Food identification project final report

1.Exploratory Data Analysis

Our data mainly contains 3 sets: train set, validation set, and test set. Each set contains lots of food images from different classes, and except for the test set, the other two sets are labeled. The train set is used for the learning process and the validation set is used for evaluating our learning algorithm to tune the parameters for feature selection. In the meanwhile, we also have to divide a subset to help us with choosing the best classifier among all the classifiers (or stacking).

We first divided the images in the train set in different classes using their labels to analyze the images better. Thus, we have 251 folders (classes) and each folder only contains the images associated with that class. A general analysis of the whole images in the train and validation sets informed us about different image sizes in the first place. This fact tells us that before starting any further analysis we have to resize all the images in different classes and sets to have the same image size for further processes. An example of a resized (downscaled) waffle image from the waffle class is represented below.

2.Challenges

Automatic food identification can be difficult since there are large amounts of food and thousands of different kinds of food in the world. Besides, food doesn’t have specific discriminative criteria, and it is hard to tell every ingredient from a cooked dish.

For humans, the picture of the food can be misleading sometimes like, for vision appreciation, food can be made into a flower or cute animal shapes. So it is even hard for humans to tell the class or ingredient. Meanwhile, humans may eat limited kinds of food in their life and have a specific preference for food, thus they can not tell the class of some food they have never seen before.

For machines, it is really tricky to tell the class of a dish with only one picture. And pictures sometimes will be influenced by the angle of view, the lighting conditions, but also the very realization of a recipe are among the sources of high intra-class variations. Since the light condition will change the color of the food, the angle to take this picture may influence the shape

or size of the food. All these uncertain factors will add complexity for machines to correctly identify the food.

3.Data pre-processing and data augmentation

For further and more detailed analysis, we analyzed the images in two random classes. We chose fried_rice and waffle classes for data analysis. We realized that each class contains many irrelevant images that we call outliers. For example, we found images train_043609 and train_101298 in the class of waffle and fried_rice, respectively, as represented below.

Moreover, we realized that the images in each class vary in terms of size, the number of pixels, light conditions, photo angles, food amount, rotation, food scale, and pose. On the other hand, we see that the images in each class have some specific similarities such as food color and shape. As an example, in the fried_rice class, the angles of taking photos are different, some pictures were taken from the upper direction while others were taken from front or side facets. And with the light condition and ingredient difference, the color of fried rice is different. Some

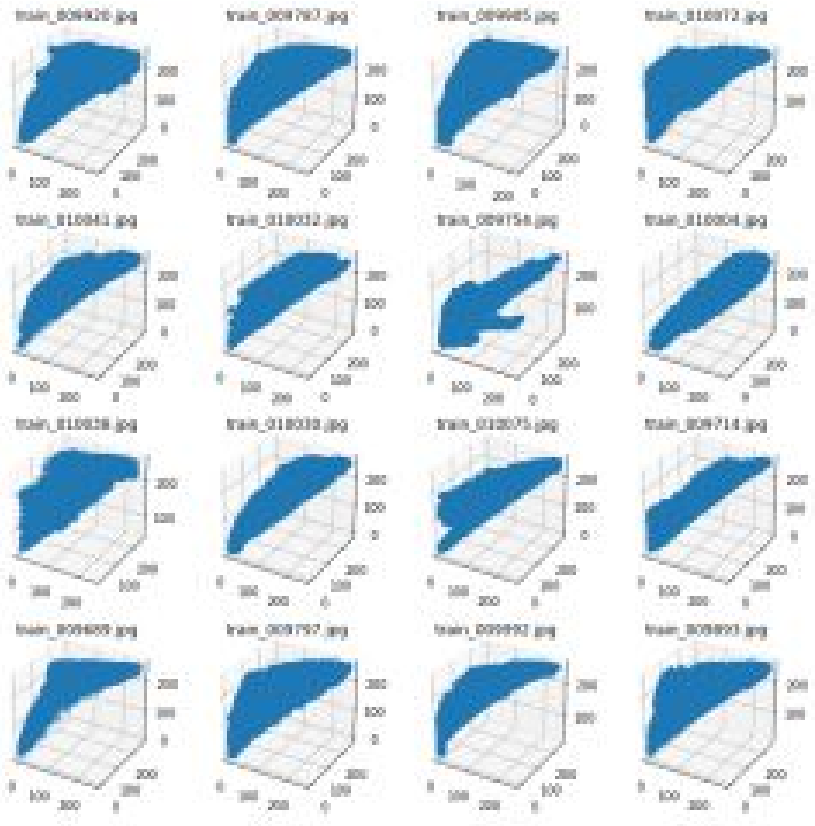
are deeper brown while others are light yellow. In addition, some images contain over 2 bowls of fried rice per image. The majority contain only one item per image. And the shape of fried rice is not quite regular since it is made of hundreds of rice and other side dishes such as eggs, shrimps, and shallot. Observed from the class, the fried rice could be put in plates, bowls or pots. As for the waffle class, it also remains the same problem of different photo angles and different amounts per image. Waffles are similar most in terms of color and texture on the food (cubic holes on the waffles) and they have regular shapes, such as square, triangle, or circle shapes. Waffles are often served with ice cream, butter or fruits. We can see these problems clearly from the pictures below.

Due to the facts mentioned above, we need to define the discriminative features of all these images in different classes, and give them appropriate weights based on the assigned category, so that we can capture the most similarities within each class and discriminate the food of each class from other classes.

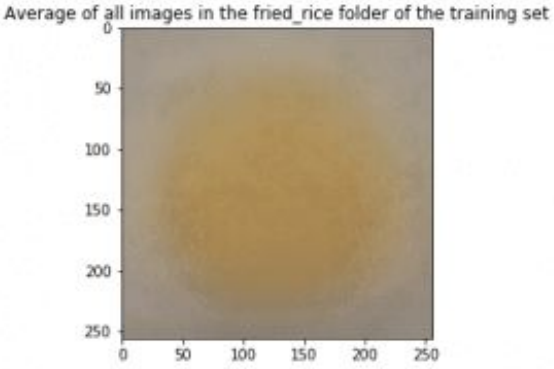
We also have to take into account those images that are somehow relevant to the desired food in that category but are not food themselves. As an example, the menu of a restaurant containing the word “waffle” or the waffle maker are all classified as being waffles. Most probably, we have to remove these images for more accurate classification.

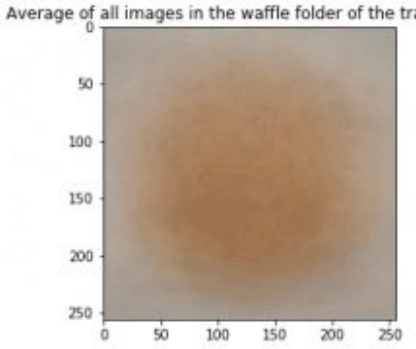
Moreover, one important fact about most of the images in all classes is the presence of other stuff in addition to the desired food. For example, the background view and the dishes on the side such as fruits, eggs, and potatoes in the waffle images might lead to misclassification. Therefore, after removing the outliers and resizing the images, in the next step, we have to distinguish the desired food from the background foods and other background stuff as much as possible. We got the idea of segmentation from "Automated flower classification over a large number of classes" in which they classify the flowers from 103 classes.

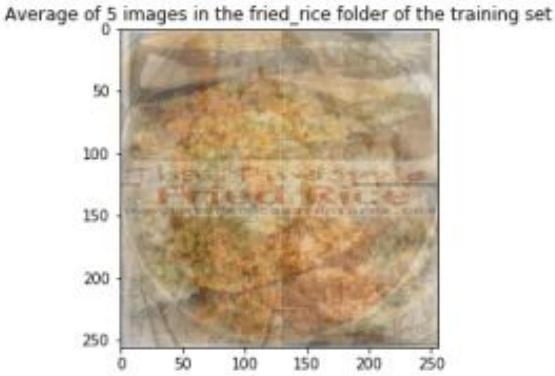
First, we plot each image in 3D space. From the 16 images in the cheesecake folder of the training set, we can find obvious patterns. All cheesecake images’ 3d plot tend to show an inverted triangle shape. And the outliers(which are 8 and 9) show different shapes.

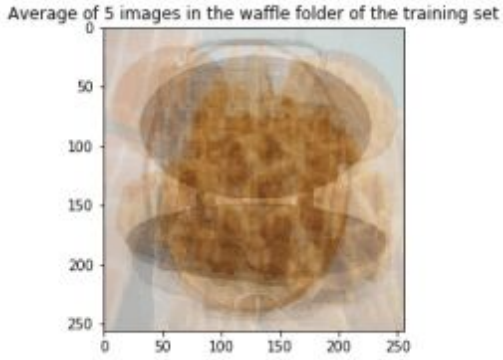


Moreover, we checked the centrality of images for 5 and all images in two classes of the training set. We did that by taking the mean of 5 and all (resized) photos in every class, respectively. We realized that in the waffle folder, the average of the images is almost in the center of the image. This was also true for the fried_rice class. Below we only show the centrality results of the images in the waffle folder.

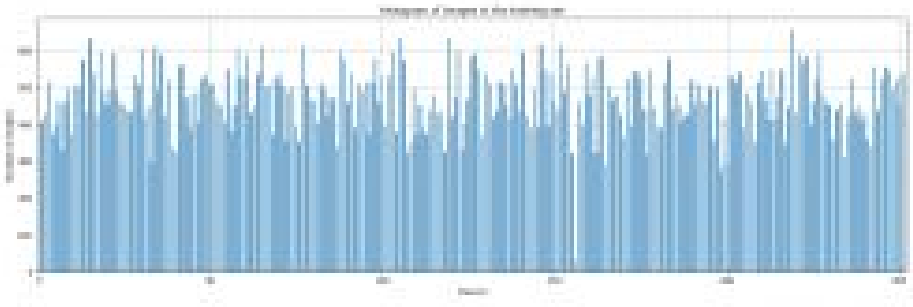


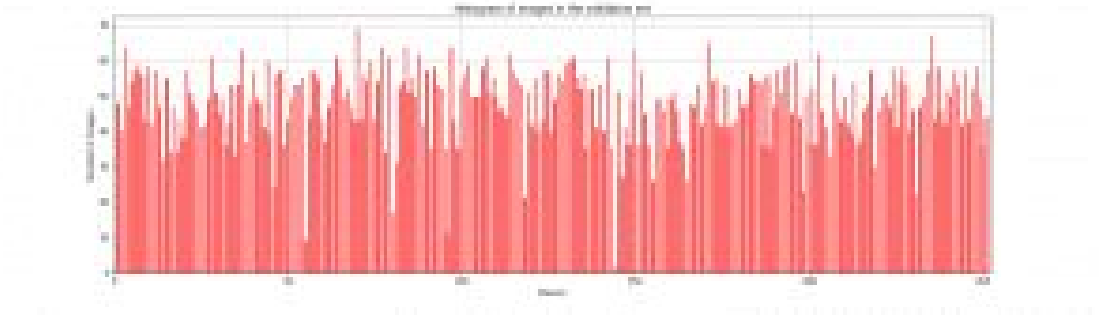




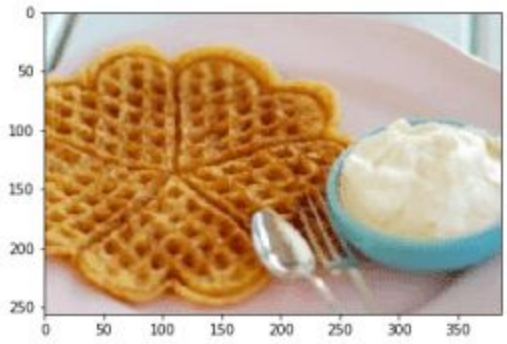


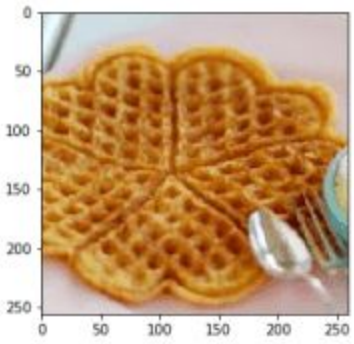
In the next step, we took the histogram of images in both training and validation sets to see the distribution of the amounts of images in different classes. By looking at the plots below, we can see that the distribution of the amount of images is almost similar (or close) in all classes in the training and validation set.

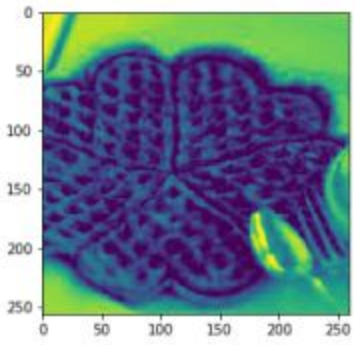




To reduce dimensions, we can look at the dimensionality of data from two aspects. One is the number of pixels constructing the image and one is the color. In cases where the color of the food is not very informative about the desired food, we can reduce the color dimension to one. Also in so many images, not all the pixels are informative about the desired food (such as the plate of the food, other dishes, and the background view). Therefore, we can do segmentation on the image to only focus on the pixels that represent the desired food. Dimension reduction will increase the classification accuracy, decrease memory consumption and is more computationally efficient. An example of dimensionality reduction in terms of the number of pixels and color for a waffle is represented bellow. The first figure shows the existing waffle image in the waffle folder. The second image represents decreasing the number of pixels to remove the uninformative part of the image. The last image shows the cropped image with only one-dimensional color rather than 3 colors.

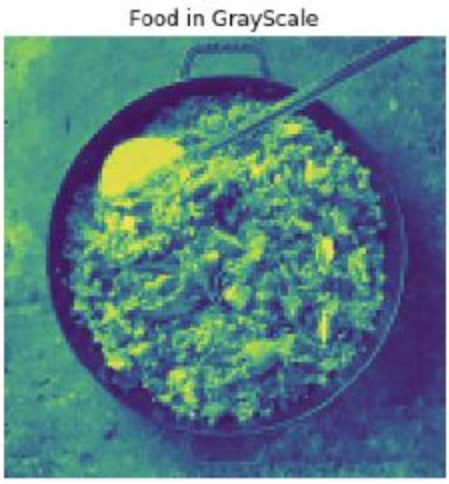






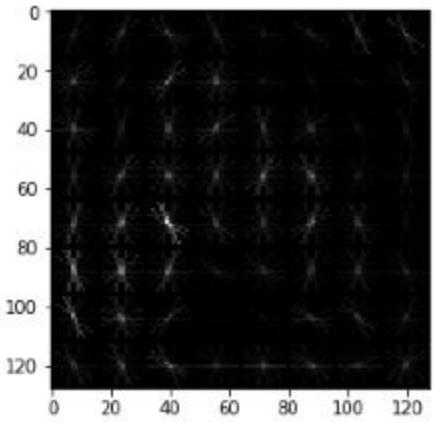
4. The comparison of models

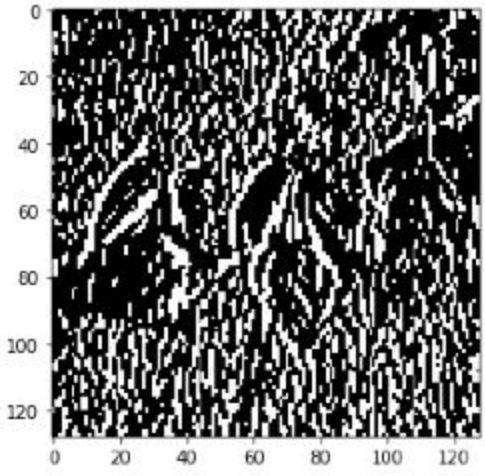
We first tried to use densenet as the perfect established model for classifying large classes of images. Due to the large size of data sets and not being able to use cpu for this project, and the lack of sufficient memory while using cuda, we could not work with deep nets. Therefore, we decided to extract features from the data set and use a multiclass classifier to classify our images based on the extracted feature. To do so, we relied on the histogram of oriented gradients (HOG) and texture features of the images. Before extracting these features, we first resized every image to have a 128 by 128 pixels and then made a grayscale of all images. The reason for this was the variety of image size and food colors of images in different and the same classes. Therefore, we realized that color is not a good feature to rely on. We extracted the hog feature of the resized gray-scale images using the hog built-in function in python. We also used the gabor built-in function to extract the texture of each image. Finally we made a vector of combinations of these features for each image. The gabor filter provided a vector of size 32768 for each image and the hog feature provided a vector of size 2916 for each image. Below we show the extracted texture and hog feature for the shown gray-scale image.



texture feature:

HOG feature:





5.The architecture of the final model

For our final model we used hog feature and Gabor filtering to extract the textures of images. We tuned the hyperparameters for gabor filters using 10-fold cross validation on the training set. Our final choice for the frequency of Gabor filter was 0.5.

We therefore used PCA with 500 components to only focus on the most pronounced parts of the texture feature for each image. For our final image classifier we used SVM with the rbf kernel. We used the SVC built-in function using python to use SVM as our final classifier. In order to tune the parameters of SVC we used the built-in grid search using python to choose the final classifier with parameters C = 1 and gamma = 0.1. Using this method, we could also control overfitting.

Finally we used the tuned model to train using both train and validation set to finally predict on the unseen test set.

The big challenges of this work was the large number of classes as well as the large number of samples. Since these are all food images, the similarity between different classes might exist. Therefore, one big challenge was to extract the appropriate feature. On the other hand, when training, due to the large size of the combination of train and validation set, the training time would take days to finish. Therefore, we only used a small batch of the combination of the train and validation set to train our model.

As for the densenet model, we failed so many times. The biggest challenge is that the training dataset is too large. If I set batch_size = 32, this will make the model run faster. However, my cuda doesn't have that much memory to support this operation. So I deduce the batch_size = 8 to make the cuda run.

10.Conclusion

Due to the limited device and large dataset, our densenet model doesn't perform well. So we use the hog feature and Gabor filtering combined with the SVM model. As illustrated above,however it takes too much time to run.

In this project, we try to accurately classify the food images.We apply three models: Logistic Regression, DenseNet and Hog feature and Gabor filtering combined SVM models. All models can accurately identify food images. However, all the models are limited to input data size.

So we haven't gotten sufficient results. We illustrate our thoughts and methods here, hope there is more time to run our algorithm.

11.References

[1] Regularizing Neural Networks by Penalizing Confident Output Distributions

[2] Trained Ternary Quantization

[3] Deep Networks with Stochastic Depth

[4] SmoothGrad: removing noise by adding noise

[5] XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks

[6] Snapshot Ensembles: Train 1, get M for free

[7] Automated flower classification over a large number of classes