Module INF473V: Cheese Classification Challenge

Lucas MEBILLE
École polytechnique
lucas.mebille@polytechnique.edu

Adib MELLAH
École polytechnique
adib.mellah@polytechnique.edu

Abstract

The objective of this project was to design an image classification model capable of recognizing 37 varieties of cheese, ranging from Camembert to goat cheese and Raclette. The uniqueness of this project lay in training the model exclusively using images artificially generated by generative AI models such as DALL-E or Stable Diffusion. The performance of our model was evaluated during a one-month Kaggle challenge. Our initial approach focused on optimizing the image generation models to create a high-quality, realistic, and diverse training dataset for each variety of cheese. We then enhanced the classification model by building upon an existing architecture, DinoV2. Ultimately, we succeeded in achieving an accuracy score that outperformed the CLIP model in a zero-shot setting.

1. Introduction

In the field of computer vision model development, image classification is a vast domain with applications ranging from object recognition to medicine. This project focuses on an original challenge: classifying 37 varieties of cheese using artificially generated images. By leveraging generative AI models such as Stable Diffusion, we created a synthetic dataset to train our classification model.

Choosing to work with generated images instead of real-world data offers several advantages, including the ability to generate data in large quantities and to represent diverse conditions and contexts. However, this approach also presents unique challenges, such as ensuring that the synthetic images are sufficiently realistic and diverse to enable effective learning.

In this report, we detail our methodological approach, which includes generating the synthetic training dataset, improving the classification model using the DinoV2 architecture, and employing advanced techniques such as fine-tuning with DreamBooth and data augmentation. Additionally, we incorporated optical character recognition (OCR) techniques to enhance classification accuracy in specific cases.

Our results demonstrate that an approach using artificially generated images can compete with the performance of models trained on real-world data, opening new possibilities for creating training datasets in domains where real data is scarce or difficult to obtain. This project also highlights the importance of optimizing generative models and maintaining rigor in data selection and augmentation to achieve optimal performance.

2. Synthetic trainset generation

The first step of the project, and arguably the most crucial, was generating the synthetic training dataset. With the advancement of diffusion models such as DALL-E and Stable Diffusion, along with their various versions, we had numerous options for image generation.

Initially, we imported different models and experimented with various prompts to evaluate their raw performance. We divided the task of testing different versions of the models available on HuggingFace, such as Stable Diffusion 1.5, SDXL, SDXL Lightning, and SDXL Turbo [7]. Ultimately, we decided to use Stable Diffusion 1.5, as it produced higher-quality images.

As shown in figure (1), the initial results were not very conclusive. Since the models did not inherently have a deep understanding of all the cheeses under study, they captured and represented the specific features of each class in a very limited way. The generated images lacked diversity, and distinguishing between the different classes was challenging.

(a) Parmesan (b) Brie de Melun (c) Mimolette

Figure 1. Stable Diffusion with a simple prompt.

At this stage, we had not yet refined the classification

model, and our score was still very low (2).

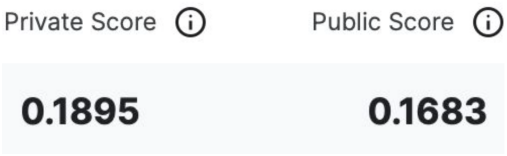


Figure 2. Scores at the beginning of the Kaggle challenge.

2.1. Use of DreamBooth

To generate a more realistic synthetic dataset that accurately represents the different classes of cheese, we fine-tuned Stable Diffusion 1.5.

For this, we considered two approaches: IP-Adapter [2], explored by Lucas, and DreamBooth, implemented by Adib [1]. IP-Adapter proved to be quite complex to implement, and the native image variations it allowed were rather limited. Therefore, we decided to focus on the second solution. We used DreamBooth, a fine-tuning model for text-to-image diffusion models. The idea is to input images of a specific type of cheese along with its corresponding class name (e.g., "Parmesan") and output a personalized text-to-image model that encodes a unique identifier referencing the cheese. During inference, this unique identifier can then be embedded into various sentences to synthesize the cheese in different contexts.

Using this approach, we trained 37 personalized DreamBooth models, one for each type of cheese, utilizing the images from the validation set.

Once the 37 models were available, we generated several synthetic training sets using different prompting strategies. Initially, by employing simple prompts closely related to the token used during training (e.g., the cheese's name), we obtained realistic and accurate images for most of the classes (figure 3).

To obtain more diverse images with specific scenarios, such as "parmesan on pasta" or "a tomato salad with mozzarella," we used customized prompts for each class. These personalized prompts also allowed us to address DreamBooth's underperformance for certain classes by specifying geometric details, such as the "heart shape" of Neufchâtel, or generating more varied views by refining lighting and viewing angles (figure 4). However, we observed that using overly detailed prompts was counterproductive [6]. The cheese token became lost in the surrounding context, leading the model to generate anomalies where the cheese and its context were awkwardly merged.

Finally, for cheeses that were not properly understood by DreamBooth during the initial fine-tuning, a second fine-tuning with additional training steps was usually sufficient to produce accurate and realistic images.

(a) Tête de moines (b) Époisse (c) Raclette

(d) Mozzarella (e) Féta (f) Neufchâtel

Figure 3. Stable Diffusion 1.5 with DreamBooth fine-tuning.

(a) Scamorza

(b) Pouligny-Saint-Pierre

Figure 4. Improvement of generation with additional training steps and customized prompts.

Figure 5. Examples of anomalies generated with overly detailed prompts.

Even without making any specific modifications to the classification model, we significantly improved our score thanks to DreamBooth (6). This improvement was expected, given the quality of the generated images.

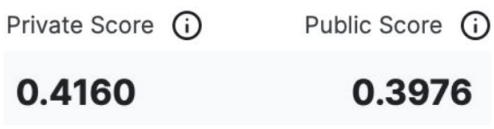


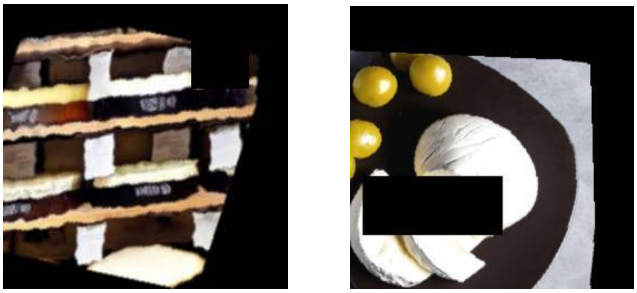
Figure 6. Scores after using DreamBooth.

2.2. Data Augmentation

Despite the efforts made to create a high-quality, extensive synthetic dataset, the number of images generated was still insufficient. Data augmentation significantly enriched the dataset, preventing overfitting and enabling better generalization.

We both experimented with various methods of augmentation, exploring different effects applied to the images and the number of additional images generated. After extensive testing, we finalized our approach: generating 20 additional images per image in the initial dataset, with highly aggressive modifications (7).

This approach notably improved classification accuracy (8). By significantly altering the images, we increased the diversity of the training data, which contributed to the model's improved performance.



(a) Beaufort (b) Mozzarella

Figure 7. Example of data augmentation.

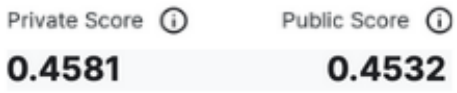


Figure 8. Scores achieved after data augmentation.

2.3. Refinement of the synthetic dataset

During the implementation of DreamBooth and the successive refinements of the customized models, we both had to fine-tune DreamBooth models for several cheese classes multiple times and initiate several waves of image generation. The richness and quality of our final dataset stem not only from the raw performance of DreamBooth and data augmentation but also from the rigor and meticulousness we applied when selecting the images to retain from those generated. Indeed, the final dataset is the aggregation of no fewer than four generated datasets, each carefully and rigorously curated by hand before applying augmentation.

2.4. Analysis of the relevance of the generated datasets

By analyzing the accuracy curves during validation with both real and synthetic images, we can measure the Sim2Real gap. For our final models, we observed a gap of approximately 35%. This indicates that our training dataset can still be improved.

Through the study of the confusion matrix generated from one of our experiments, we identified several issues inherent to the challenge. For example, some images in the validation and test datasets cannot be logically classified.

This is particularly true for goat cheeses, which appear in their specific categories but also under the broader category of "goat cheese" (e.g., "bûchette de chèvre"). Consequently, it is impossible to achieve a perfect dataset for this challenge. This ambiguity in the categories creates inevitable overlaps that impact the overall classification accuracy.

These findings highlight the complexity of the problem and the importance of considering these limitations when evaluating the models' performance.

3. Classification Model

The second part of the project focused on training and refining the architecture of our classification model. To achieve this, we started with pre-trained architectures available online, such as DinoV2 [5], EffNet [3], and ResNet [4].

3.1. Model Selection

The classification model was selected by conducting experiments at the beginning of the challenge on a basic dataset. The results were clear among the DinoV2, ResNet, and EffNet models (9), with fine-tuning applied only to the final layer. Consequently, we chose the most recent model: DinoV2.

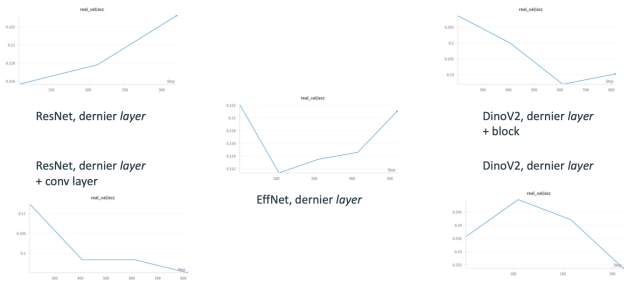


Figure 9. Comparison of model *accuracies*

Among the different versions of DinoV2, we compared the "Base" Vit-B version, which includes 86M parameters, and the "Large" Vit-L version, composed of 300M parameters. The "Large" version showed better results (10) despite

a predictably longer training time and higher memory requirements. Therefore, we selected this version, as we were unable to test the "Giga" Vit-g version with 1100M parameters due to hardware limitations.

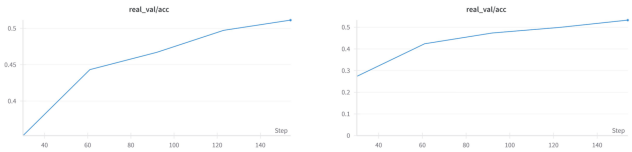


Figure 10. Comparison of DinoV2 versions: Vit-B on the left, Vit-L on the right.

3.2. Hyperparameter Selection

The selection of hyperparameters further enhanced the efficiency of the models. We relied on the Optuna library (https://github.com/optuna/optuna) to make these choices. Using this tool, we studied the influence of the optimizer and its parameters, a scheduler, and the batch size to achieve the best results in terms of both accuracy and training time.

3.3. Correction of obvious flaws through double fine-tuning.

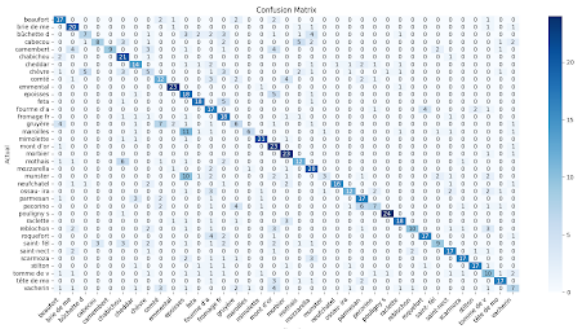
At this stage, the results were already promising, as evidenced by the confusion matrix obtained (11a). However, analyzing these results revealed some easily addressable flaws in the model. For instance, the confusion between Maroilles, Munster, and Époisses was not inevitable. Clear characteristics distinguish each of these cheeses, so we determined that the model could learn to differentiate them as well.

To achieve this, we trained the model again (a second fine-tuning after the initial one), focusing the learning efforts on areas of improvement using a specially designed dataset. This dataset primarily consisted of images of the three mentioned cheeses, each carefully selected to emphasize their differences, while still including images from other categories to prevent performance degradation elsewhere.

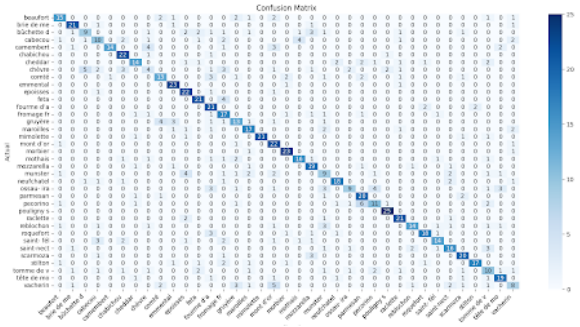
This approach was successful, significantly reducing the identified errors, as shown in the confusion matrices in figures (11a) and (11b).

3.4. Model ensemble

After optimizing all the aspects detailed earlier, we established a final training method. To further enhance the performance of our classification, we employed an ensemble method. By combining 2 to 4 identically trained models, we achieved our best results with three models (12). This represents the final improvement to our classification algorithm, reaching an accuracy of 64.15% (private score).



(a) Confusion matrix after the first training.



(b) Confusion matrix after the second training.

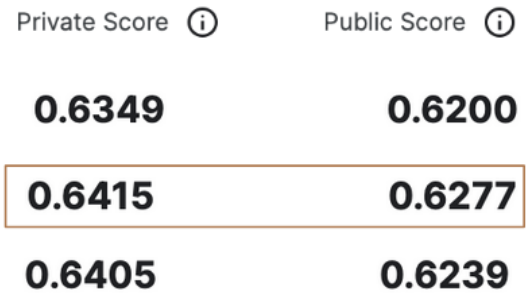


Figure 12. Results obtained for the ensemble of 2, 3, and 4 models.

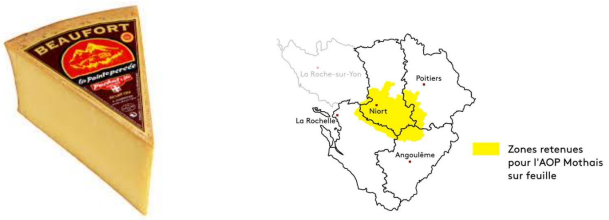
4. Classification using OCR.

Optical character recognition (OCR) techniques enable the detection and transcription of text present in an image. Specialized libraries can then be used to match the detected text with specific target words. In our case, OCR not only improved the organization of tasks but also contributed to more convincing classification results.

4.1. Motivation for using OCR.

The analysis of the validation dataset clearly highlighted the potential benefits of using OCR for our task. A significant proportion of the images were difficult to identify without relying on the text, even for a human, and in some

cases, identification was simply impossible (13).



(a) Beaufort (b) Mothais

Figure 13. Examples of images classifiable using OCR.

The implementation of OCR aimed to achieve three objectives: filtering out images with excessive text from the fine-tuning datasets for diffusion models without rendering them unclassifiable; enabling the models to focus on the intrinsic characteristics of the cheeses; and resolving identification cases that would otherwise be impossible for the model.

4.2. Choice of libraries and hyperparameters.

We aimed to implement an OCR-based classification method optimized for our problem. To this end, we tested numerous combinations of libraries, both for OCR (easyOCR, tesseractOCR) and for matching detected text with cheese names (difflib, fuzzywuzzy, rapidfuzzy). Evaluating accuracy alone was insufficient to identify the best choice; we also considered the number of decisions made by the algorithm, as decisions are only made when confidence surpasses a defined threshold.

The results strongly favored the combination of easyOCR and fuzzywuzzy (14). The decision threshold of 0.8 was selected as the best trade-off between decision frequency and accuracy (15).

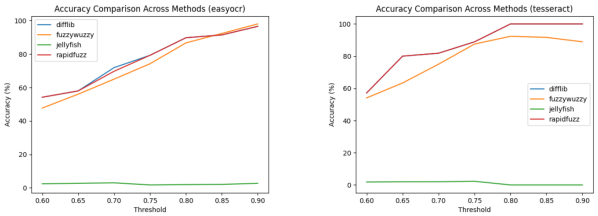


Figure 14. Comparison of metrics for different libraries.

4.3. Customization of the search scope and decision-making process.

Despite the high accuracy of OCR-based decisions, we knew there was significant room for improvement. A detailed analysis of all the images in the validation dataset highlighted correctable errors in the OCR classification. Among these, the over-identification of "fromage frais"

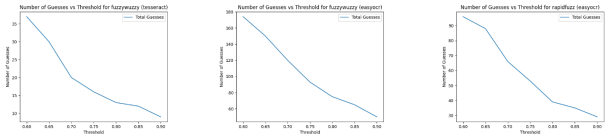


Figure 15. Number of decisions made based on the threshold for different combinations.

(fresh cheese) stood out as a clear issue, likely due to the frequent occurrence of the word "fromage" on packaging.

Most of the adjustments focused on expanding the search scope and refining the decision threshold. The search scope was broadened by adding frequent mentions (e.g., "parmigiano" or "fromage blanc") to the 37 original labels, helping the algorithm interpret detected text in varied contexts. Additionally, we made decisions more stringent (i.e., increased the required confidence threshold) for ambiguous labels like "fromage frais," while cross-checking for the presence of more distinctive cheese names.

These improvements refined the decision-making algorithm, addressing recurrent errors and achieving more decisions with fewer mistakes. The effectiveness of this approach is illustrated in figure 16.

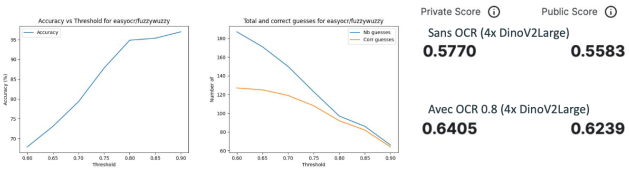


Figure 16. Contribution of OCR-based classification.

5. Conclusion

In conclusion, this project demonstrated the feasibility and effectiveness of using artificially generated images to train an image classification model. Our results show that with optimized image generation and a robust classification architecture, it is possible to achieve performance comparable to traditional models trained on real-world data.

This approach opens new possibilities for creating training datasets in fields where real data is scarce or difficult to obtain. The results could potentially be further improved by exploring new methods to combat overfitting in classification models, such as boosting or bagging techniques.

References

[1] Nataniel Ruiz Yuanzhen Li Varun Jampani Yael Pritch Michael Rubinstein Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv:2208.12242*, 2022. 2

[2] Sibo Liu Xiao Han Wei Yang Hu Ye, Jun Zhang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv:2308.06721*, 2023. 2

[3] Anton Kummert Ido Freeman, Lutz Roese-Koerner. Effnet: An efficient structure for convolutional neural networks. *arXiv:1801.06434*, 2018. 3

[4] Shaoqing Ren Jian Sun Kaiming He, Xiangyu Zhang. Deep residual learning for image recognition. *arXiv:1512.03385*, 2015. 3

[5] Théo Moutakanni Huy Vo Marc Szafraniec Vasil Khalidov Pierre Fernandez Daniel Haziza Francisco Massa Alaaeldin El-Nouby Mahmoud Assran Nicolas Ballas Wojciech Galuba Russell Howes Po-Yao Huang Shang-Wen Li Ishan Misra Michael Rabbat Vasu Sharma Gabriel Synnaeve Hu Xu Hervé Jegou Julien Mairal Patrick Labatut Armand Joulin Piotr Bojanowski Maxime Oquab, Timothée Darcet. Dinov2: Learning robust visual features without supervision. *arXiv:2304.07193*, 2024. 3

[6] Diane Larlus Yannis Kalantidis Mert Bulent Sariyildiz, Karteek Alahari. Fake it till you make it: Learning transferable representations from synthetic imagenet clones. *arXiv:2212.08420*, 2023. 2

[7] Dominik Lorenz Patrick Esser Björn Ommer Robin Rombach, Andreas Blattmann. High-resolution image synthesis with latent diffusion models. *arXiv:2112.10752*, 2021. 1