Snack Classification

W281 Computer Vision
Aditya Shah, Kai Ding, Sudhir Suvva, Zukang Yang

Introduction

Image classification is an important field of computer vision with many applications. One interesting application of image classification is the classification of snacks based on their images. This application is invaluable, especially in retail and inventory management where automated systems can accurately identify snacks on shelves, keep track of inventory, and manage expiration dates.

Dataset

Description

We obtained the dataset named snacks from Hugging Face. This dataset is a subset of a larger dataset named Google Open Image Dataset released in 2017, further cleaned and processed by the author of the snack dataset.

The dataset has 6,745 images spanning a total of 20 different snacks, where a train-validation-test mask is applied, resulting in 4,838 in the train set, 955 in the validation set, and 952 in the test set. The distributions of snack categories in all three sets are balanced, thus eliminating the need for the handling of an unbalanced dataset. Moreover, each image in the dataset is resized so that its smallest side is 256 pixels.

Image Samples

The images for each snack category contain a lot of diversity. In other words, these images are not merely the snack sitting in the image taking up most of the pixel space, but rather, the snack might be in any shapes, conditions, quantities, and locations, with a variety of background noise. For example, the three images of apples appear visually different: the leftmost image is a single apple held by three human fingers; the middle image is a cluster of apples; and the rightmost image is an apple cut into slices resting on the cutting board. This diversity makes our dataset resemble real-life situations and is beneficial for us to build a robust snack classifier.

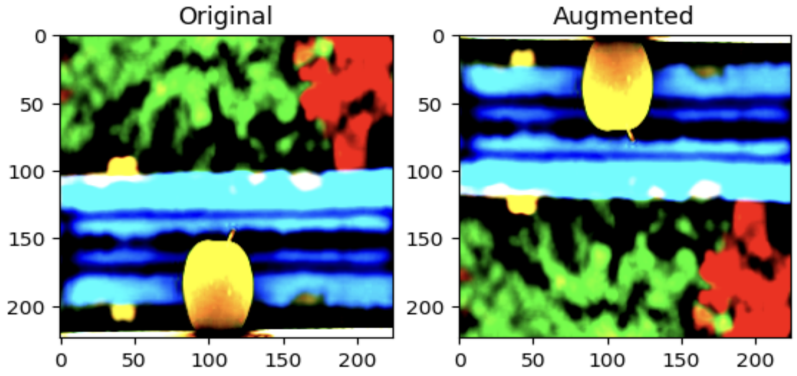
Data Processing

General Processing

We are benefiting from the fact that this dataset was processed neatly such that only minimal data processing is needed. First, the images were previously processed such that the smallest side is 256 pixels. However, as the largest sides across images differ, we further resize the images into 224 x 224 pixels. Second, since many advanced ML models benefit from input features within the same and small scale, we use the min-max scaling technique to normalize the pixel values into the range of 0 and 1.

Data augmentation

We explored 1 data augmentation technique, a vertical flip on every image. We hypothesized that the augmented data can train the model to recognize variances in shapes and positions, therefore becoming more robust during inference.



Feature Vectors

We hand-crafted two simple feature vectors, i.e. hue histogram and HOG, along with a complex feature embedding using ResNet-50. Our goal is to investigate whether the

simple feature vectors are strong indicators of the snack categories by comparing their performance against that of the popular ResNet-50 embedding model.

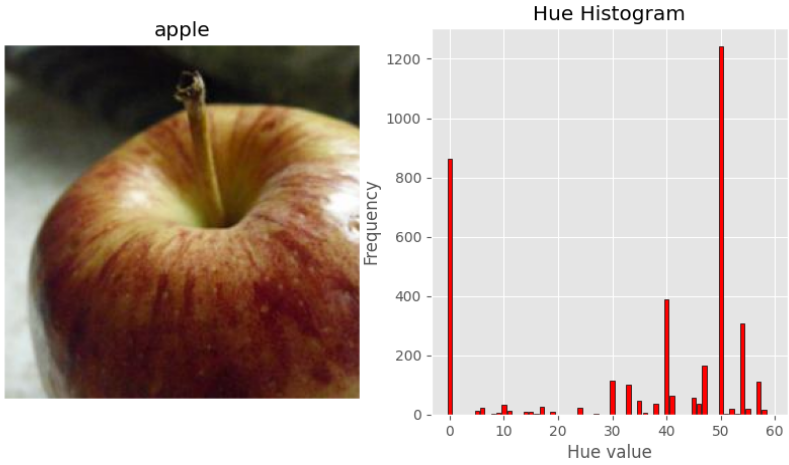
Additionally, we applied PCA, a common dimensionality technique, on each feature vector to reduce their feature space down to an acceptable range for modeling, to optimize computation power. Specifically, we selected the number of principal components for the PCA process such that >= 90% of the model variance be explained.

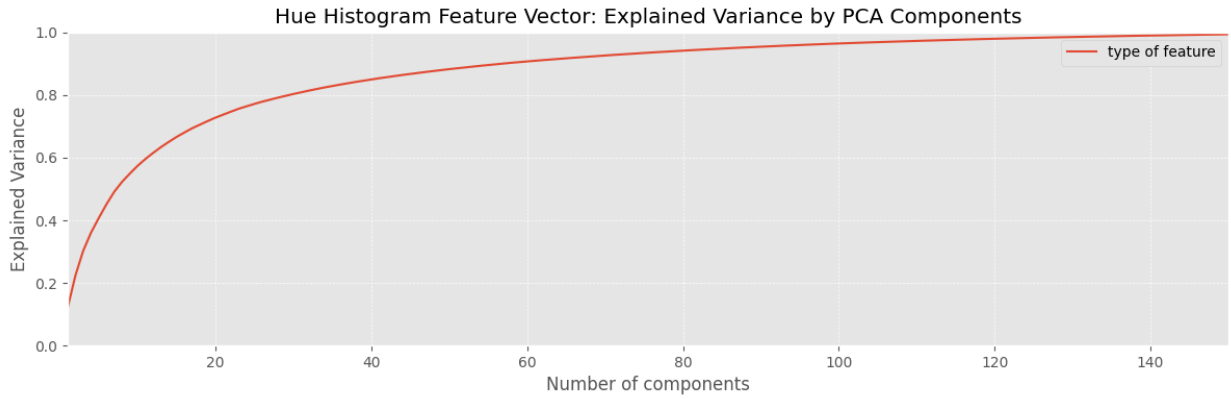
Lastly, we used the t-SNE algorithm to visualize the efficiency of each feature vector. Ideally, an efficient feature vector, in a 2D plot, should present a clear separation among different snack classes.

Hue Histogram

Intuitively, snacks tend to display relatively fixed color patterns unique to their kind. For instance, apples are mostly always red and bananas are mostly always yellow. Other snacks available in the dataset, such as watermelons, cookies, and hot dogs, all have their unique set of color patterns. Therefore, we engineered a color-based feature vector to capture these characteristics. Among all the choices, we chose the HSV histogram because it is intuitive and easy to implement.

HSV histogram works by first converting the RGB image into HSV where HSV respectively stands for hue, saturation, and lightness. Then, we extracted the hue channel which then was plotted into a histogram. The hue channel is expressed as a number from 0 to 360 degrees where a certain sub-range within the range represents a certain color. For example, red typically falls between 0 and 60 degrees. This way of representing color trends is more straightforward than the RGB approach as RGB usually consists of three color channels, leading to a higher difficulty in processing. Finally, we convert each hue histogram into a 1-D vector. Below is an example of the hue histogram with 60 bins for an image of an apple. From the histogram, we can discern that for the particular image, hue values concentrate on the 0-th and the 50th bin.





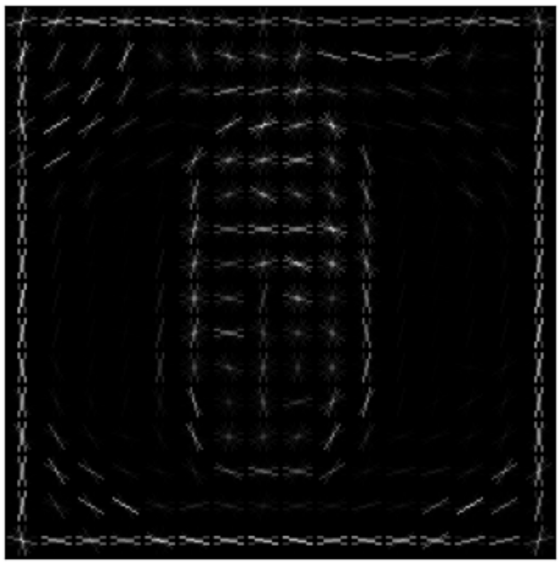
Besides, the plot of cumulative explained variance by PCA components for the hue histogram with 1,600 bins (obtained through a series of experiments) shows that the optimal number of components is 150 which could explain almost 100% of the data variance.

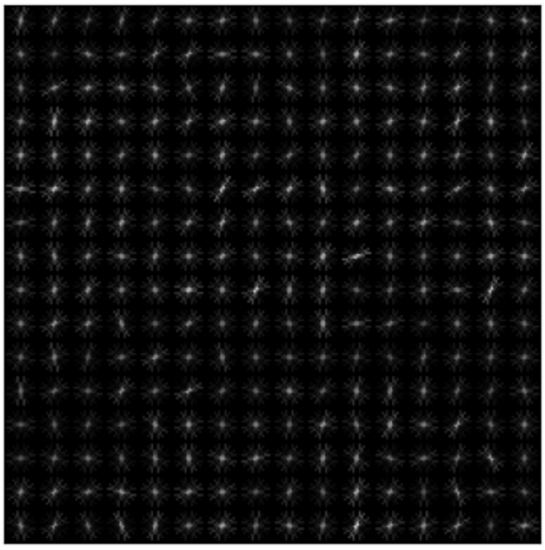
HOG

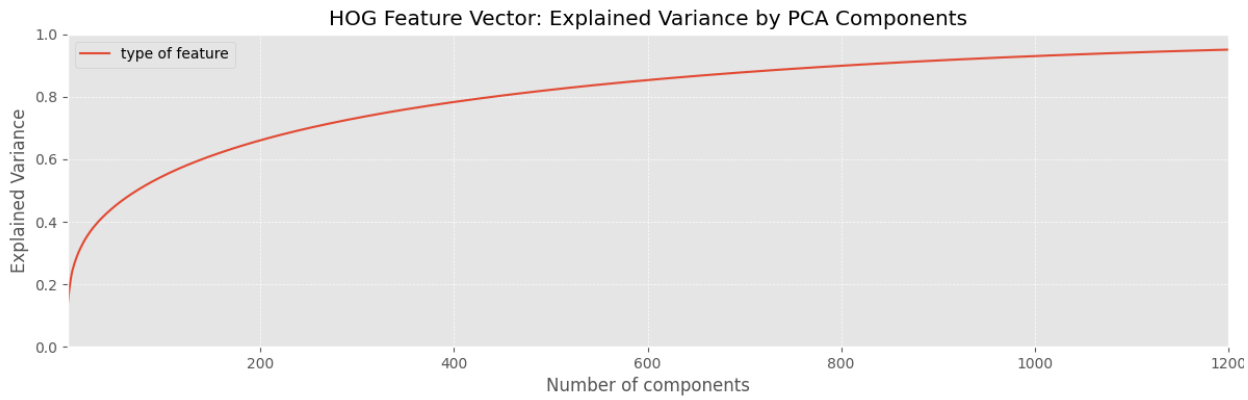
Aside from the intuition that color patterns can be an accurate indicator of the types of snacks, the shapes or edges can be another one. The comparison of shapes between apples and bananas is an example.

Among many edge detection techniques, we pick HOG (histogram of oriented gradients) as our edge-based feature vector. The main idea behind the HOG algorithm is to describe an image in terms of the distribution of both the direction and magnitude of the edges in the image. This benefit allows HOG to provide more information about the shapes in an image, compared to other edge detector algorithms, such as canny filter. However, below are two examples of HOG as a great indicator and as a bad one, respectively. The first image is a hot dog, and HOG was able to capture the rough shapes of it. Yet, the second HOG, which is the HOG of popcorn, did not display any

shape patterns. Therefore, we know that edge detection has limited capability as it can't capture the characteristics of some snacks. As HOG is a 2D representation of its original image, we flatten the output of the HOG algorithm into a 1D vector.



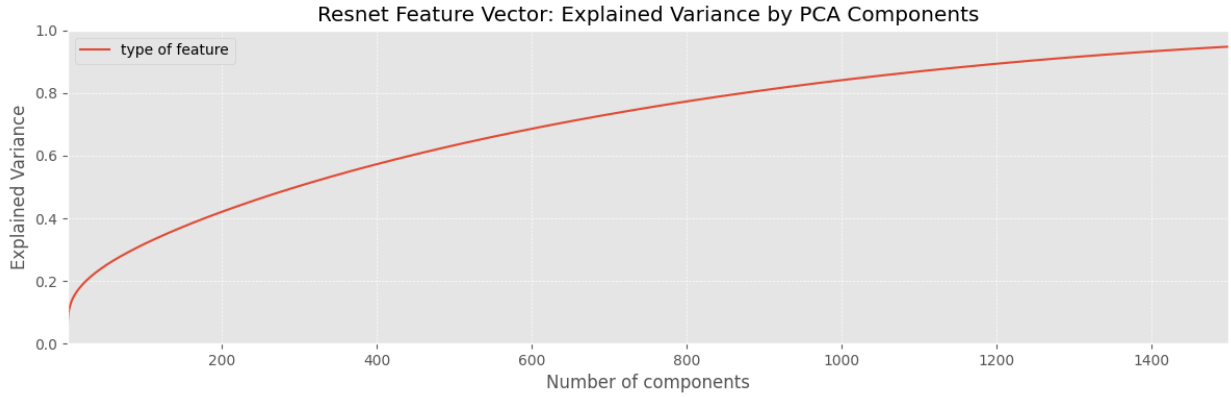




Besides, the plot of cumulative explained variance by PCA components for HOG above shows that the optimal number of components is 1200 which could explain ~95% of the data variance.

ResNet-50 PCA explained the variance plot

We used pre-trained ResNet50 to extract feature representations of the dataset from the average pool layer. The extracted embedding vector has a dimension of 2,048, and we decided to apply PCA for dimensionality reduction. The resulting explained variance plot clearly showed that the top 500 dimensions explain over 90% of the variance in the dataset, while the top 1,000 dimensions explain almost 100% of the variance. This represented a clear opportunity for faster, more efficient computations through PCA, and we decided to use only the top 1,000 dimensions of the feature vector.



Besides, the plot of cumulative explained variance by PCA components for ResNet embedding above shows that the optimal number of components is 1,500 which could explain ~95% of the data variance.

T-SNE Plot

We compared t-SNE plots among 4 feature vectors, including hue histogram, HOG, and ResNet50 embeddings, by projecting high-dimensional data onto 2 dimensions. Based on the plots, neither the hue histogram nor HOG was able to clearly distinguish the snack categories from each other. However, with the ResNet-50 embedding feature vector, different snack categories are already separated quite nicely. Here, we could already tell which feature vector is the most informative of the snack class.

Methodology

In this project, we aimed to develop a snack classifier to classify 20 different snacks. We will use two classifiers, i.e. logistic regression and support vector machine classifiers. Specifically, we will use the hue histogram feature vector, a combination of hue histogram and Hog, and a pre-trained ResNet-50 embedding feature vector separately on the two classifiers mentioned above. Therefore, there will be 6 comparisons. The reason why we use a combination of hue histogram and HOG, rather than using HOG on its own, is because the performance of solely using the HOG feature vector is poor. Further, as all three feature vectors are in very high-dimensional space ranging from 1600-dimension to 256^2-dimension, we applied PCA to reduce their dimensions yet preserve enough dimensions to be able to explain over 90% of the data variance.

The general workflow of the modeling stage is we first train a model with the training set and concurrently use stratified K-fold cross-validation for hyper-parameter tuning with the validation set. Then, we fetch the best classifier to evaluate its performance on the test set.

Eventually, we compared the performance among the 6 comparison models and selected the best model for efficiency and accuracy, respectively.

Evaluation Metric

The main evaluation metric we tested against is the accuracy score. The accuracy score measures the overall correctness of the model predictions. In a real-life setting, most datasets are unbalanced, making the accuracy score a poor indicator of success. However, our dataset is well-balanced, so the accuracy score is a reliable indicator of our model performance.

Furthermore, to correctly understand our models' behaviors, and gauge an idea of which snack categories our models struggled with, we used a confusion matrix. A confusion matrix is a table whose row represents the actual class while the column represents the predicted class. In a confusion matrix, the cell (i, j) represents the number of observations known to be in group i (actual class) but predicted to be in group j (predicted class). A confusion matrix provides a comprehensible visualization to understand the model behavior against its predictions.

Logistic regression

Logistic regression is the classification version of linear regression. It strives to find the best-fitted line to minimize the probabilistic difference between the predicted and actual classes. Logistic regression assumes that all features used form a linear relationship with the target class. This is a simple yet cheap algorithm for many classification tasks.

SVM

SVM, or support vector machine, is a better solution for high-dimensional classification tasks in which logistic regression does poorly. It tries to find the hyperplane that best separates different classes from each other. SVM is relatively more expensive than logistic regression in computational cost, but its improvement in prediction power can outweigh the cost.

Performance

As mentioned earlier, the accuracy score is our main indication of success. In this report, we provide the accuracy scores on the training set, validation set, and test set.

By comparing the accuracy score on the training set against the corresponding validation and test set, we can gauge an idea of how well the model will perform on the unseen images. From the provided table, at a high level, neither of the hand-crafted feature vectors did well as a snack classifier. This is no surprise as using color trends or

shape patterns could only tell part of the story about an image. For example, an apple and a strawberry might have a similar color, or a banana and a hot dog might have a similar shape.

On the other hand, while the combination of hue histogram and HOG with the SVM model produces a relatively high accuracy score (~0.73), the model did not generalize well on the test set. The best performance model using the hand-crafted feature vector is SVM on the hue histogram. However, an accuracy score of 0.21 on the test set failed to push the model into the production line.

However, the open-sourced pre-trained image embedding model, ResNet-50, produced a highly informative feature vector, resulting in an almost perfect accuracy score on the training set, and a decent score on the validation and test set (~0.83). As a result, we strongly consider the SVM as our best-performing model as it produced the highest accuracy scores on both the test and validation set (~0.83).

Feature Vectors + Model	**Accuracy (Train / Val / Test)**
HSV + SVM	0.4 / 0.37 / 0.21
HUE + logistics	0.21 / 0.195 / 0.197
(HUE + HOG) + SVM	0.73 / 0.176 / 0.175
(HUE + HOG) + logistics	0.457 / 0.414 / 0.193
Resnet + Logistics	1.0 / 0.815 / 0.821
Resnet + Logistics (best params)	0.988 / 0.834 / 0.837
Resnet + SVM	0.983 / 0.833 / 0.834
Resnet + SVM (best params)	0.960 / 0.830 / 0.825

More importantly, we created confusion matrices to understand what each of our models performed well in and struggled with.

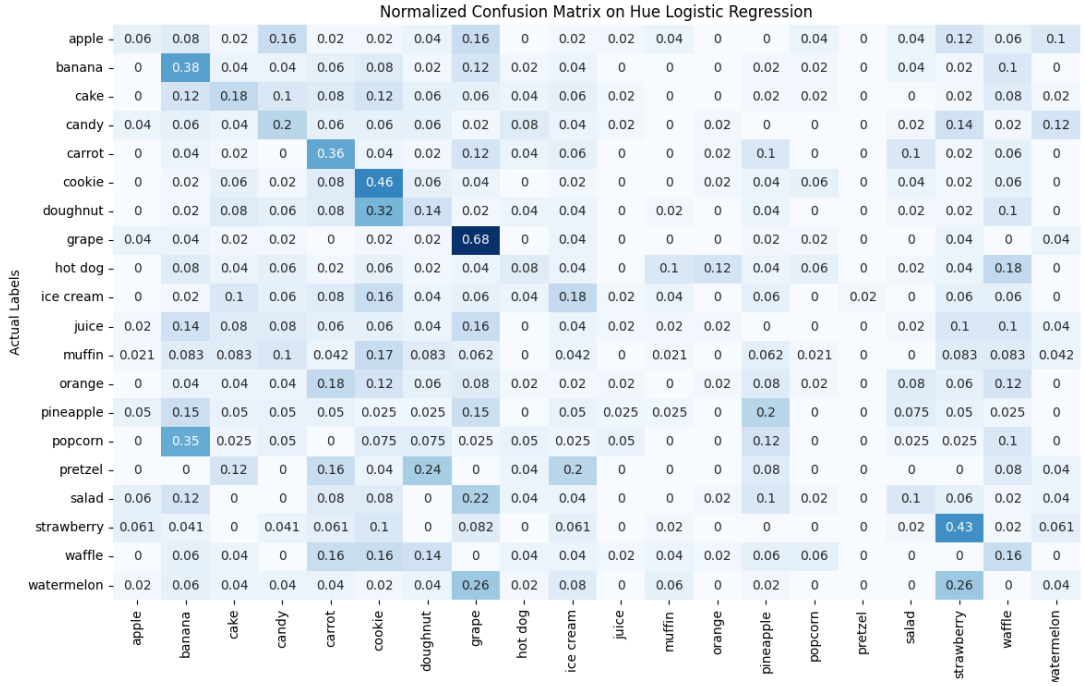
Confusion matrix:

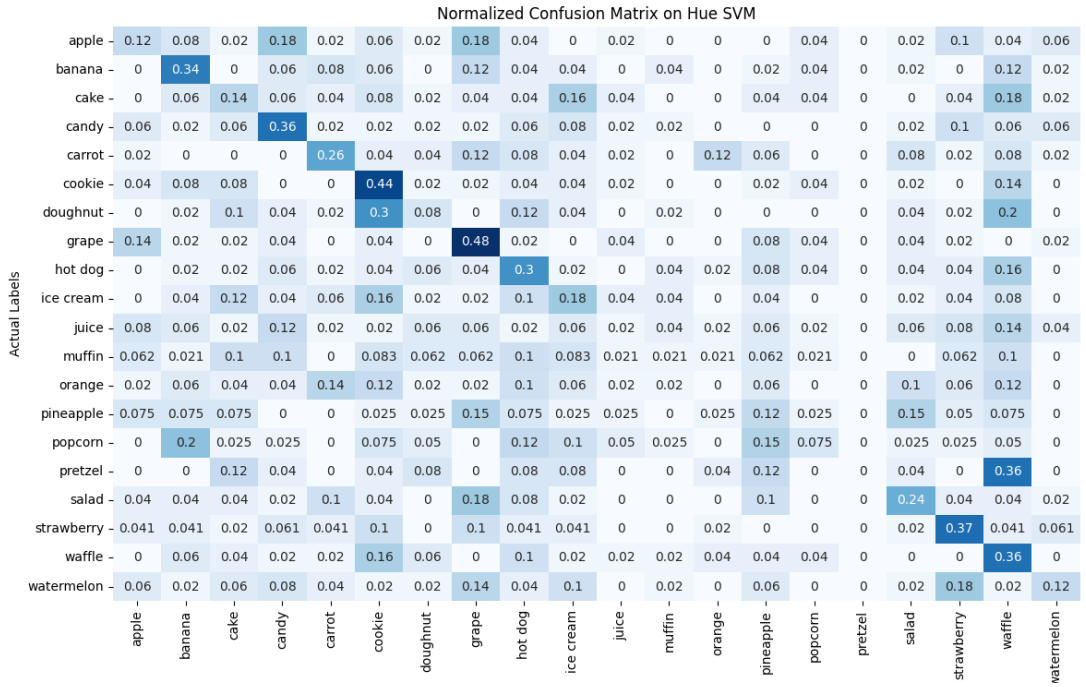
Here we show predictions for each of the 20 different categories, where the rows represent the ground truth labels and the columns represent the predicted class. The cell color is indicative of the proportion of the predictions that fall into the cell, with darker shades corresponding to a higher proportion.

A classifier that is performing well would, for each category, place most of the predictions for that class into the cells on the diagonal line that goes from the top left to the bottom right corner. Thus what we are looking for in the confusion matrix is for the cells on this diagonal line to be shaded with darker color, with all other cells being shaded with light gray.

Hue Histogram

Both confusion matrices show that our models struggled to distinguish among most of the snack categories. What the logistic regression did relatively well is the grape class. However, along the diagonal, we can see that it can't classify popcorn and pretzels, with negligible performance in most other classes. This feature failed because color trends alone can't give much information about what the snack is, especially since most images have multiple objects on top of the snack itself, displaying a variety of color patterns, and making different images of the same snack dissimilar.





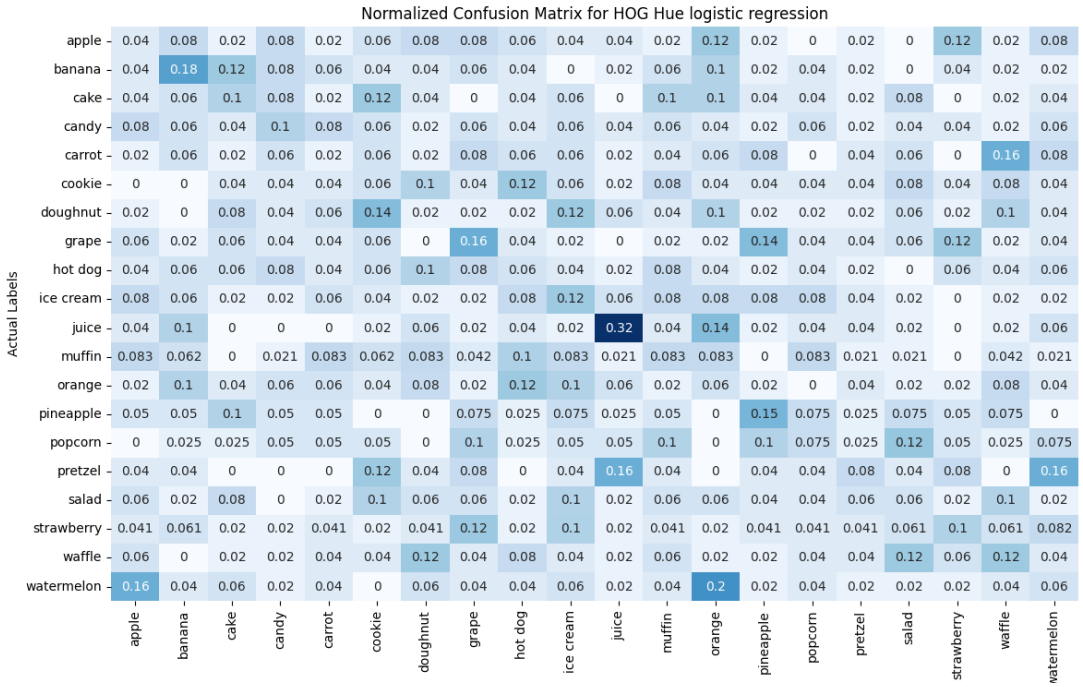
Hue + Hog:

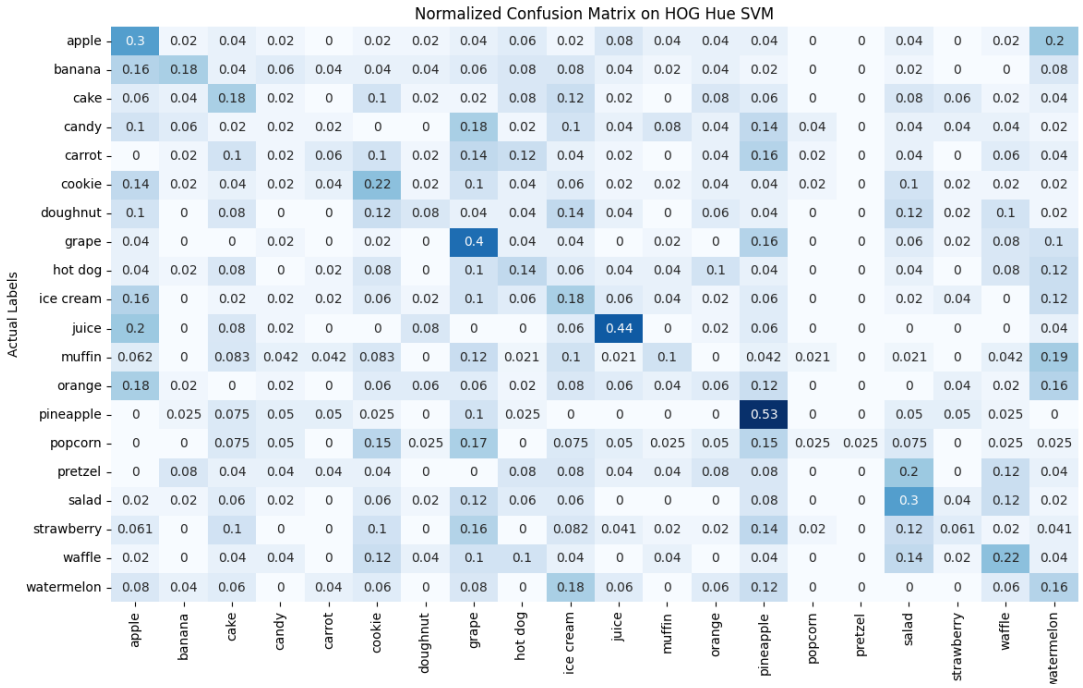
Looking at the confusion matrix for both Logistic Regression and SVM using Hue + Hog features, there is a lot of light shading on the diagonal.

Some of the highlights are:

1. Both LR and SVM models could not identify any particular category by more than 50%.
2. The only class that Logistic Regression could identify with the highest accuracy, about 30%, was Juice and a substantial percentage of remaining predictions exhibited false positives for Oranges.
3. SVM could identify a handful of classes though the prediction is 50% or less. The highest accuracy is for the pineapple.
4. Both the models fared worse with highly nuanced snacks like cookie, popcorn, and pretzel, etc while simple features like Hue and Hog is less effective in identifying with regards to computer vision.

What we are seeing is that both the models are having a hard time differentiating between the categories. There is huge variation in snack images within each class which makes it difficult for the classification algorithms to predict hand-crafted features like Hue and Hog. Arguably, SVM performed relatively well due to the algorithm's ability to categorize non-linear snack data into higher dimensions.



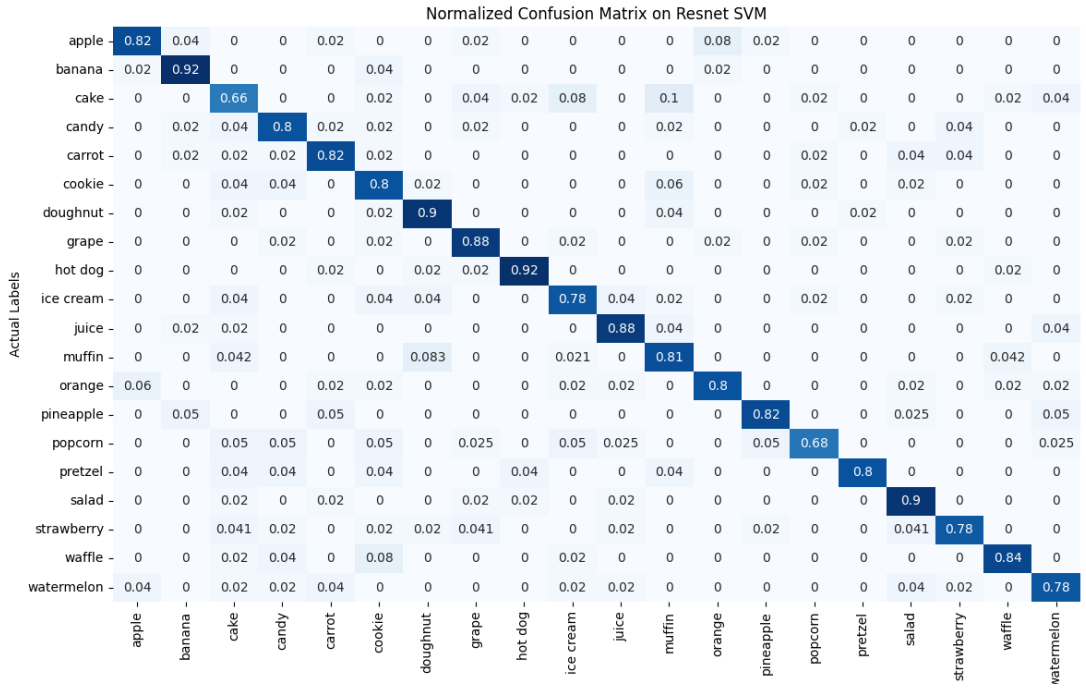


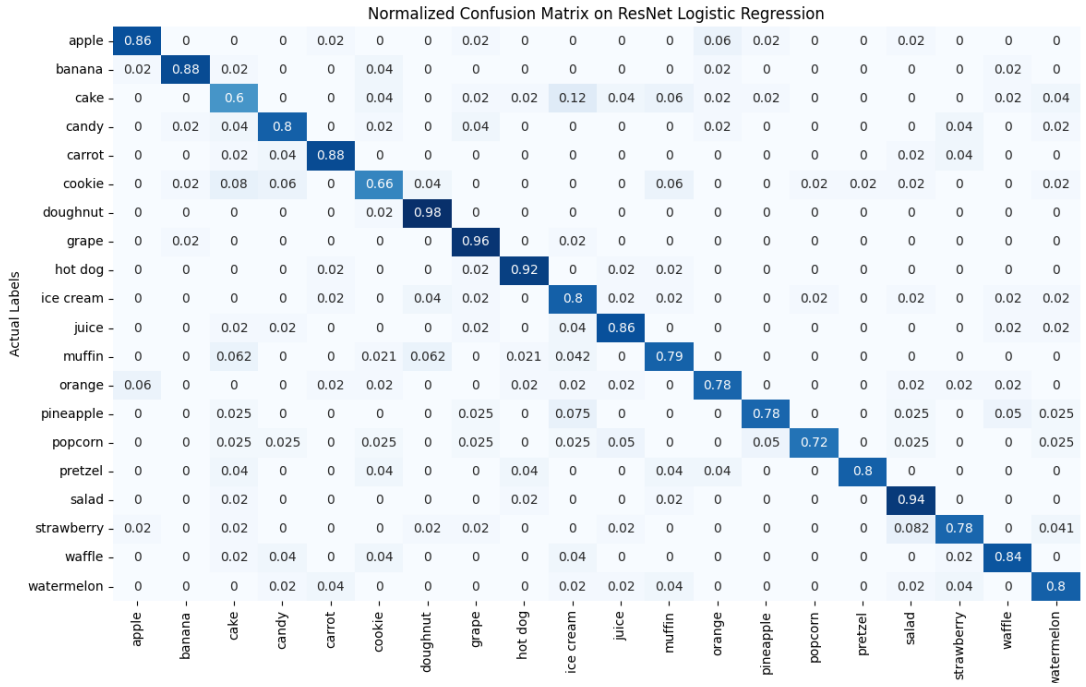
ResNet-50 Embedding:

In the confusion matrix for Resnet 50, the picture looks very good, with a good amount of dark shading on the diagonal, and lighter cells everywhere else. Some of the highlights are:

1. In Logistic Regression, 12% of the cake images are classified as Ice cream. Whereas in SVM, 8% of the cake images are classified as Ice cream.
2. Almost all the classes' accuracy is reasonably high.

Complex features extracted from Resnet 50 performed very well with the training data set with a decent prediction rate for all the classes. We think with a large data set we may see improvement in accuracy in the validation and test data as well.





Limitations on the Feature Vectors

While it makes intuitive sense to design feature vectors to capture the color and shape patterns of an image, in a real-life setting, an image of our use case not only has snacks in it but also other objects. Often, the snack object only occupies a very small portion of the image, leaving the remaining image with noise. This issue limits our feature vectors' capabilities. Observing the performance table, many models produced similar scores in the validation and test set, while the score in the training set was much higher. In an extreme case, the (Hue + HOG) + SVM model has an accuracy of 0.73 in the training set and less than 0.2 in the validation and test set. This overfitting situation demonstrates two possibilities: 1) the model captures some information about the snacks of our interests and a lot of the irrelevant noise, such that it does not generalize well to the validation and test set with different noise distribution; 2) the model captures important information about the snacks of our interests, but the images in the validation and test set contain images that are very different from those from which the model learned. The latter explanation suggests a larger dataset for modeling, while the former challenges us to reconsider the effectiveness of our feature vectors.

Similarly, though our ResNet-50 embedding model yielded high performance, the difference between the accuracy score on the training set and the test and validation set suggests some level of over-fitting. We explain that the validation and test sets both have images that look very different from the images in the training set.

Efficiency vs. Accuracy

When it comes to deployment, we should always discuss our models in two aspects, i.e. efficiency and accuracy. Generally speaking, an accurate model is complex such that it is typically less efficient. On the other hand, an efficient model can train and produce inferences promptly thanks to its simplicity, at the cost of reducing predicted power.

Moreover, due to the nature of our project, we not only want to develop the best model but also aim to identify the most informative feature vectors, we will also consider the runtime in generating each type of feature vector.

Efficiency

In our work, both logistic regression and SVM are computationally efficient in terms of training and generating inferences. Thus, we will only compare the runtime in generating the feature vectors. From the table, we observe that only the hue histogram has a fast generation speed. Moreover, due to its simplicity, the PCA runtime on the hue histogram is always efficient. On the other hand, both HOG and the ResNet embedding require a considerable amount of time to generate the feature vector, let alone the resulting high-dimensional vectors which slow down the PCA process. We conclude that for efficiency, the optimal model will be an SVM model on the hue histogram feature vector.

Accuracy

If accuracy plays a more important role in model selection, we need to pick a model based on its predictiveness. From this table, the best and most robust model for deployment is the logistic regression with ResNet-50 embeddings.

Future Work

Our experiment indicates that our simple hand-crafted feature vectors performed poorly on the dataset. This could be a result of multiple reasons. First, the size of the dataset is small relative to the number of snack categories the dataset has available. This could lead to underfitting of the models with the simple feature vectors. Besides, our analysis suggests that our feature vectors might capture the noise from each image during training more than the useful information, leading to severe overfitting to many of our models. Thus, we should explore other feature vectors that could overcome these challenges.

Moreover, we discovered that augmentation techniques add significantly to the computation of ResNet50 embeddings. Computing the embeddings of the original data with 1 augmented technique took >10 mins on Google Colab GPU. In the future scope, we can potentially explore more augmentation techniques including but not limited to horizontal flip, color jitter, etc.

Conclusion

In this project, we developed two different classification models on 2 hand-crafted feature vectors, along with a ResNet-50 embedding feature. From experiments, we discovered that our feature vectors are not as effective as expected in classifying snacks in an image. Thus, it remains a challenge for us to think more deeply about different ways of improvement in the future scope.

References

1. Snack dataset. https://huggingface.co/datasets/Matthijs/snacks
2. Deep Residual Learning for Image Recognition. https://arxiv.org/abs/1512.03385
3. TSNE plot. https://arxiv.org/abs/1512.03385

Additional Tables

```
apple
banana
cake
candy
carrot
cookie
doughnut
grape
hot dog
ice cream
juice
muffin
orange
pineapple
popcorn
pretzel
salad
strawberry
waffle
watermelon
```

Runtime	
Hue feature vec	3 seconds for 5k samples
Hog feature vec	Slow, 120 seconds for 5k samples
Resnet embedding	Slow, 150 seconds to run on M2 Macbook pro GPU; 2 times longer on CPU.
Data augmentation	Slow; 1 augmentation technique took >5mins on Google Colab V100 GPU
SVM training	Fast
Logistic regression training	Fast
Simple MLP (3 hidden layers) training	Fast on M2 Macbook Pro GPU