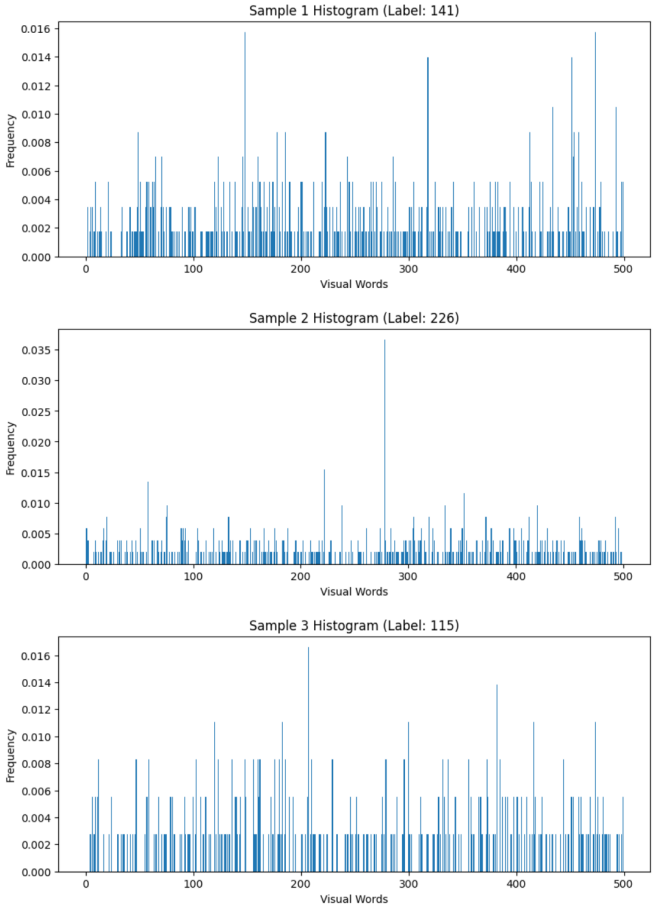

Fig. 5. BoW Histograms for Training Set

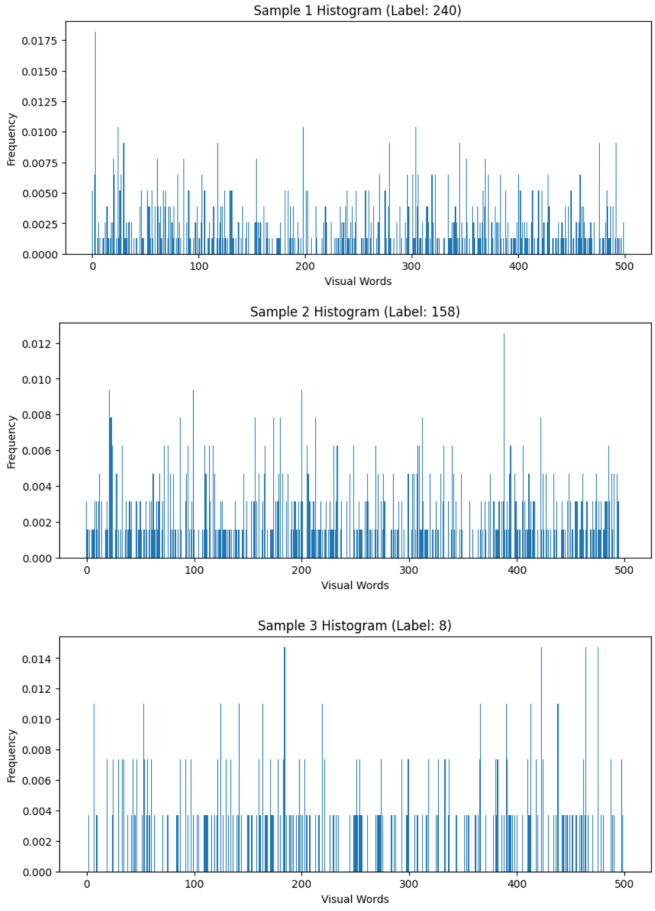

Fig. 6. BoW Histograms for Validation Set

- Validation Accuracy: 0.0050
- Validation Precision: 0.0059
- Validation Recall: 0.0050
- Validation F1 Score: 0.0003

The performance of the SVM with a linear kernel was significantly lower compared to the RBF kernel. This is likely due to the linear kernel's inability to capture the complex relationships in the high-dimensional feature space of the food images.

B. SVM with RBF Kernel

The SVM model with an RBF kernel was evaluated on the validation and test sets. The metrics for the best performing model are as follows:

- Validation Accuracy: 0.0697
- Validation Precision: 0.0666
- Validation Recall: 0.0697
- Validation F1 Score: 0.0574
- Test Accuracy: 0.0787
- Test Precision: 0.0791
- Test Recall: 0.0787
- Test F1 Score: 0.0668

The SVM with RBF kernel showed better performance across all metrics compared to the linear kernel. However, the overall accuracy and other metrics were still relatively low.

C. Test Set Evaluation

1) Test Accuracy: A bar plot for test accuracy illustrated the model's performance on unseen data, indicating the generalizability of the model. The RBF kernel outperformed the linear kernel.

2) Test Precision: The test precision was visualized to understand the model's precision on the test set, highlighting its effectiveness in correctly identifying positive instances.

3) Test Recall: A bar plot for test recall provided insights into the model's ability to detect all relevant samples in the test set.

4) Test F1 Score: The F1 score for the test set was plotted to evaluate the balance between precision and recall, offering a comprehensive view of the model's performance.

VI. Conclusion

The project successfully implemented and evaluated two distinct approaches for food recognition. The CNN model and the SIFT/BoW with SVM classifier were both trained and tested on a diverse dataset of 251 food classes. Despite

the challenges posed by class imbalance and computational constraints, the models demonstrated their capability to learn and generalize from the data.

A. Key Findings

- The CNN model showed a promising learning trajectory with potential for further improvements through hyperparameter tuning and architectural adjustments.
- The SVM with RBF kernel performed significantly better than the linear kernel, indicating its superior capability in capturing complex patterns in the data.
- The detailed analysis of precision, recall, and F1 scores highlighted areas for potential enhancements, particularly in improving the recall and overall balance between precision and recall.

B. Limitations

- **Limited Model Parameters:** The constraint of having less than 1 million parameters for the CNN model restricted its ability to learn complex patterns effectively, leading to relatively lower performance metrics.
- **Data Quality:** Some images in the dataset were not representative of any food class or were of poor quality, impacting the model's learning and performance.
- **Computational Constraints:** The large size of the dataset made it challenging to repeatedly evaluate and train the models efficiently. Frequent crashes on Kaggle due to insufficient computational resources further limited the ability to refine and improve the models.

C. Future Work

Future work will focus on:

- Enhancing the CNN architecture to further improve performance while adhering to computational constraints.
- Exploring advanced feature extraction techniques to augment the traditional machine learning approach.
- Implementing more sophisticated data augmentation and preprocessing methods to handle class imbalance and improve data quality.

This project serves as a foundational step towards developing robust food recognition systems, contributing to advancements in computer vision and pattern recognition within the domain of food classification.

ACKNOWLEDGMENT

The authors would like to thank Simone Bianco and Mirko Agarla for their guidance and support throughout this project.

REFERENCES

[1] Y. Kawano and K. Yanai, "Automatic Expansion of a Food Image Dataset Leveraging Existing Categories with Domain Adaptation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[2] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – Mining Discriminative Components with Random Forests," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[3] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *arXiv preprint arXiv:1409.1556*, 2014.

[4] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," in *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[5] G. Csurka, C. Bray, C. Dance, and L. Fan, "Visual Categorization with Bags of Keypoints," in *Workshop on Statistical Learning in Computer Vision (ECCV)*, 2004.