UMEÅ UNIVERSITET

# Optimizing Fast Food Image Classification: A Comparative Study of VGG, ResNet, and Additional CNN Architectures in Pre-Trained Models

## Impact of Pre-Trained CNN Architectures on Accuracy and Performance

Hadi Saghir

Freestanding course: 5TF078

7,5 hp

2023

Git Repository: https://github.com/Hadi-Saghir/Fast-Food-Image-Classification

22-01-2023

# Introduction

## Project Overview

Image recognition is extensively utilized for food identification, making it a vital component within the ImageNet dataset. Food classification is often prioritized during the training process of numerous models. However, these pre-trained models are not exclusively designed for food classification but rather excel in transferring knowledge across various models intended to solve diverse tasks.

The objective of this project is to achieve image recognition and classification of ten distinct fast-food categories. The dataset, obtained from Kaggle, consists of 20,000 images equally distributed among the following ten categories: Burger, Donut, Hot Dog, Pizza, Sandwich, Baked Potato, Crispy Chicken, Fries, Taco, and Taquito.

## Problem Statement

This project aims to explore the impact of different pre-trained models on accomplishing specific tasks. Furthermore, it seeks to provide concise and precise recommendations supported by demonstrable reasoning to maximize accuracy by leveraging the full potential of pre-trained models in transferring learning through feature extraction for the classification of various fast foods.

The primary focus of this study is to gain a deeper understanding of utilizing pre-trained models to optimize the accuracy of image recognition for different types of foods, particularly fast foods. Consequently, the project emphasizes the analysis and selection of an appropriate pre-trained model, rather than fine-tuning and further development of the pre-trained model, in order to maximize accuracy.

22-01-2023

## Metric

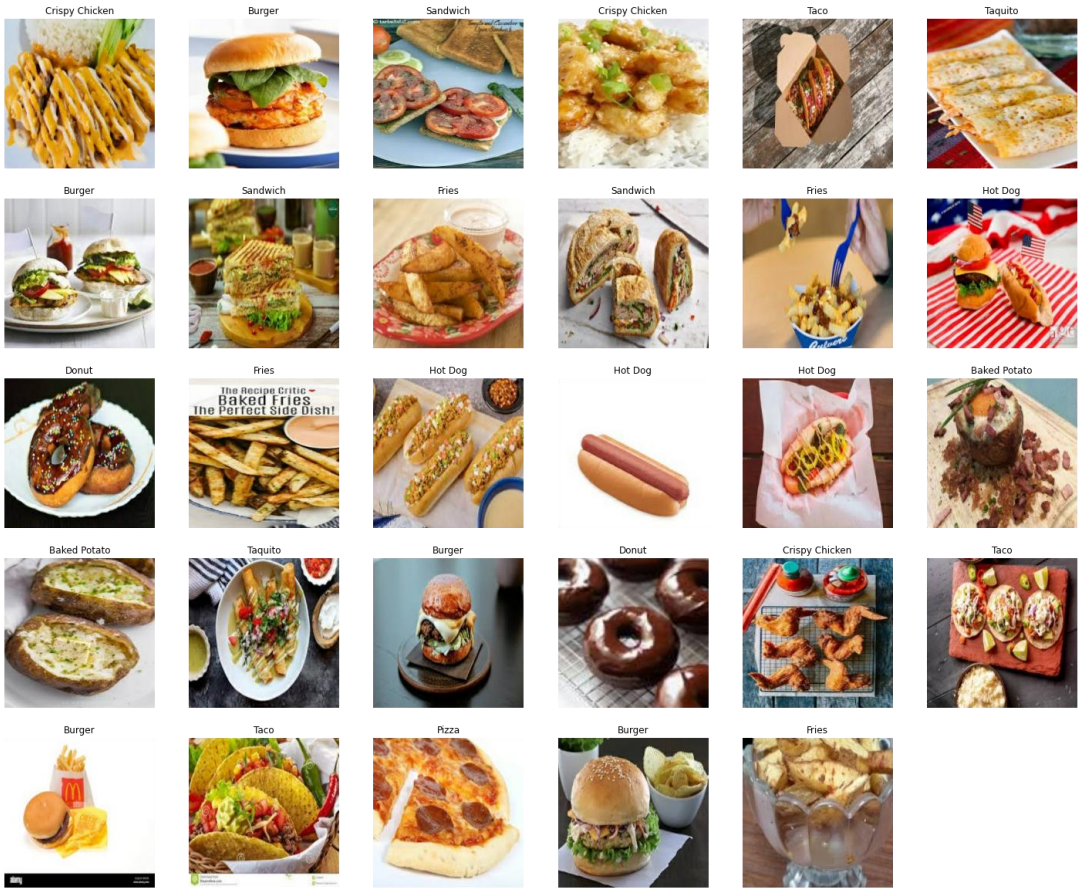
Accuracy serves as a common metric for classification tasks, representing the proportion of correctly classified instances (i.e., true positives and true negatives) out of the total number of instances in the dataset. The accuracy can be calculated as follows:

$$\text{Accuracy} = (\text{true positives} + \text{true negatives}) / \text{dataset size}$$

This metric is typically employed when the class distribution is balanced, indicating a roughly equal number of instances belonging to each class. In the case of this dataset, a balanced class distribution exists, and the goal is to minimize false negatives.

# Data Exploration

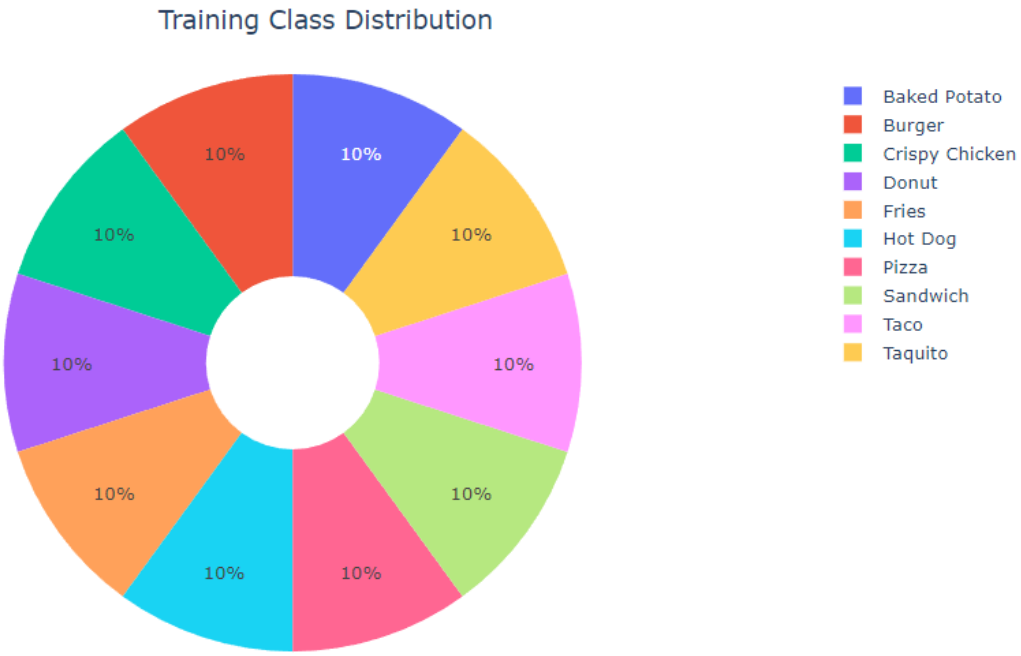
Img 1 shows pictures and their labels from the dataset:




Img 1

22-01-2023

Img 2 graphs the training class distribution




Img 2

Image 1 and 2 can be found under the "Prelude -> Data Visualization" section. Executing the code that generates Image 1 will result in a new image composed of various values from the dataset.

## Analysis

This analysis aims to evaluate the provided dataset, which will help us assess the suitability of different models for classification purposes.

The dataset exhibits an even distribution among the different classes; however, some food classes are represented in unconventional forms.

22-01-2023

Upon visualizing the data, it becomes apparent that there may be challenges in pattern detection, particularly the lack of focus on the food itself and its appearance in various forms.

- For instance, a hotdog can be presented in a jar, as part of a sandwich, garnished with sauces, or accompanied by fried onions. It can be depicted on a dish, being held, or featured in an advertisement.
- Crispy chicken burgers may be displayed alongside a side of fries, as is often the case with burgers. A pizza can be shown as a whole pie, sliced, or in its entirety.
- Burgers and crispy chicken items often overlap, making it difficult to distinguish between them. Personally, I had some hesitation when classifying the crispy chicken burger.

When determining the appropriate model for such images that exhibit significant variations within the same class, it is crucial to consider a large model capable of dissecting the image into different components. A seamless transition is needed from analyzing the overall image, filtering out irrelevant elements, focusing on the medium-sized portion containing the food item (shape), and finally attending to the finer details (texture).

### Facts on the dataset

Here are some key facts about the dataset:

- Image resolution: 256x256 (RGB)
- Dataset size: 20,000 images in total
- Total classes: 10
- Data per class: 2,000 images per class

## Literature Review

For this project, I will utilize two primary sources: the course book and a research article titled "Recognition and Classification of Fast Food Images." The course book serves as a valuable resource for understanding different algorithms and techniques applied in this project.

The research article focuses on an in-depth study of image processing technology and its prevalence in food recognition. The authors employ powerful machine learning techniques to

22-01-2023

recognize and classify various types of fast food images. Specifically, they utilize a pre-trained Convolutional Neural Network (CNN) model, the ResNet50, as a feature extractor for training an image category classifier.

The researchers argue that CNNs possess the ability to learn rich feature representations, which have been proven to outperform manual feature extraction techniques such as Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), or Speeded Up Robust Features (SURF). They train a multiclass linear Support Vector Machine (SVM) classifier with the extracted CNN features to classify fast food images into ten different categories. Through experiments conducted on two benchmark databases, they achieved an impressive success rate of 99.5%, surpassing the accuracy obtained using Bag of Features (BoF) and SURF.

In this project, I will focus on feature extraction from pre-trained models rather than emulating manual techniques. This approach aims to enhance overall performance, robustness, generalization, and understanding of the task of recognizing and classifying fast food images.

It is worth noting that the datasets used in the research article and the Kaggle-imported dataset differ. The researchers' dataset experiences fewer areas of contention compared to the Kaggle dataset. For instance, their hamburger category specifically refers to beef burgers and doesn't commonly include a side of fries. Moreover, their dataset comprises more specific types of food, such as chocolate cake, club sandwich, samosa, and others.

## Algorithm and Techniques

Transfer learning is a technique employed to leverage the knowledge acquired by a pre-trained model on a different but related task to improve the performance of a new model on a similar task. This technique is particularly beneficial in deep learning tasks like image and video recognition, where training a model from scratch requires a substantial amount of labeled data.

22-01-2023

There are two main approaches to transfer learning in CNNs:

- Feature Extraction: In this approach, the pre-trained model functions as a fixed feature extractor. The last fully connected layers of the pre-trained model are removed, and a new fully connected layer is added to adapt the network to the new task. Only the weights of the new fully connected layer are trained, while the remaining layers' weights are held fixed. This approach is useful when the new task shares similar input data with the pre-trained model but differs in output classes.

- Fine-Tuning: In this approach, the pre-trained model serves as a starting point, and the weights of some or all of the layers are further trained to adapt the model to the new task. Fine-tuning is suitable when the new task involves similar input and output data as the pre-trained model. However, this specific approach will not be utilized in this project.

To gain a better understanding of the pre-trained model's behavior and its adaptation to our task, it is essential to analyze the architecture of the Convolutional Neural Network (CNN). CNNs are deep learning neural networks commonly used for image and video recognition. They are designed to learn spatial hierarchies of features automatically and adaptively from input data. The name "convolutional" comes from the application of convolution, a mathematical operation that combines input data with learnable parameters (filters) to extract features.

A typical CNN consists of multiple layers, including:

- Convolutional layers: These layers play a crucial role in a Convolutional Neural Network (CNN) by applying filters to the input data. Each filter, represented as a small matrix of weights, is convolved with a specific region of the input data known as the "receptive field." The result is a "feature map" that captures relevant features from the input.

22-01-2023

- Pooling layers: Pooling layers are employed to reduce the spatial dimensions of the feature maps. By aggregating information from neighboring regions, typically through max pooling, these layers effectively downsample the data. This process not only reduces computational requirements but also enhances the network's ability to handle slight translations in the input.

- Fully connected layers: These layers utilize the extracted features to make predictions. Neurons in these layers receive input from all the neurons in the previous layer, establishing complete connectivity. The output of the fully connected layer is a set of scores or probabilities associated with each class in the dataset.

- Activation layers: Activation layers introduce non-linear transformations to the output of the preceding layer. These transformations, implemented through activation functions like ReLU, sigmoid, or tanh, enable the network to capture complex and non-linear relationships between the input data and the output. Activation layers enhance the network's learning capabilities

Benchmarking serves as a valuable process to assess the performance of a model compared to established standards or other models. By measuring its performance against predefined criteria, we can evaluate the model's effectiveness and identify areas that can be further improved.

A confusion matrix provides a visual representation of the model's performance on a given dataset with known true values. It organizes predictions into a table where rows represent instances in predicted classes and columns represent instances in actual classes (or vice versa). The matrix facilitates the identification of any confusion between classes, highlighting common misclassifications.

The confusion matrix includes four values: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). These values serve as the foundation for calculating various performance metrics, such as accuracy, precision, recall, and F1-score.

To leverage benchmarking and the confusion matrix together, the model is first trained and evaluated on a test dataset. The confusion matrix helps determine the number of true positives, false positives, true negatives, and false negatives. Subsequently, the model's performance can be compared to established benchmarks or other models using metrics like accuracy, precision, recall, and F1-score. This comparative analysis allows us to gauge how well our model performs relative to others and identify potential areas for improvement.

## Methodology

The methodology employed in this project entails a comprehensive analysis of the architectures of various pre-trained models, utilizing benchmarking and confusion matrix evaluation. The following steps outline the general approach:

- Architecture analysis: Thoroughly examine the architecture of each model, considering factors such as the number and types of layers utilized, as well as the parameter count. This analysis provides insights into the complexity and computational demands associated with each model.

- Model selection: Choose a set of diverse and suitable models for further evaluation. By incorporating a variety of models, we can explore different approaches and capabilities.

- Dataset preparation: Acquire the dataset from Kaggle and conduct a thorough analysis to gain a comprehensive understanding of its properties, such as data distribution, size, and quality.

- Dataset preprocessing: Prepare the dataset by applying necessary preprocessing operations, which may include resizing the images, ensuring consistency in format, or normalizing the data.

- Model training and evaluation: Train each selected model on the preprocessed dataset. Evaluate the performance of each model using benchmarking techniques and the accuracy metric. Utilize a confusion matrix to visualize the model's performance and assess its ability to classify fast food items.

- Model comparison: Compare the performance of the models and perform a detailed analysis of the results obtained. Consider factors such as architecture, parameter count, and computational requirements when assessing the strengths and weaknesses of each model.

- Model selection: Based on the analysis and evaluation results, select the best-performing model for the given task. Take into account factors such as accuracy, suitability for the specific problem domain, and the overall balance between model complexity and computational efficiency.

While it is good practice to employ various evaluation methods and test models on different datasets for a comprehensive assessment, this project focuses solely on evaluating the models' performance on the fast food dataset.

## Result

The adopted approach in this project involves utilizing four pre-trained models fine-tuned on the 'food' subtree of the ImageNet dataset: VGG16, ResNet50, DenseNet121, and BiT. In the report, these models will be referred to as mymodel.1, mymodel.2, mymodel.3, and mymodel.4, respectively. The specific architectures of these models are as follows:

These models will be further referred in the report as,

mymodel.1:

- BatchNormalization
- Flatten

- Dense with ReLU activation
- LeakyReLU activation
- Dense with softmax activation

mymodel.2:

- GlobalAveragePooling2D
- Flatten
- Dense with ReLU activation
- Dropout
- Dense with softmax activation

mymodel.3:

- Flatten
- Dense with softmax activation

mymodel.4:

- Flatten
- Dropout
- Dense with zero initialization and softmax activation

The benchmarking results for each model are as follows:

VGG16 (Block 1 to 4) (untrainable) -> VGG (Block 5) (trainable) -> mymodel.1:

- Total params: 31,499,594
- Trainable params: 23,863,306
- Non-trainable params: 7,636,288
- Test accuracy: 0.806

ResNet50V2 (untrainable) -> mymodel.2:

- Total params: 25,665,034

22-01-2023

- Trainable params: 2,108,426
- Non-trainable params: 23,556,608
- Test accuracy: 0.815

DenseNet121 (untrainable) -> mymodel.3:

- Total params: 8,097,354
- Trainable params: 1,059,850
- Non-trainable params: 7,037,504
- Test accuracy: 0.857

BiT (untrainable) -> mymodel.4:

- Total params: 23,520,842
- Trainable params: 20,490
- Non-trainable params: 23,500,352
- Test accuracy: 0.916

It's worth noting that VGG, ResNet50, and DenseNet121 were initially trained on ImageNet, while BiT was trained on a larger dataset. However, all models were fine-tuned on the ImageNet 'food' subtree to align with our task.

The confusion matrices, available in Attachment 2 of the file, provide additional insights:

- The VGG confusion matrix suggests a focus on texture features, which can lead to confusion when foods have similar textures, such as foods with a lot of cheese or fried foods. For example, it may confuse a taco with a taquito.

- The ResNet50 confusion matrix indicates a focus on shape features, resulting in confusion between similarly shaped items. For instance, a baked potato filled with cheese might be confused with a taco filled with cheese.

22-01-2023

- DenseNet121 strikes a balance by considering multiple features, leading to a more balanced confusion matrix with less bias towards specific features.

- BiT demonstrates the importance of the dataset on which the pre-trained model was trained. Compared to ResNet50, BiT performs significantly better in classification tasks.

For a more comprehensive analysis of the models and confusion matrices, please refer to the linked code file.

## Discussion

Upon analyzing the dataset and conducting confusion analysis, it became evident that the models encountered difficulties with certain images. To improve the performance and address these issues, refinement of the dataset could be beneficial. Here are some suggestions for refining the dataset:

- Adding classes that differentiate between similar food items, such as distinguishing between baked potato cubes and baked potato slices or differentiating between crispy chicken burgers and beef burgers.

- Removing images that do not align with a specific pattern. However, it is important to note that eliminating such images might result in training a model that is not capable of handling real-world situations, where burgers, for example, are often served with a side of fries.

During the training process, the models learn to extract features from the images, including edges, textures, and shapes, which are crucial for object recognition. These features are captured in the weights of the convolutional layers within the model.

22-01-2023

VGG architecture excels at extracting fine-grained features, such as edges and textures, from images. ResNet, on the other hand, leverages residual connections to extract both fine-grained and high-level features. DenseNet incorporates dense connections, enabling the network to use features from previous layers efficiently. DenseNet is recognized for its ability to extract both fine-grained and high-level features while promoting feature reuse.

Further research could involve fine-tuning the DenseNet121 model on the 'food' subtree of the ImageNet dataset and utilizing the fine-tuned model to improve the task at hand. The DenseNet architecture's focus on both texture and shape makes it a promising candidate for achieving high accuracy in fast food classification.

Despite ResNet50's struggle with Taco, which could be attributed to its emphasis on shape over texture, a similar limitation was observed in the BiT model. The architectural constraints of ResNet influenced the performance of the fine-tuned BiT model. In contrast, DenseNet outperformed BiT on the confusion matrix, displaying fewer areas of confusion and a more prominent diagonal line. The ability of DenseNet121 to consider both texture and shape attributes makes it less prone to confusing certain fast-food items with similar brioche bread or cheese toppings. Thus, fine-tuning the DenseNet121 model holds the potential to surpass the 94% accuracy achieved by BiT, based on the analysis of architecture, dataset, and the significant improvement seen from ResNet50 (0.815) to BiT (0.916).

In the research article "Recognition and Classification of Fast-Food Images," which employed ResNet50, the authors effectively utilized deep learning techniques to recognize and classify various categories of fast-food images. Their remarkable achievement of 94% accuracy aligns with the conclusions drawn in this project. Combining the understanding of feature extraction from different pre-trained models and the researchers' utilization of both pre-trained and handcrafted methods can lead to the development of an improved model for fast food classification. Leveraging a more advanced pre-trained model provides a solid foundation for feature extraction, potentially yielding superior results, as the researchers mentioned that feature extraction from pre-trained models "has been proven to perform better than other manual feature extraction techniques."

## References

The research article titled "Recognition and Classification of Fast-Food Images" by Akhi, Amatul, Akter, Farzana, Khatun, Tania, Uddin, Mohammad, Uddin, Shorif (2018) provides valuable insights into the recognition and classification of fast food images. The article was published in the Global Journal of Computer Science and Technology.

On the other hand, the course book "Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems" by Aurélien Géron (2021) is an authoritative resource that covers various aspects of machine learning. The book offers practical guidance, utilizing tools such as Scikit-Learn and TensorFlow, to build intelligent systems.

Both the research article and the course book contribute to the understanding and implementation of machine learning techniques. While the research article focuses specifically on the recognition and classification of fast-food images, the course book provides a broader perspective on machine learning concepts, tools, and techniques.

By combining the insights and knowledge gained from these sources, one can enhance their understanding of machine learning and apply it effectively in real-world scenarios.

## Attachment 1

Link to the Lab:
https://colab.research.google.com/drive/1HBTon88WaeCFS38OTVF3TKKSnttbXZGx?usp=sharing

Link to the BiT model used:

https://tfhub.dev/google/experts/bit/r50x1/in21k/food/1

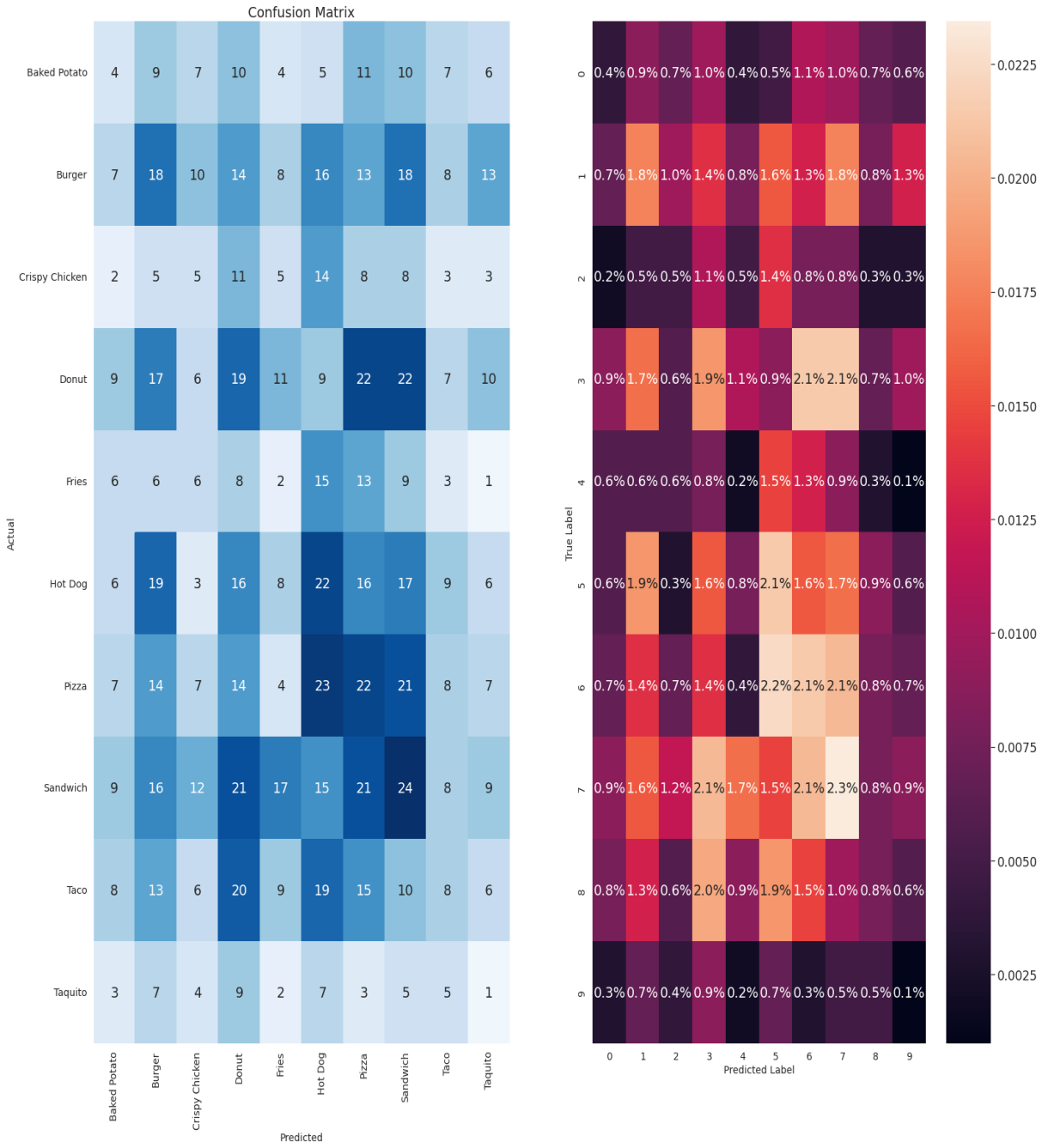
Link to the research:

https://www.researchgate.net/publication/350845216_Recognition_and_Classification_of_Fast_Food_Images_Recognition_and_Classification_of_Fast_Food_Images

# Attachment 2

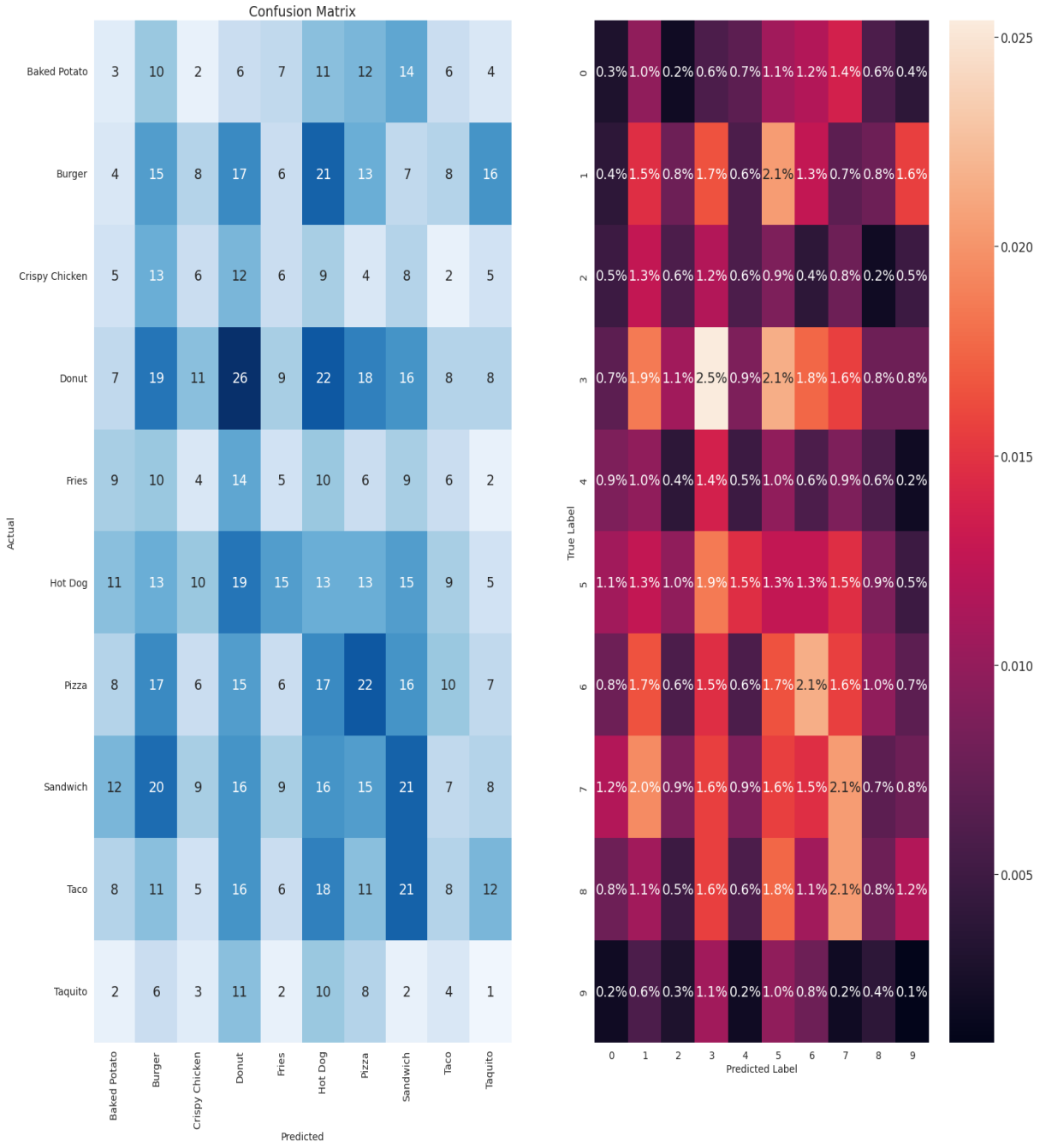
22-01-2023

Confusion matrices



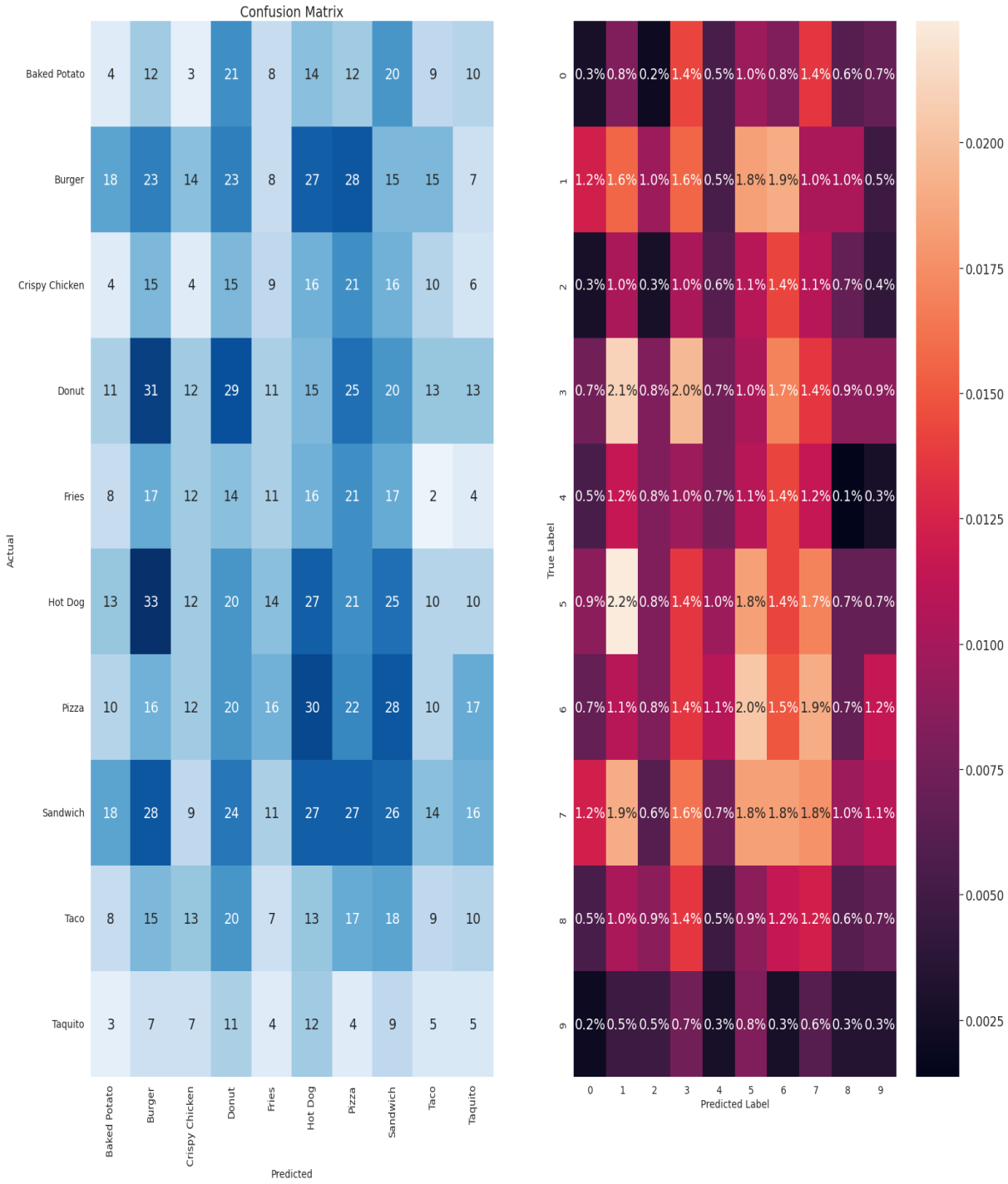

BiT

22-01-2023



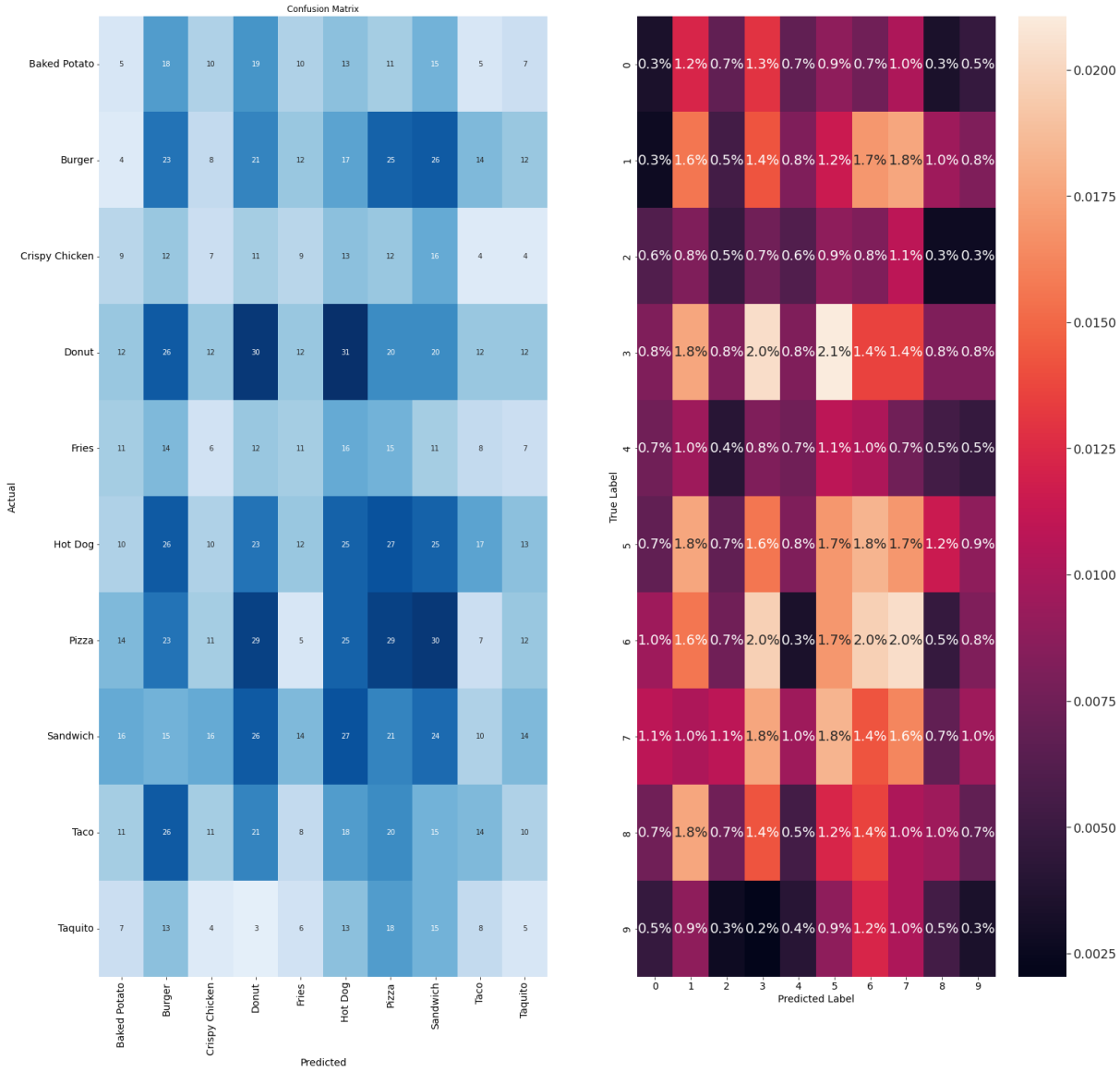

DensNet

22-01-2023




ResNet50

22-01-2023




VGG model