# APS360
# FINAL REPORT: INGREDIENT CLASSIFICATION

**Kevin Qu**
Student# redacted
fake.email@mail.utoronto.ca

**Yang Xu**
Student# redacted
yanguoft.xu@mail.utoronto.ca

**Julia Ye**
Student# redacted
fake.email@mail.utoronto.ca

**Henry Zhang**
Student# redacted
fake.email@mail.utoronto.ca

## ABSTRACT

For our project, we developed a CNN model that processed images of food dishes and classified the ingredients in the dish with an ANN classifier. The dataset of food images we used to train the model was found online and was heavily pre-processed to ensure consistency in images and standardized labels. We also collected new data to evaluate the performance of our model which resulted in a model accuracy of 81.4%. —-Total Pages: 9

## 1 INTRODUCTION

The goal of this project is to provide a convenient and intuitive means of discovering new recipes and expanding one's culinary repertoire. By using deep learning techniques to identify the ingredients in a picture of a dish, an algorithm can then search through a database of recipes to find those that include the same ingredients. This project is interesting and important because it can help people discover new recipes and expand their culinary horizons. Additionally, it has the potential to reduce food waste by suggesting recipes that use ingredients that might otherwise go to waste.

Deep learning is a particularly well-suited approach to the task of identifying ingredients in a picture of a dish for several reasons. Firstly, deep learning models are capable of learning complex features and patterns from large amounts of data. In the context of identifying ingredients in food images, a deep learning model can learn to recognize and distinguish different ingredients by analyzing patterns of pixels and colour within an image. This is especially important since food images can be very diverse and complex, with variations in lighting, camera angles, and the arrangement of the ingredients in the dish. Secondly, deep learning models perform well at recognizing and classifying multiple classes when given large amounts of training data. Although the number of ingredients used in common recipes is relatively small compared to large-scale general object classification models, deep neural nets are much better suited to multi-class classification tasks compared to traditional machine learning models like SVMs (Pin Wang et al. (2020)), making them the best choice of machine learning model for this project. Finally, deep learning can make use of large data sets to train, validate, and test the model. In our context, large data sets containing images of food such as (Goel (2021) are available with some data processing. Data sets such as this help train the model to recognize the wide range of variations in food images, increasing its accuracy in identifying future ingredients. Overall, the ability of deep learning models to learn from large amounts of data, recognize complex patterns, and analyze them makes them the best approach for the task of identifying ingredients in food images and suggesting alternative recipes using those same ingredients.

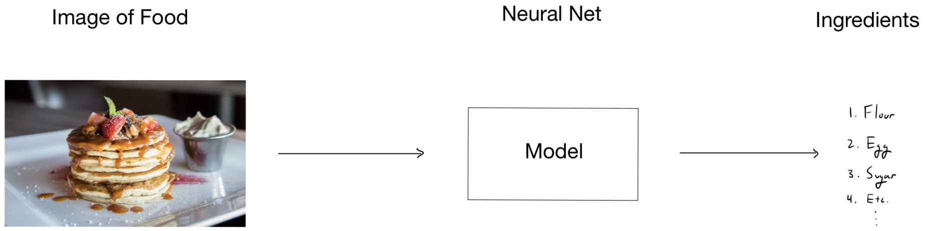


Figure 1: Illustration of Project Idea

## 2 Background & Related Work

There have been several research studies and existing products in the field of identifying ingredients in food images and suggesting alternative recipes. Here are five examples:

1. Recipe1M+ (mit): This is a large-scale, open-source recipe dataset that includes over one million recipes with associated images and metadata. The dataset has been used by several research studies to develop deep learning models for ingredient recognition and recipe recommendation. For instance, in a recent study, researchers developed a deep learning model that could recognize ingredients in food images with high accuracy and then suggest alternative recipes using those ingredients.

2. Whisk (Sawers (2019)): Whisk is a popular recipe and meal planning platform that uses natural language processing and machine learning to analyze recipe ingredients and suggest alternative recipes based on those ingredients. Whisk's algorithm can identify ingredients in recipes, account for common ingredient substitutions, and suggest recipes that use similar ingredients. Additionally, Whisk provides a personalized recipe recommendation system that takes into account user preferences, dietary restrictions, and nutritional goals.

3. sous-chef.ai (Hall (2021)): This app is intended to fulfill a similar need to our project. It's designed to take in an image of a dish, predict the ingredients, and attempt to find a recipe for the dish. Their app is designed to classify 75 of the most common ingredients, achieving 78.6% accuracy when using transfer learning.

4. Pic2Recipe (met (2019)): This model developed by Meta also performs a similar function. Pic2Recipe is designed to take in an image of a dish and output a recipe containing a title, ingredients, and cooking instructions. While similar, this project is slightly different in that it does not actually predict ingredients from an image, rather, it simply generates new recipes inspired by images of a dish.

5. Iwate Prefectural University (Zhu & Dai (2021)): This paper by Zhu et al. also develops a model that is capable of identifying ingredients of a dish from an image. Their model is based on transfer learning from ResNet50 and achieves about an 83% accuracy. However, their model is both trained and tested on heavily pre-processed data that only shows only one or two clear cut ingredients in each "dish" (ex. an image of only chopped carrots with nothing else in the image). This is not suitable for real-life dish classification applications where ingredients are more mixed together and there are additional objects/distractions in the image.

This demonstrates the ongoing research and development in the field of ingredient recognition and recipe recommendation, and that deep learning models are effective tools for achieving accurate and effective results.

## 3 Data Processing

### 3.1 Data Collection

The primary source of our data was a pre-existing dataset found online. This dataset (Goel (2021) contains images of thousands of dishes along with their ingredients and extraneous information such as the dish name and recipe. Many of the photos in these datasets were scraped from internet sources such as restaurant websites. As a result, some of these photos are very professional and artificial in appearance (ex. over-saturated photos, food plated in a special way, professional lighting, etc.) (figure 5).

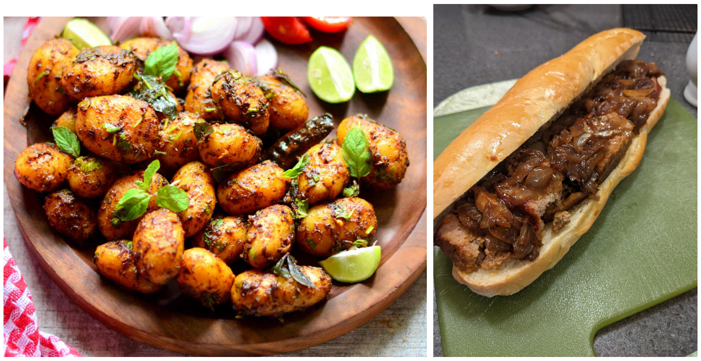


Figure 2: Example of Professional Photo vs Typical Photo

Since we are designing a model that is intended for general consumer use, we want the model to work well in everyday situations (eg. with typical phone cameras, normal/bad lighting, no extravagant presentation of the food, etc.). Thus, as part of our testing procedure, we will also be gathering some additional testing examples by taking photos of food in our lives to verify the performance of our model for a typical consumer.

### 3.2 Data Processing

Although we are relying on pre-existing datasets for our data, there was a significant amount of data processing that was required for this project. First, the images provided in the dataset are of different sizes, so we wrote scripts to crop the images to a standardized size. If there was an alpha channel, that was removed so that only the RGB channels remained. Second, we used data augmentation to increase the number of available examples to help our model generalize better to different conditions. For example, we could change the brightness of images to help our model become invariant to differences in the brightness of the surroundings. We also used the image manipulation functions from the torchvision library to apply basic transformations like changing the brightness, contrast, saturation, and rotating the image. Third, the ingredients list was standardized. The ingredients list originally contained unnecessary information like the amount of each ingredient, fluff text such as “make sure to clean the fish cleanly”, and “round Italian bread loaf” rather than just “bread”. We wanted the ingredients for each dish to be a clear-cut list of standard ingredients, so we combed through all the provided ingredient lists to standardize the information. To do this, we checked each word in the provided ingredients list with the food synset in WordNet. If the word was a food and a noun, we added the word to a new ingredients list for the dish. This combed out unnecessary descriptions and generalized ingredients that were too specific (ex. fresh lemon to lemon). We also had to do some additional filtering for plural vs singular nouns. Using the method described above, the dishes could have the same ingredient but with different labels due to some words being plural in the original data (ex. tomato vs tomatoes). This issue was resolved by using the inflect library to check every word to see if it was plural and change the label to match the singular version. Finally, the examples were shuffled and split into training, validation, and testing data with a respective 80%, 10%, and 10% split.

Although we initially planned to develop a model using all the ingredients present in over 11000 dishes, we quickly realized this would be impossible as there are over 5600 unique ingredients present. For this project, we decided to scale down to only 15 ingredients. These 15 ingredients were chosen by listing out the most popular ingredients and then choosing a variety of different types of ingredients from the most popular ones (ex. vinegar for sauces, chicken for meats, onions for vegetables). We chose a variety of ingredients rather than simply the top 15 because ingredients like ”olive oil” are present in almost all dishes, allowing the model to predict the same ingredients

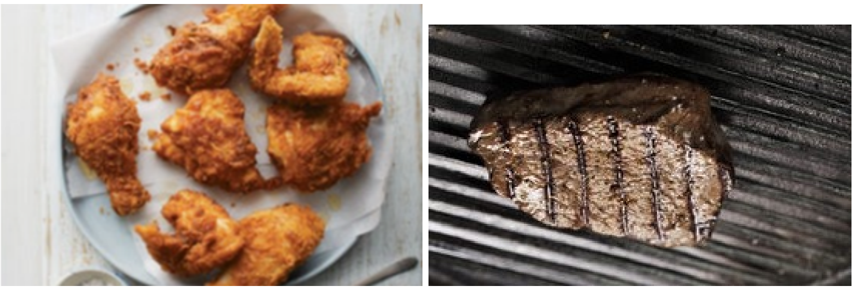


Figure 3: Cleaned data sample v. Post-augmented data sample

and achieve a high accuracy, which would not be an honest representation of how well deep learning performs for this task. After picking out only the dishes that contained those ingredients, we were left with 5749 training samples (before data augmentation), 1229 validation samples, and 1232 testing samples. In addition to the testing samples from the original data set, we collected an additional 50 samples by manually gathering realistic images. These 50 images were manually labelled with the chosen 15 ingredients and pre-processed in the same way as the other samples before being run through the model for testing. This was done to assess how well our model would perform in real-life situations, which is what we originally designed it for.

## 4 Architecture

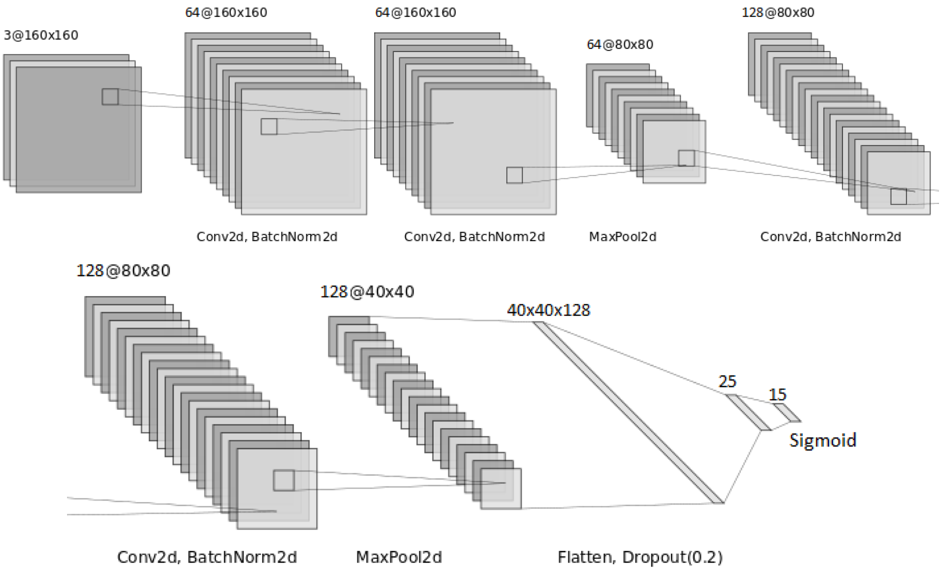


Figure 4: Primary model architecture left side (top) and right side (bottom) for legibility

As stated in the introduction, we are using a CNN-based model, specifically, a scaled-down version of a VGG architecture to perform our multilabel classification. Our model consists of 5,381,343 total parameters, of which there are 4 2D convolutional layers, 2 2D max pool layers, 4 2D batch norm layers, 1 dropout layer, and 2 linear layers. The input image is first passed into the convolutional layers where the features are extracted from the image. After the convolution layers, the features produced by the CNN were flattened and passed through fully connected layers to produce an output which predicts a probability for each ingredient. The output nodes then use a sigmoid activation function to provide a label between 0 and 1. These labels are then changed to 1 if greater than 0.5 and to 0 if 0.5 or less to indicate if the ingredient is present in the picture. To train the model, we used the Nesterov Adaptive Moment optimizer (NAdam) and Binary Cross-Entropy Loss (BCELoss), and hyperparameter tuning was done using GridSearchCV. This gave us a final learning rate of 0.003, batch size of 256 and 15 total epochs.

The accuracy of our first version of the primary model was ∼98% but we soon realized that was because we had over 1100 ingredients and a dish was highly unlikely to have that many ingredients. This resulted in many of the labels being filled with 0 which greatly inflated the accuracy. To resolve this, we changed the labels so that we are only considering 15 ingredients (as described in Section 3 (Data-Processing). Through tuning, we achieved a best primary model, which has a testing accuracy 78.4%. The performance of the model will be explored Section 6 (Results).

# 5 Baseline Model

The baseline model we will use to compare our neural network against is a multi-label k-nearest neighbours (kNN) classifier with feature extraction using the SIFT algorithm. On a basic level, the scale invariant feature transform (SIFT) from OpenCV takes a processed image, determines a number of keypoints in the image, and returns the image as local feature coordinates that are invariant to translation, scale, and other image translations (OpenCV). These features can then be fed into a multilabel kNN classifier from the Scikit-multilearn library to make a prediction (Zhang et al. (2007)). General kNNs work by grouping available training cases in a multi-dimensional space based on their features and classifying new cases by calculating which group the case is closest to (IBM). The multi-label kNN from Scikit-learn uses the classic kNN to find nearest examples to a class and Bayesian inference (a learning technique based on prior knowledge) to select assigned labels (Zhang et al. (2007)).

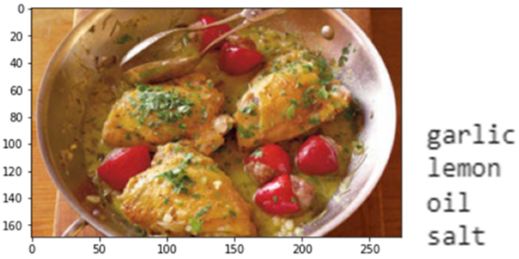


Figure 5: Example input picture (left) and output (right) from baseline model

Since our baseline model is not a neural network, there is no loss or learning curves we can provide. The accuracy for our baseline model was 54%. However, we noticed that some of the ingredients it correctly guessed were very common and present in a large portion of the dishes, such as salt, which can be seen in the diagram above. This suggests the baseline model would have trouble identifying dishes comprised of less common ingredients. This is a fair performance for the baseline model, but we believe that it has much room for improvement. Thus, we decided to go with the CNN model.

# 6 Results

## 6.1 Quantitative Results

Our best model achieved a training accuracy of 81.4%, a validation accuracy of 79.7%, and a testing accuracy of 78.4%. We determined that this was our best model because it had both the highest validation and training accuracy out of all versions. The accuracy of this model is similar to the accuracy of previously discussed related works (Section 2). While they did classify more ingredients, their models were also much larger, which is not something we can compete with given the resources we have available.

We also calculated the precision, recall, and F1 scores of all ingredients we used. Looking at the data, there are two interesting trends we would like to highlight. First, we noticed that all three scores were heavily correlated with the number of samples that contained that ingredient. For example, oil and salt were two of the most popular ingredients in our data set, and they correspondingly have

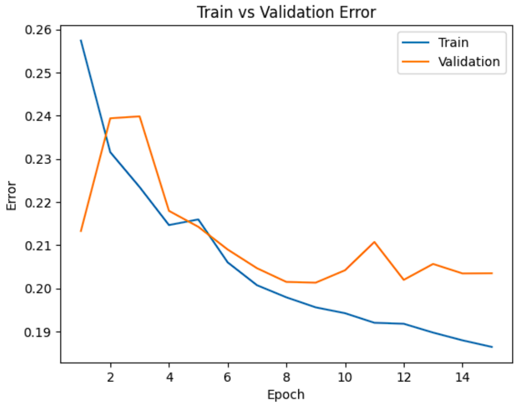
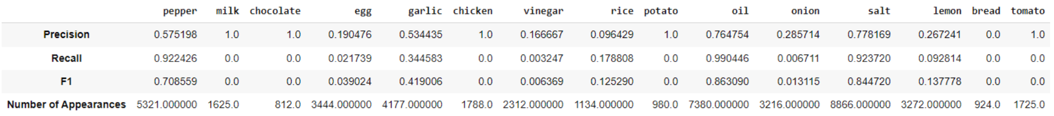


Figure 6: Training Curves



| | pepper | milk | chocolate | egg | garlic | chicken | vinegar | rice | potato | oil | onion | salt | lemon | bread | tomato |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Precision** | 0.575198 | 1.0 | 1.0 | 0.190476 | 0.534435 | 1.0 | 0.166667 | 0.096429 | 1.0 | 0.764754 | 0.285714 | 0.778169 | 0.267241 | 0.0 | 1.0 |
| **Recall** | 0.922426 | 0.0 | 0.0 | 0.021739 | 0.344583 | 0.0 | 0.003247 | 0.178808 | 0.0 | 0.990446 | 0.006711 | 0.923720 | 0.092814 | 0.0 | 0.0 |
| **F1** | 0.708559 | 0.0 | 0.0 | 0.039024 | 0.419006 | 0.0 | 0.006369 | 0.125290 | 0.0 | 0.863090 | 0.013115 | 0.844720 | 0.137778 | 0.0 | 0.0 |
| **Number of Appearances** | 5321.000000 | 1625.0 | 812.0 | 3444.000000 | 4177.000000 | 1788.0 | 2312.000000 | 1134.000000 | 980.0 | 7380.000000 | 3216.000000 | 8866.000000 | 3272.000000 | 924.0 | 1725.0 |

Figure 7: Precision, Recall, and F1 scores

the highest F1 score, indicating that they are better classified. However, for ingredients like onions, which are not nearly as popular, the F1 score is not nearly as high. The better classification for more popular ingredients might be due to 1) the abundance of training data for these ingredients compared to the less popular ones; 2) some degree of overfitting. Though overfitting might have occured, we conclude that the model still learned abstract information about the ingredients, as the model was not predicting salt and oil for every recipe. Second, we noticed that a few of our ingredients, like chicken, milk, and chocolate, had a recall of 0 and a precision of 1. From this and some additional experimentation, we believe that the model is very rarely predicting these ingredients, even when they are present, but when they do, it is always correct. This may be due to both a lack of training data for these ingredients (few dishes involve chocolate) and also because they do not have a clear indicator of their presence. These ingredients often take many forms (ex. shredded chicken, whole chicken, etc.) and have very common colours. This makes it very difficult for the model to learn any sort of general pattern for these ingredients, resulting in few predictions.

Moving forward, these results indicate that a larger data set with more balanced ingredients may help the model to better learn when certain ingredients are present, improving accuracy. However, this may be difficult to obtain as there are simply just a disproportionate number of dishes that use salt versus an ingredient like chicken. Furthermore, many of the ingredients are coupled, when chicken is present, salt is likely to be present as well, making it hard to balance the data set.

### 6.2 Qualitative Results

As seen from this example (Figure 9), our model is very good at predicting the common ingredients like oil, salt, and pepper. However, it incorrectly predicts rice, which may be due to the lack of data for this ingredient as previously discussed. The incorrect prediction of rice may also be contributed to by the visual similarity of the noodles in the dish with rice.

This example (Figure 10) shows that the model fails to recognize chocolate, possibly due to the reasons discussed in the previous section, but does predict onion and egg. The onion prediction may be due to the surrounding colours that are similar to an onion. More interestingly for the egg, from our experimentation, we believe the model has learned what a baked good is and that baked goods often have eggs. Despite eggs not being visible in things like bread and cake, we found our model often correctly predicts egg for these dishes. In this image, the model is likely predicting egg because the marshmallow looks more like cake than a marshmallow, leading the model to believe the dish is a baked good.

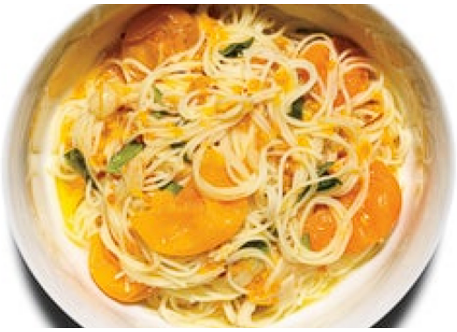


Figure 8: ”Sun Gold Pasta”
**True labels**: oil, garlic, salt, pepper
**Predicted labels**: oil, garlic, salt, pepper, rice

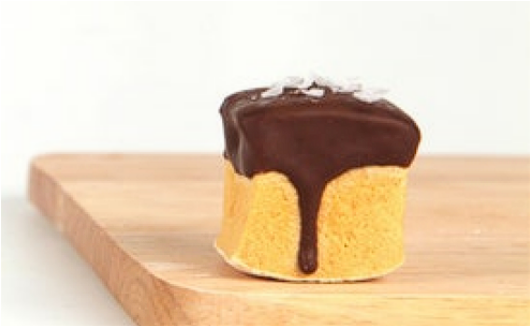


Figure 9: ”Chocolate Dipped Salted Caramel Marshmallows”
**True labels**: chocolate, oil, salt
**Predicted labels**: pepper, egg, onion

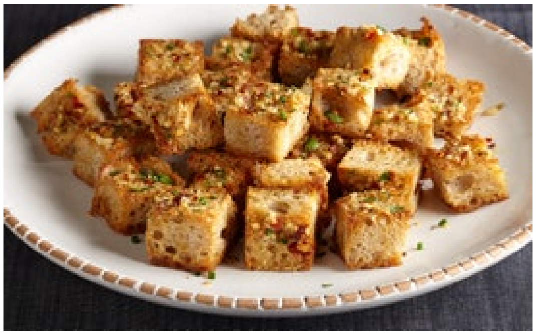


Figure 10: ”Savory Garlic Bread Bites”
**True labels**: garlic, oil, egg
**Predicted labels**: oil, lemon, egg

This example (Figure 11) shows the model correctly predicting egg for garlic bread, a baked good. This once again shows that the model has learned some sort of connection beyond the simple visual image that it is using to make its predictions. Surprisingly, our model failed to predict garlic. This may be due to the low resolution of the image which makes it difficult for our model to process the bits of garlic in the image and the infrequent use of garlic in baking/baked goods in our data set that may have dissuaded the model from predicting garlic.

### 6.3 Evaluation Of Model On New Data

To ensure the results are a good representation of the model's performance on new data, new samples were obtained from real-life pictures that are not in the original dataset. This means the new data did not influence the model's hyperparameters in any way and were an accurate representation of how well the model performs. Using this new data, the group was able to evaluate the model which resulted in an observed accuracy of 81.4%. In the context of ingredient identification in food pictures, this means our model will incorrectly predict an ingredient roughly 1 in 5 times. This value is comparable with the other pre-existing models performing a similar function of ingredient identification mentioned above which all had accuracies of 80%. We can conclude from this that our model meets the expectations in the domain of ingredient identification.

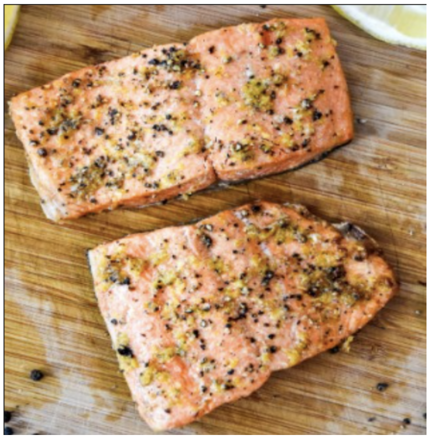


Figure 11: ”Lemon Pepper Salmon”
**True labels**: lemon, pepper, oil, salt
**Predicted labels**: oil, garlic, lemon, salt, pepper, rice

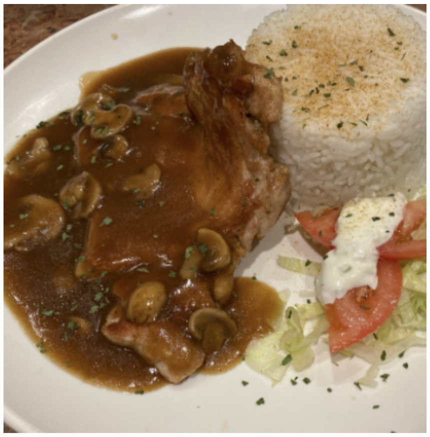


Figure 12: ”Teriyaki Mushroom Chicken with Rice”
**True labels**: chicken, rice, pepper, oil, salt, tomatoes, garlic
**Predicted labels**: rice, pepper, oil, salt, tomatoes, garlic, chocolate, potatoes

## 7 Discussion

### 7.1 Current Model

After investigating various models in computer vision, we found that our model performed decently across all scenarios, ranging from the initial dataset to new data as well. Over the course of the project, we learned a lot about the engineering challenges behind making deep learning models, especially in problems such as computational complexity. For example, we tried to use ResNet but found that VGG was a better performer and was less computationally intensive.

In terms of model performance, we noted that CNN's primarily learn from image features rather than contextual data. This has consequences in regional dish variety such as in cuisines where certain ingredients may look different from one regional dish to another. Our model also had comparatively lower precision than recall. This is because classifying an ingredient as not belonging to a dish is easier than the opposite. Another challenge we ran into was that the dataset was biased. From earlier, we learned that our dataset had a large number of food items that had oil, salt and pepper, which are understandably common ingredients. This led to the model frequently guessing these ingredients which led to good performance on testing data even on samples with no oil due to the high number of samples with oil.

Overall, we believe that our model performed better than expected for our task. Predicting ingredients from an image of a dish is a very difficult task because there are so many variables that change. The ingredients used in each dish could appear differently (ex. different sizes, colours, etc.) or not even be visually apparent in the image of the dish at all. There are also often over a dozen ingredients used in a single dish out of thousands of possible ingredients, making it difficult for any model to accurately predict every ingredient and for engineers to gather enough data to properly train a model for every ingredient. Furthermore, since we want our model to work for every day images, the model needs to be invariant to all environmental factors, such as the background of the photo and lighting. The accuracy we achieved on real-world data without the use of transfer learning is comparable to the accuracy achieved in previous research (section 2), so we see our model as a success.

### 7.2 Future Improvements

Although our model's performance is good, we also consider the next steps to improve it. The most significant next step for us is to consider other model architectures. Right now, transformers and attention mechanisms are state-of-the-art deep learning models with strong performance on computer vision tasks. As a possible next step, we could try to further investigate them to evaluate their performance on our multi-label tasks. We could also explore implementing an encoder/decoder model and then use the encoder portion of that to replace the convolution layers that we implemented. However, this might not provide a significant improvement as this type of model still has a very similar architecture to the CNN's we have been primarily using.

## 8 Ethical Considerations

The training data that we are using was from Kaggle, (Goel (2021)) which contained information about European and Continental dish that was scraped from an American digital food brand, Epicurious. Using this dataset can give rise to the following ethical issues and limitations:

- Eupicurious has been accused of its cultural bias: it undervalues the authenticity of dish from other cultures, and tends to provide information on an exotic cuisine through a white lens (Italie (2020)). Using this "whitewashed" dataset might lead to cultural misunderstandings.
- This limits our models to only producing reliable outcomes for European and Continental cuisines, and ignoring food from other cultures. Though the team members can introduce more cuisine by making customized datasets, all team members' mostly consume Asian and North American cuisines. There is still a lack of representations of cultural dishes.
- The dataset does not seem to contain any information about dietary restrictions. For example, there is a clear lack of vegan dishes in the dataset. This will rule out vegans in the user base.

## References

Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. URL http://pic2recipe.csail.mit.edu/.

Using ai to generate recipes from food images, Jun 2019. URL https://ai.facebook.com/blog/inverse-cooking/.

Sakshi Goel. Food ingredients and recipes dataset with images, Feb 2021. URL https://www.kaggle.com/datasets/pes12017000148/food-ingredients-and-recipe-dataset-with-images.

Brian Hall. Using ai to identify ingredients and suggest recipes, Dec 2021. URL https://medium.com/@brh373/using-ai-to-identify-ingredients-and-suggest-recipes-95482e2aca7d.

IBM. What is the k-nearest neighbors algorithm? URL https://www.ibm.com/topics/knn.

Leanne Italie. Epicurious is righting cultural wrongs one recipe at a time, Dec 2020. URL https://apnews.com/article/entertainment-race-and-ethnicity-29752fa5ecf5748c3e477cb63774d849.

OpenCV. Introduction to sift (scale-invariant feature transform). URL https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html.

Wei Pin Wang, Changxing Ding, Huiru Wang, Chuan Liu, X.B. Zhang, and et al. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning, Aug 2020. URL https://www.sciencedirect.com/science/article/pii/S0167865520302981.

Paul Sawers. How whisk is using its food genome to turn recipes into smart shopping lists, Dec 2019. URL https://venturebeat.com/ai/how-whisk-is-using-its-food-genome-to-turn-recipes-into-smart-shopping-lists/.

Zhang, Min-Ling, Zhou, and Zhi-Hua. Multilearn: Multi-label classification package for python, 2007. URL http://scikit.ml/api/skmultilearn.adapt.mlknn.html.

Ziyi Zhu and Ying Dai. Food ingredients identification from dish images by deep learning, Apr 2021. URL https://www.scirp.org/journal/paperinformation.aspx?paperid=108663.