Intro to FoodX-251

Data Cleaning

Data Exploration

FoodX-251

FoodX-251 is an image classification dataset of 251 fine-grained food categories.

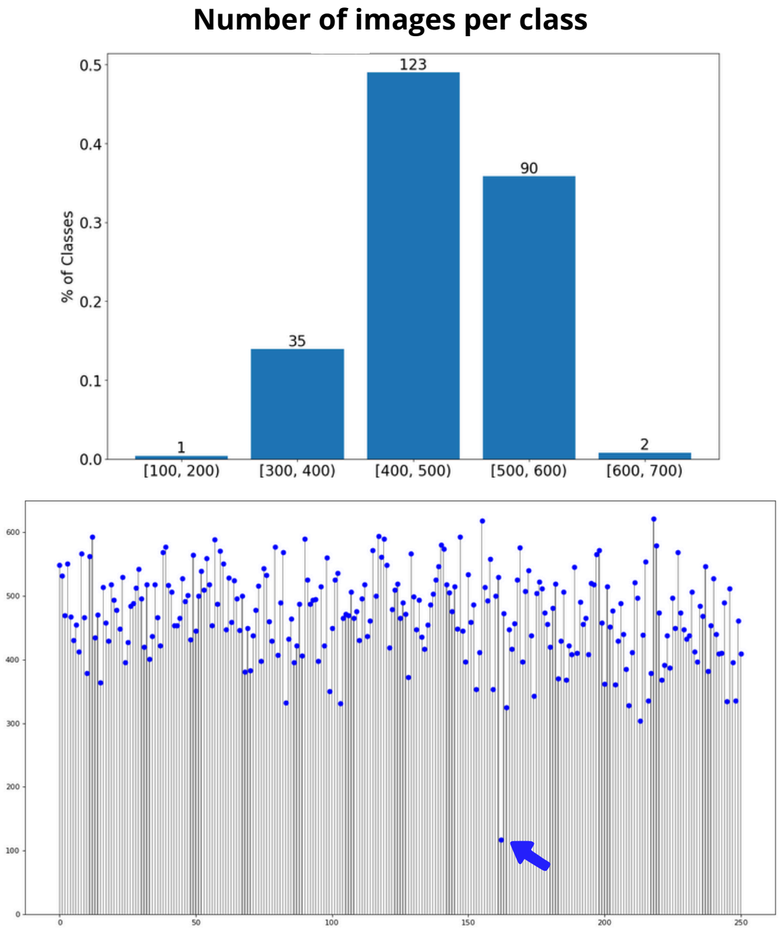
- **Training set** consists of around 120k images
- **Validation set** consists of around 12k images with human verified labels

The least and most populated classes contain 117 and 621 images respectively.

The food categories are **fine-grained**. General food categories, like “pasta” or “cake”, are not present. Instead we can find “marble cake”, “coffee cake”, or “coconut cake”.

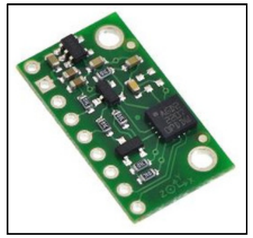
The food categories are **inherently similar** to each other because the label collection was done by extracting Food-101 sibiling categories from WordNet.

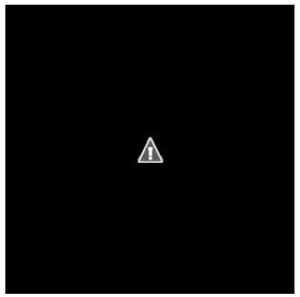
Images are collected via **web image search**, and the training data are not verified.

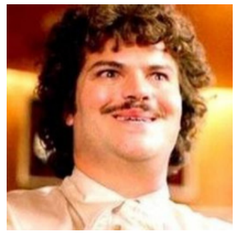


Anomaly 1: **Not Food**

People, cartoons, objects, landscapes, and more





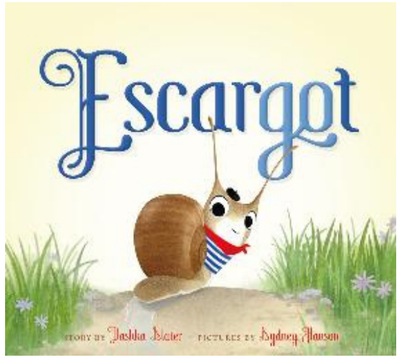


Anomaly 2: **Food Related**

Drawings and illustrations







Drawings of a food and
food related items

Clothes and costumes


Clothes with food shapes and colors

Recipes, menus, and posters

Names of food
or info about food

Anomaly 3: **Food is Not the Primary Element**

People with food

Food is not the principal object in the image

Packages

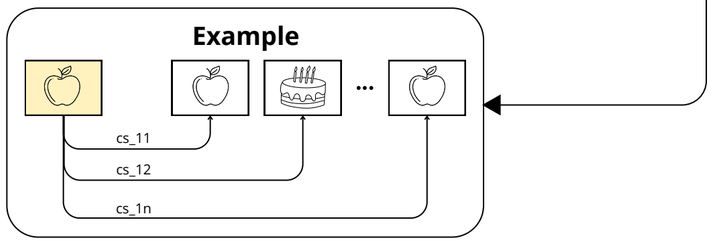
The image include a representation of food

Data Cleaning: Approach and Evaluation

Approach

- Analyze each class separately
- Obtain 1-dim vector representation of each image
  - Color Histogram
  - Bag of Visual Words
  - Features Extraction with Pretrained ViT

- Compare each vector with all the others, with cosine similarity



Avg Cosine Similarity
(cs_11 + cs_12 + ... + cs_1n) / n

- Average all the cosine similarities, so that each image has a score

- Remove the wrong images by setting a threshold on the score
- Evaluate the result with the 6 sample classes

Evaluation

To choose the best solution, 6 classes have been manually labeled.


omelette


seaweed salad


macaron


scotch egg


gnocchi


ramen

Color Histograms

Details

- Extracted histograms with 8, 16, 32, 64, 128, 256 bins

Pros

- Very fast: total exec time lower than 4 minutes
- Interpretable

Cons

- Background highly influences the histogram, as we can see from the least similar examples
- Not reliable in our case

Class **omelette** using 256 bins

Class **seaweed salad** using 256 bins

Bag of Visual Words

Details

- SIFT algorithm used to detect visual words
- K-means to construct vocabulary
- Tried with different vocabulary lengths: 10, 20, 40, 80

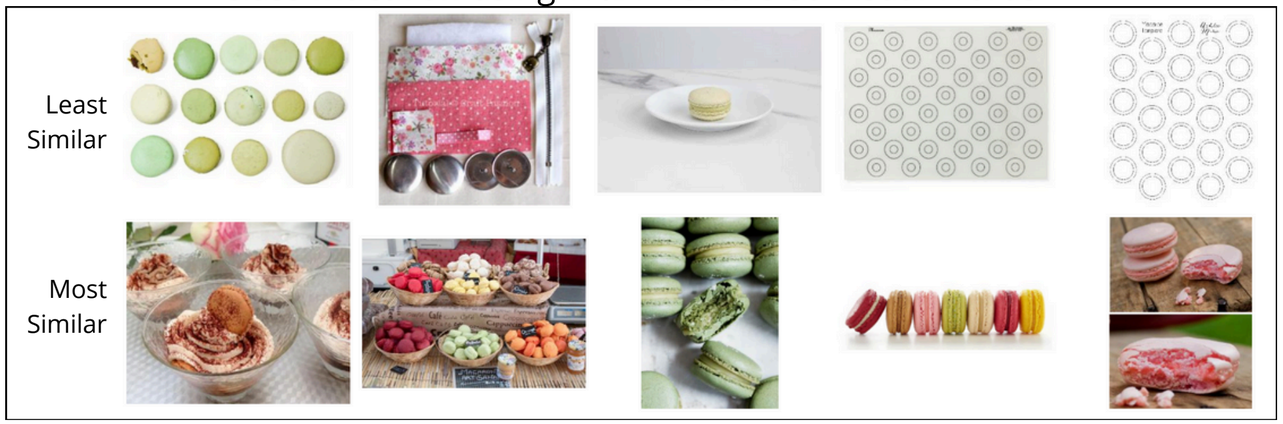
Pros

- SIFT scale-invariance
- Captures pattern in the images

Cons

- Slow: mean execution time around 3 hours
- Execution time increases as the length of vocabulary increases

Class **macaron** using TFIDF vectors 80 Visual Words



Class **scotch egg** using TFIDF vectors with 80 Visual Words

Features Extraction w/ Pretrained ViT

Details

- Model architecture presented in “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale” by Dosovitskiy et al.
- ViT trained on ImageNet-21k dataset
- ViT trained on Food-101 dataset

Pros

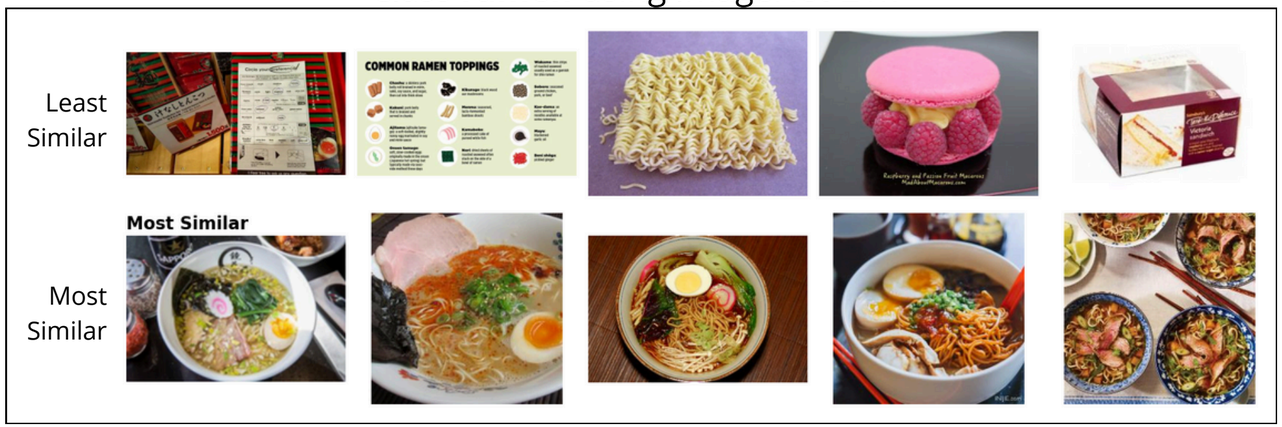
- Reliable in our case
- Captures key visual elements and their interactions

Cons

- Very Slow: execution time around 5 hours and a half
- Black box

Class **gnocchi** using ImageNet-21k

Class **ramen** using ImageNet-21k



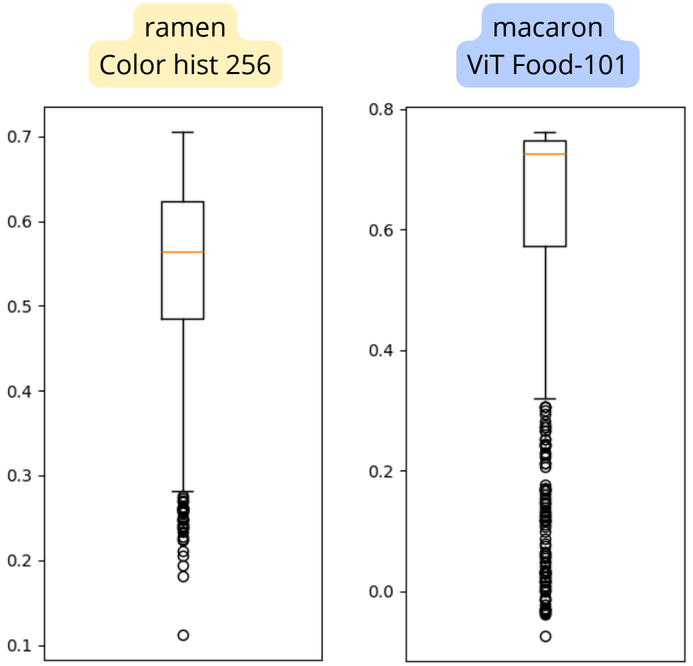
Evaluation: Results using Outlier Threshold

The first threshold evaluated is the distribution **outliers**, calculated by subtracting IQR x 1.5 to the first quartile.

How many images are considered outliers?

						
Color Hist 8	3.86%	2.75%	3.46%	3.87%	3.4%	3.51%
Color Hist 256	3.86%	4.67%	0%	5.81%	4.92%	7.03%
BoVW 40	5.6%	2.47%	2.19%	5.08%	4.42%	4.92%
BoVW 80	5.98%	3.3%	3.1%	5.08%	5.1%	4.92%
ViT ImageNet-21K	0%	0%	0%	0%	0%	0%
ViT Food-101	0%	0%	16.03%	0%	0%	0%

Examples of
Mean Cosine Similarity Distributions



Evaluation: Results using P25

The next threshold used is the **first quartile value**. Every image with a mean cosine similarity less or equal to P25 has been removed.

Precision: marked to remove, and actually to remove

						
CH 8	26.85%	25%	25.88%	30.17%	37.21%	35.76%
CH 256	28.7%	21.77%	28.24%	31.03%	34.88%	37.09%
BoVW 40	36.11%	35.48%	33.33%	25.86%	27.34%	35.1%
BoVW 80	33.33%	39.52%	33.33%	24.14%	28.91%	36.42%
ViT IN	86.11%	59.68%	81.18%	78.45%	59.69%	64.24%
ViT F	84.26%	72.58%	82.35%	79.31%	83.72%	69.54%

FPR: marked to remove, but actually to keep

						
CH 8	24.63%	25%	25%	23.23%	21.74%	19.2%
CH 256	24.15%	26.67%	24.57%	22.9%	22.39%	18.48%
BoVW 40	22.2%	19.58%	23.49%	24.92%	24.35%	19.57%
BoVW 80	22.93%	17.5%	23.49%	25.59%	23.91%	18.84%
ViT IN	9.02%	7.08%	14.87%	4.38%	15.43%	3.62%
ViT F	9.51%	0.42%	14.66%	4.04%	8.7%	0.72%

Evaluation: Results using P25

The next threshold used is the **first quartile value**. Every image with a mean cosine similarity less or equal to P25 has been removed.

Precision: marked to remove, and actually to remove

						
CH 8	26.85%	25%	25.88%	30.17%	37.21%	35.76%
CH 256	28.7%	21.77%	28.24%	31.03%	34.88%	37.09%
BoVW 40	36.11%	35.48%	33.33%	25.86%	27.34%	35.1%
BoVW 80	33.33%	**39.52%**	33.33%	24.14%	28.91%	36.42%
ViT IN	86.11%	59.68%	81.18%	78.45%	59.69%	64.24%
ViT F	84.26%	72.58%	82.35%	79.31%	83.72%	69.54%

FPR: marked to remove, but actually to keep

						
CH 8	24.63%	25%	25%	23.23%	21.74%	19.2%
CH 256	24.15%	26.67%	24.57%	22.9%	22.39%	18.48%
BoVW 40	22.2%	19.58%	23.49%	24.92%	24.35%	19.57%
BoVW 80	22.93%	**17.5%**	23.49%	25.59%	23.91%	18.84%
ViT IN	9.02%	7.08%	14.87%	4.38%	15.43%	3.62%
ViT F	9.51%	0.42%	14.66%	4.04%	8.7%	0.72%

ViT Evaluation: Results using P30, P35, P40, P45, P50

Application of other thresholds to the ViT similarity distribution. As the threshold increases, both precision and FPR increase.

ImageNet-21K results averaged over the sample classes

Threshold	Avg Precision	Avg FPR
P25	71.56%	9.07%
P30	80.63%	12.31%
P35	87.16%	16.51%
P40	92.01%	21.37%
P45	94.5%	27.01%
P50	96.44%	33.17%

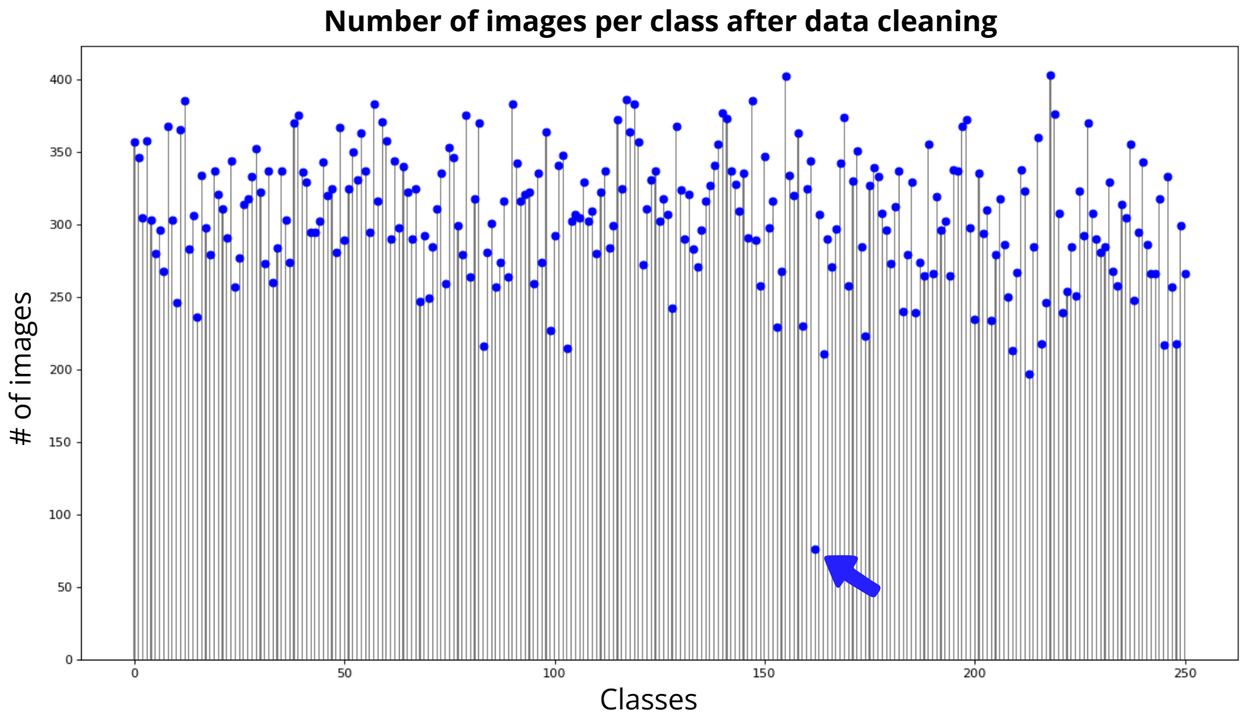
Food-101 results averaged over the sample classes

Threshold	Avg Precision	Avg FPR
P25	78.63%	6.34%
P30	86.57%	9.84%
P35	91.9%	14.43%
P40	96.11%	19.53%
P45	97.25%	25.9%
P50	97.96%	32.51%

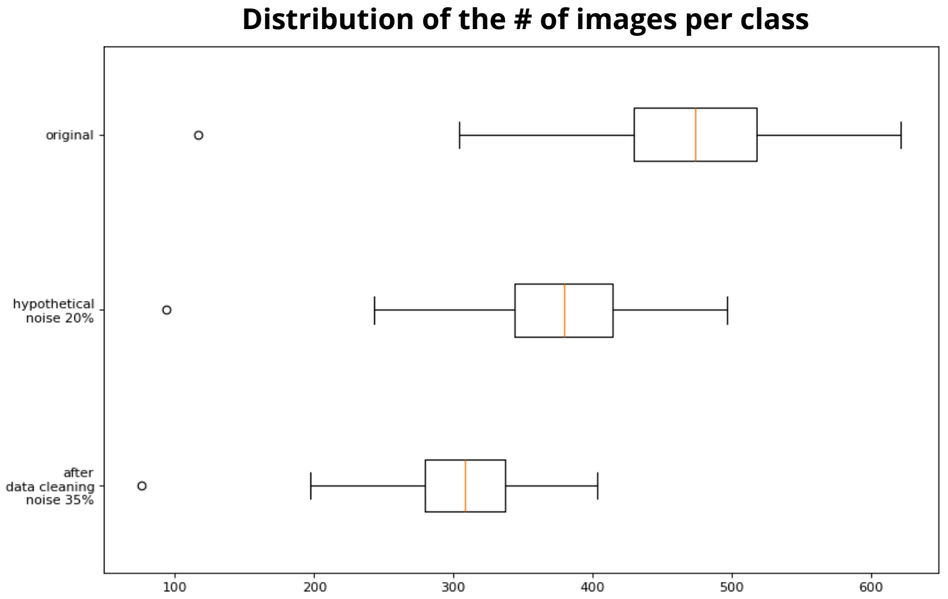
Data Cleaning: Threshold

Threshold P35

The threshold finally used is the **P35**. Every image with a average cosine similarity less or equal to P35 has been removed.



Before and After

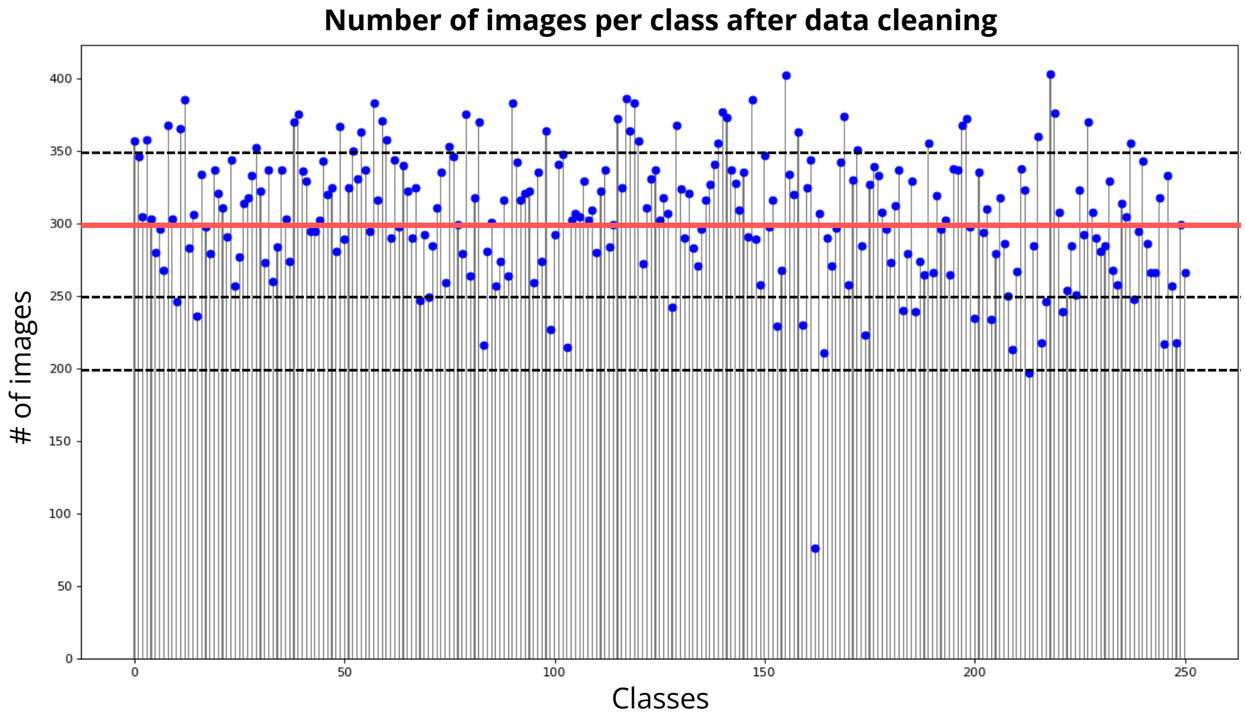


The number of images in class 162 (marble cake) is always an outlier. After data cleaning, the class contains 76 images.

Data Cleaning: Balancing

Balancing

In imbalanced datasets, classifiers tend to be overwhelmed by the large classes and ignore the small ones.



Cut off

Cut off (imgs per class)	**Total # of Images Removed**	**Total # of Images Added**
350	786	11 671
300	5 290	3 625
250	14 874	659
200	26 892	127

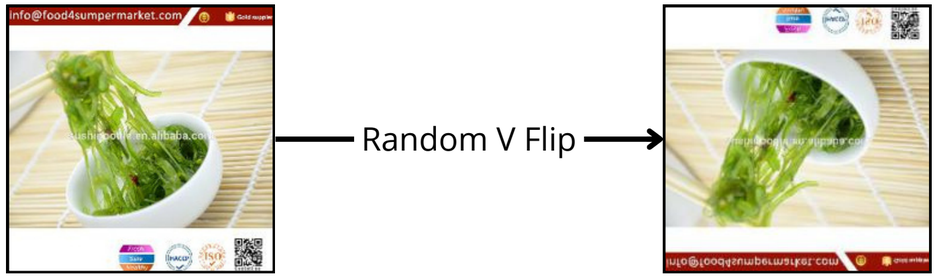
The chosen cut off is at 300:

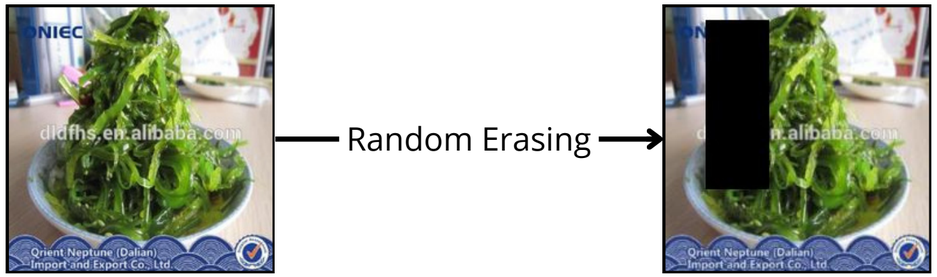
- from 143 classes we removed on avg 37 images
- in 108 classes we added on avg 34 images

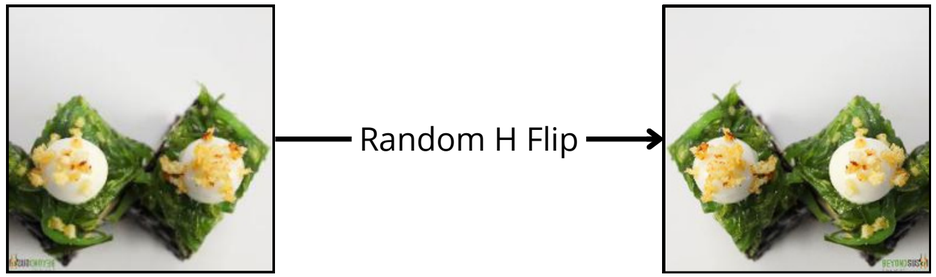
Data Cleaning: augmenting undersampled classes

Techniques used to augment undersampled classes:

- Random Vertical and Horizontal Flipping
- Random Erasing
- Random Rotation (45 degrees)
- Random Affine (15 degrees, 0.1 translate, [0.9, 1.1] scale)







Model

Results on Validation Set

Results on Degraded Validation Set

Classification

Model: MobileNetV3 Small

Why?

Given the computational capacity available, and the fact that we already had a considerable amount of data, the choice fell on a **small model** with reasonably high accuracy.

From scratch or fine-tuned?

In the first attempts, the model was initialized with random weights. But after considering the results, **the model was initialized using ImageNet-1K weights.**

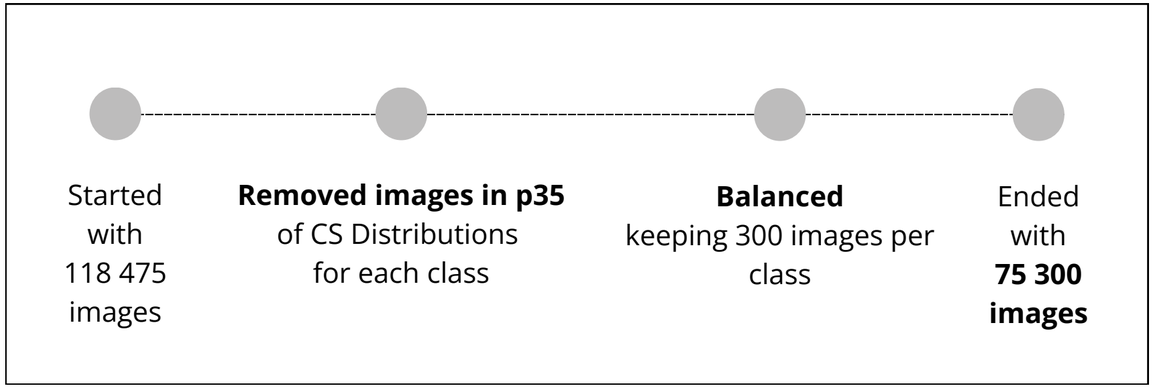
During the training phase, the model layers are not frozen.

MobileNetV3 Small Architecture

Input	Operator	exp size	*#out*	SE	NL	*s*
$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	✓	RE	2
$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	✓	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	120	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	144	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	288	96	✓	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	✓	HS	1
$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1024	-	HS	1
$1^2 \times 1024$	conv2d 1x1, NBN	-	k	-	-	1

Tuning Phase: Data and Hyperparameters

Training Data



Data transformation and augmentation levels in batch:

“none”

Scaled into 0-1
Resized 224x224
Normalization (IN)

“light”

“none”
H and V Flip
Erasing
Rotation
Affine

“heavy”

“light”
Solarize
Invert
Posterize

Hyperparameters

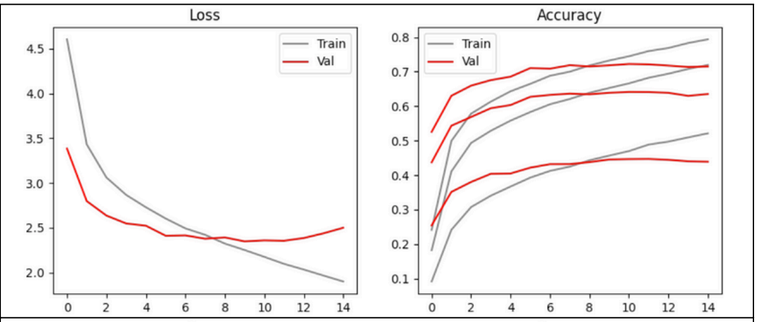
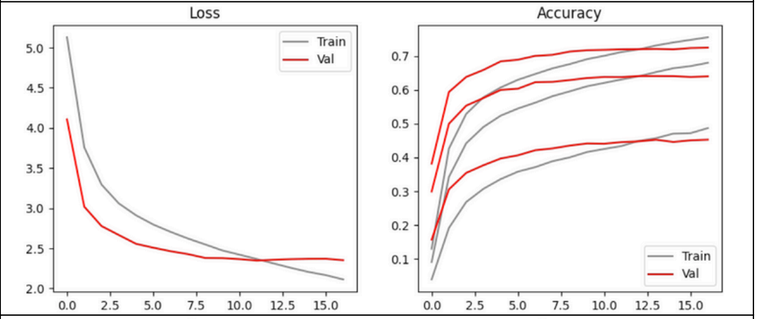
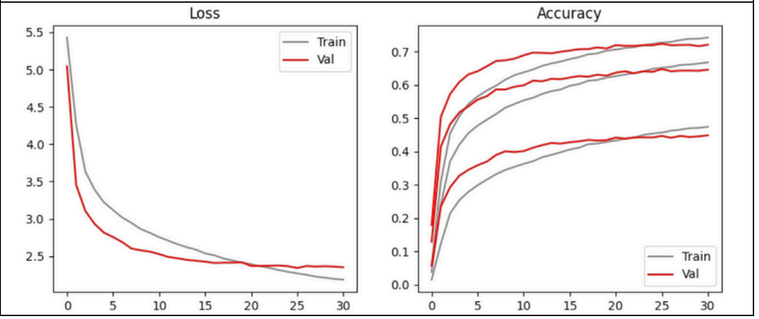
- Batch Size 128
- Loss Cross Entropy Loss
- Epochs 50
- Optimizer Adam
- Early Stopping Patience 5 on Val Loss
- LR Scheduler Cosine Annealing

Tuning

- Data augmentation level
- Weight Decay of Adam optimizer
- Start LR
- End LR

Tuning Phase

In the tuning phase, only 60% of the training and validation set were used.
The best model was then trained on all the data.

Model Configuration	Stopped at Epoch	Val Acc@1	Val Acc@3	Val Acc@5	Loss and Accuracy during Training
02_mnv3_fine_tuned	15	43.88%	63.21%	71.38%	
03_mnv3_fine_tuned	17	44.38%	63.89%	72.22%	
04_mnv3_fine_tuned	31	**44.64%**	64.60%	72.15%	

Results: Best Model

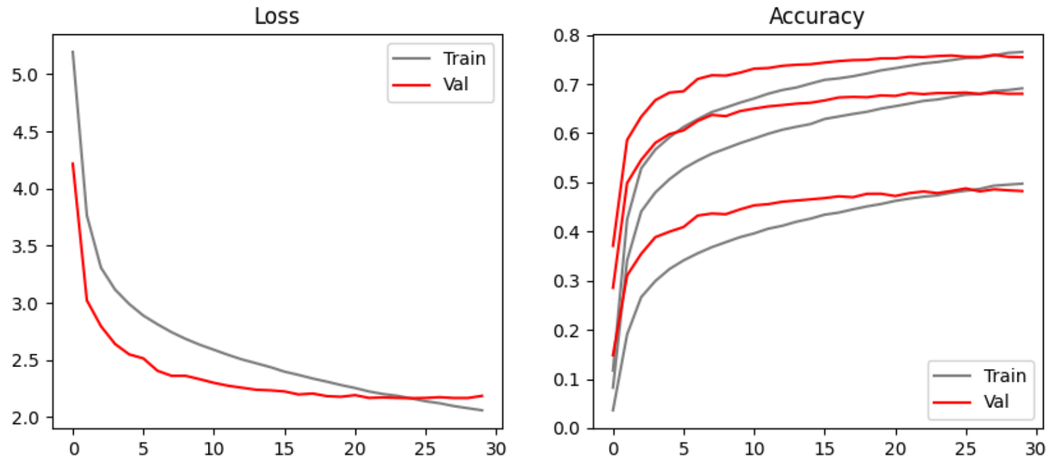
The best model is trained on all the data, and then evaluated on validation set and degraded validation set.

Configuration

- Model — mobilenet_v3_small
- Weights — ImageNet-1K
- File name — full_04_mnv3_fine_tuned

- Loss — Cross Entropy Loss
- Optimizer — Adam, Weight Decay 1e-3
- LR Scheduler — Cosine Annealing
- LR — Start at 5e-5, end at 1e-6
- Batch Size — 128
- Epochs — 50
- Early Stopping — Patience 5 on Val Loss
- Data Augmentation — “heavy”

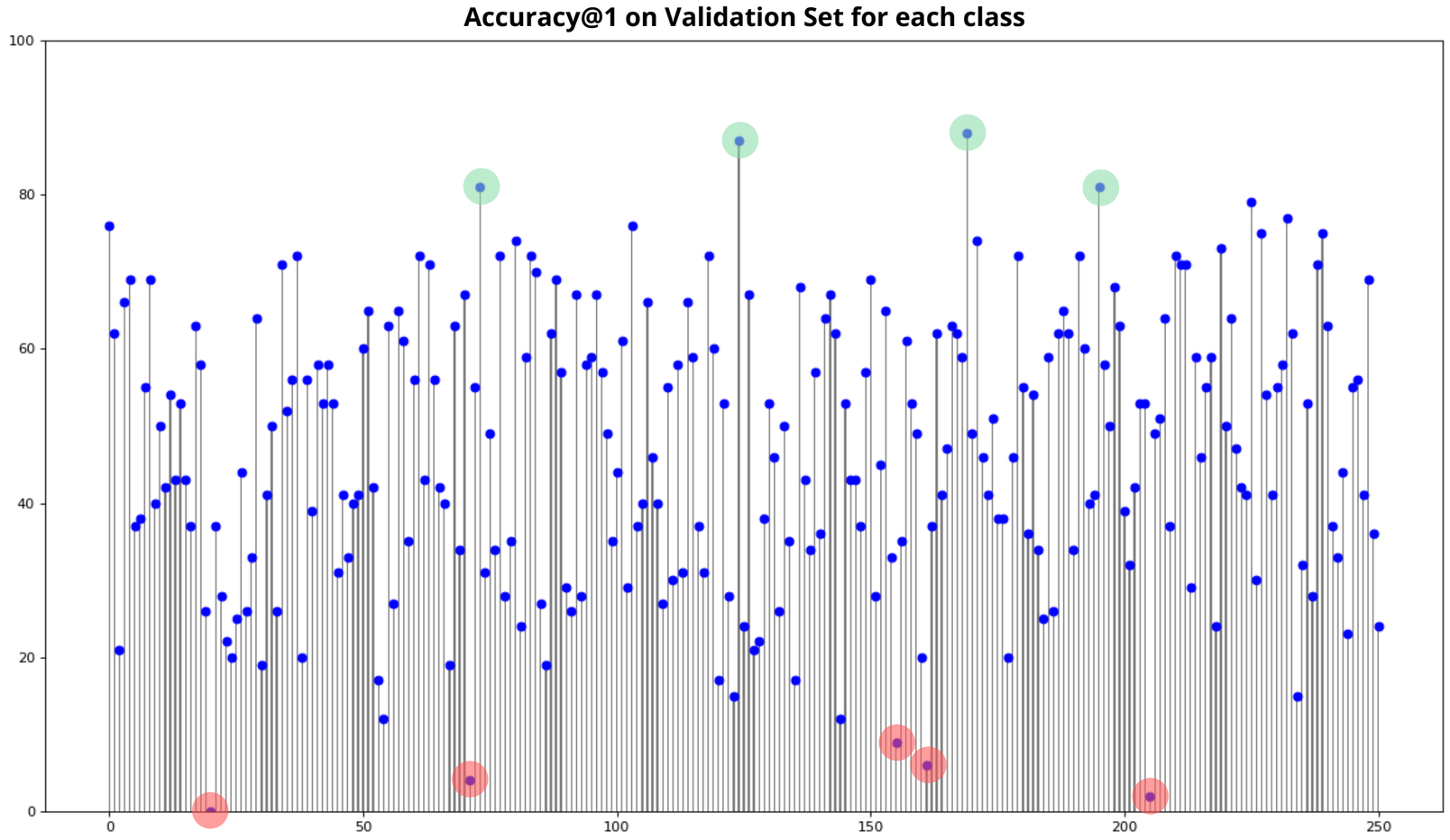
Loss and Accuracy during Training



Results

	Acc@1	Acc@3	Acc@5
Validation	48.77%	68.26%	75.95%
Degraded Validation	33.28%	48.95%	55.33%

Results: Zoom on Single Classes



Results

Classes **lowest** acc@1:

- 20 foie gras
- 205 lobster roll sandwich
- 71 sukiyaki
- 161 coconut cake
- 155 churro

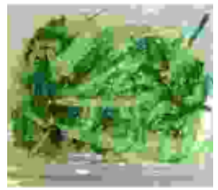
Classes **highest** acc@1:

- 169 bread pudding
- 124 boiled egg
- 195 oyster
- 73 baklava

Degraded Validation Set

Examples

JPEG Compression



Blurring

Gaussian Noise

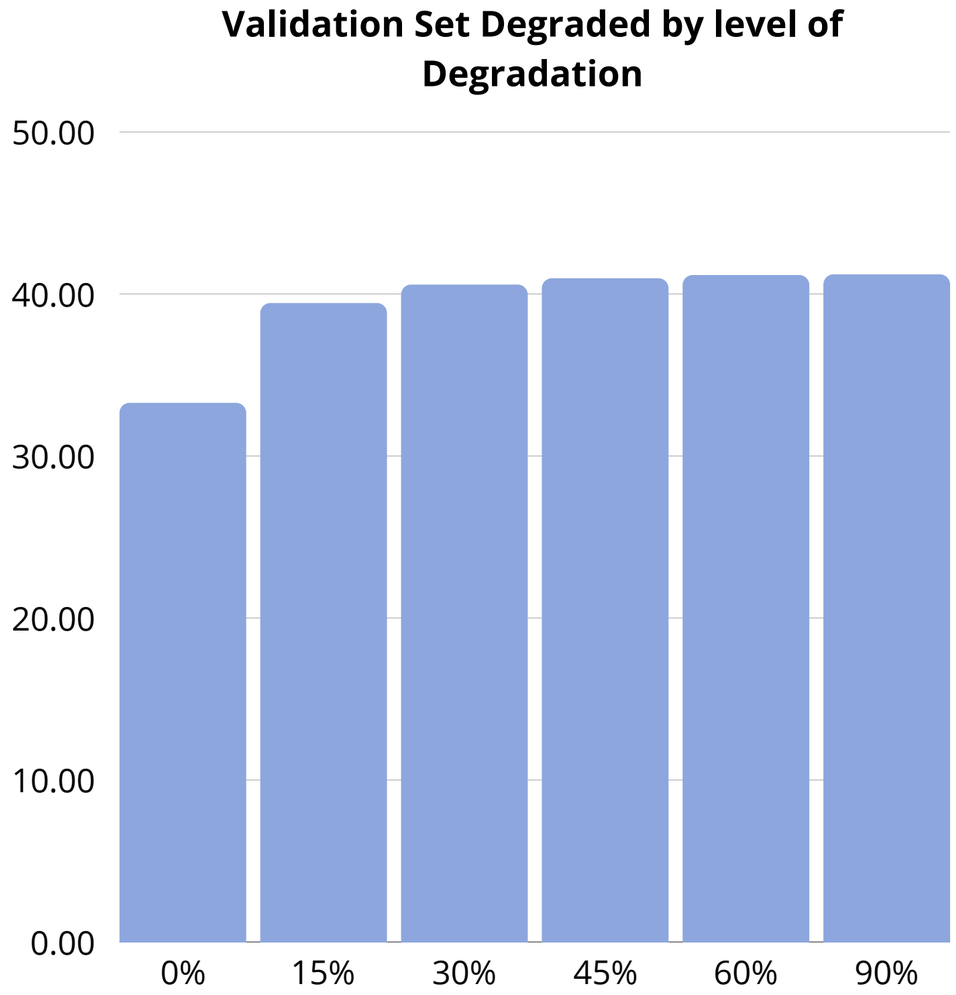
Approach

To improve the model performance on the degraded validation set, the degradation techniques have been applied to the training set.

- **15%** for each class
  (5% JPEG Compression, 5% Blurring, 5% Gaussian Noise)
- **30%** for each class
  (10% JPEG Compression, 10% Blurring, 10% Gaussian Noise)
- **45%** for each class
  (15% JPEG Compression, 15% Blurring, 15% Gaussian Noise)
- **60%** for each class
  (20% JPEG Compression, 20% Blurring, 20% Gaussian Noise)
- **90%** for each class
  (30% JPEG Compression, 30% Blurring, 30% Gaussian Noise)

Results: Degraded Validation Set

Degradation %	Stopped at E	Deg Acc@1	Deg Acc@3	Deg Acc@5
15	12	39.44%	57.29%	64.98%
30	12	40.58%	58.50%	66.40%
45	11	40.97%	59.28%	67.16%
60	9	41.17%	59.44%	67.73%
90	11	**41.21%**	59.74%	67.65%



Appendix

Cosine Similarity

Cosine similarity measures the similarity between two one-dim vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. Perfectly similar vectors have a cosine similarity of 1, and perfectly dissimilar vectors have a cosine similartiy of -1.

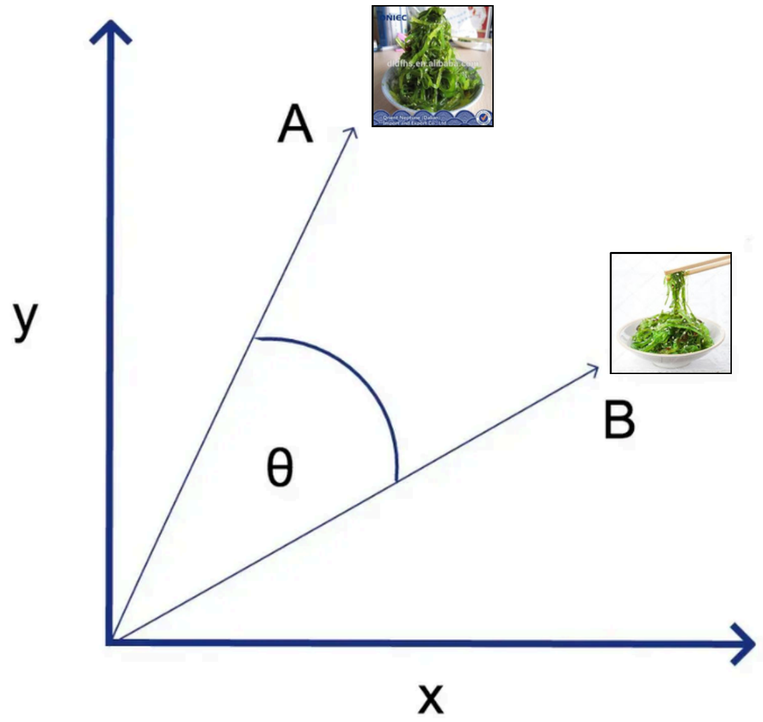
Use cases:

- semantic relationships between different texts
- recommendation systems
- measure the likeness between different images
- automate document classification processes
- finding clusters in high-dimensional data spaces

$$cos(\theta) = \frac{A \cdot B}{\|A\| \times \|B\|}$$

$$\sum_{i=1}^{n} A_i B_i$$

$$\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}$$

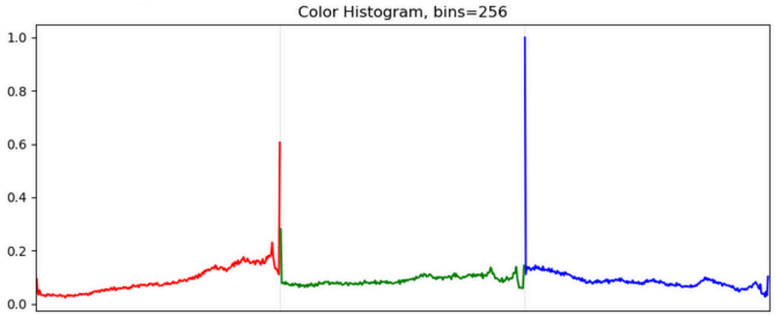


Color Histograms: Example 1

omelette

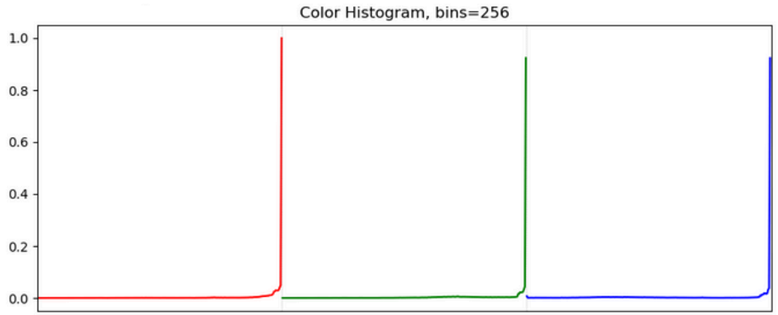
Most Similar Image

train_099739.jpg

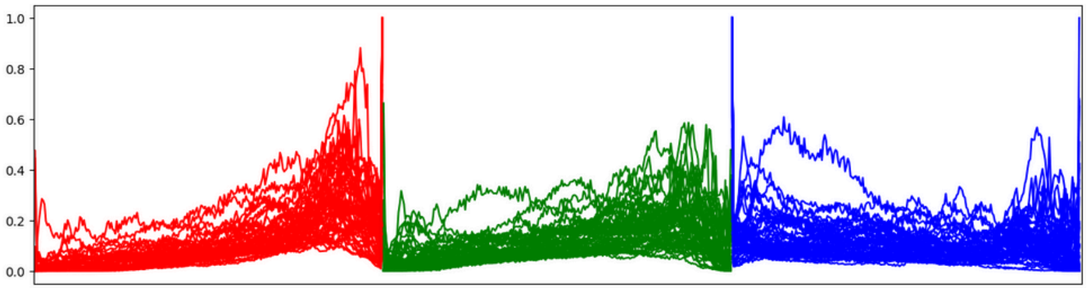


Least Similar Image

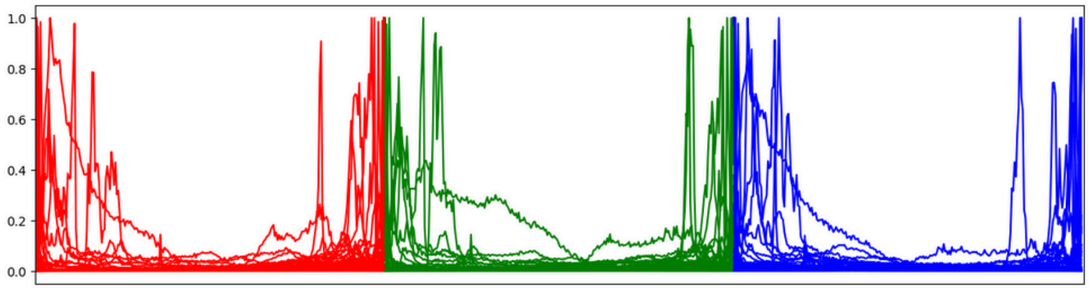
train_014276.jpg



50 Most Similar Images



50 Least Similar Images

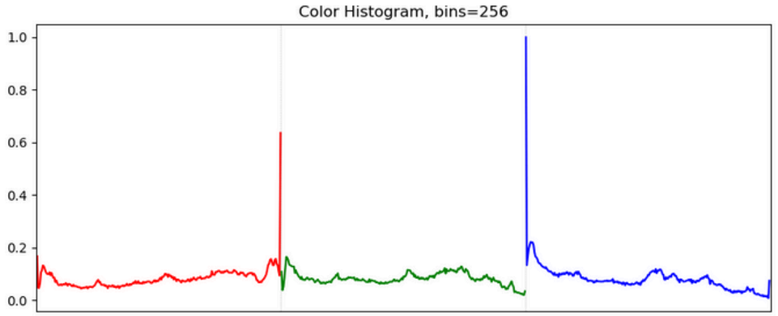


Color Histograms: Example 2

seaweed salad

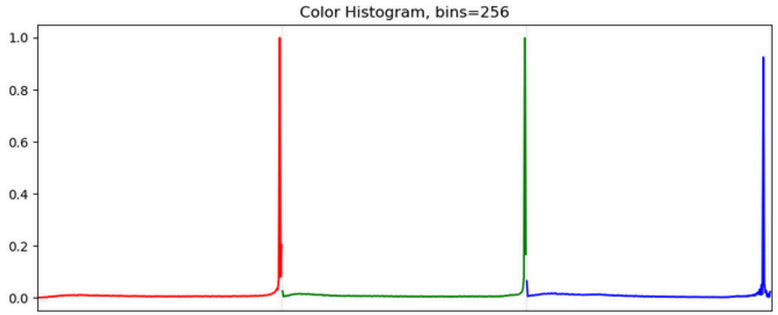
Most Similar Image

train_069532.jpg

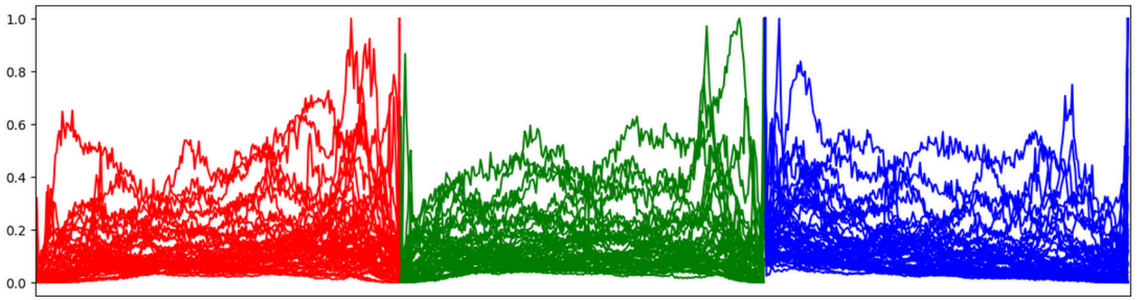


Least Similar Image

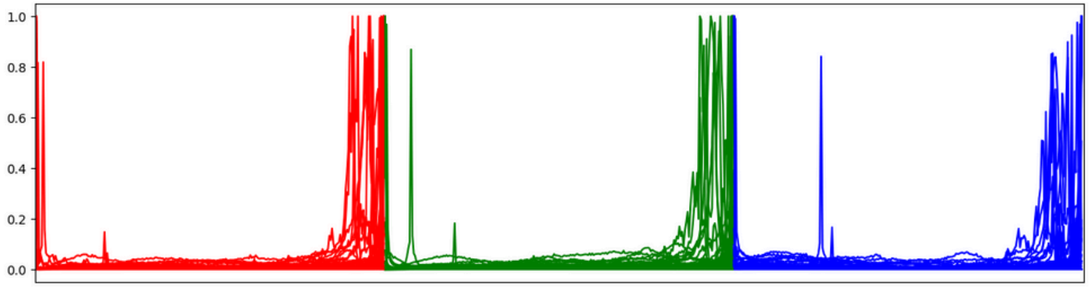
train_069623.jpg



50 Most Similar Images



50 Least Similar Images

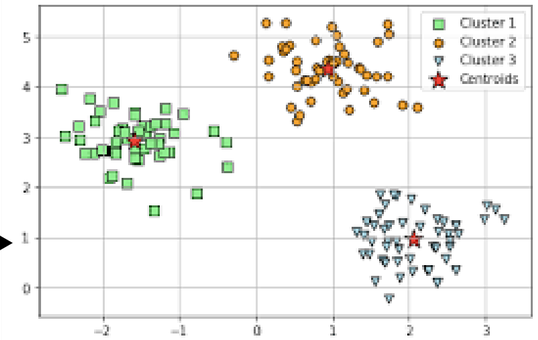


Bag of Visual Words: Steps

Images from the same class

Get all descriptors using **SIFT**

k-means



k is the number of visual words to keep, and also the number of clusters for k-means

if too small, visual words could be unrepresentative of all image regions, and if too large, there could be too many visual words with little to no of them being shared between images

Create vocabulary of length **k**

Each element is a visual word

Map each descriptor to visual word

Each images is now the vector of visual words it contains

TF Vector: how many times visual word *i* is in image?

TF-IDF Vector: account for word relevance

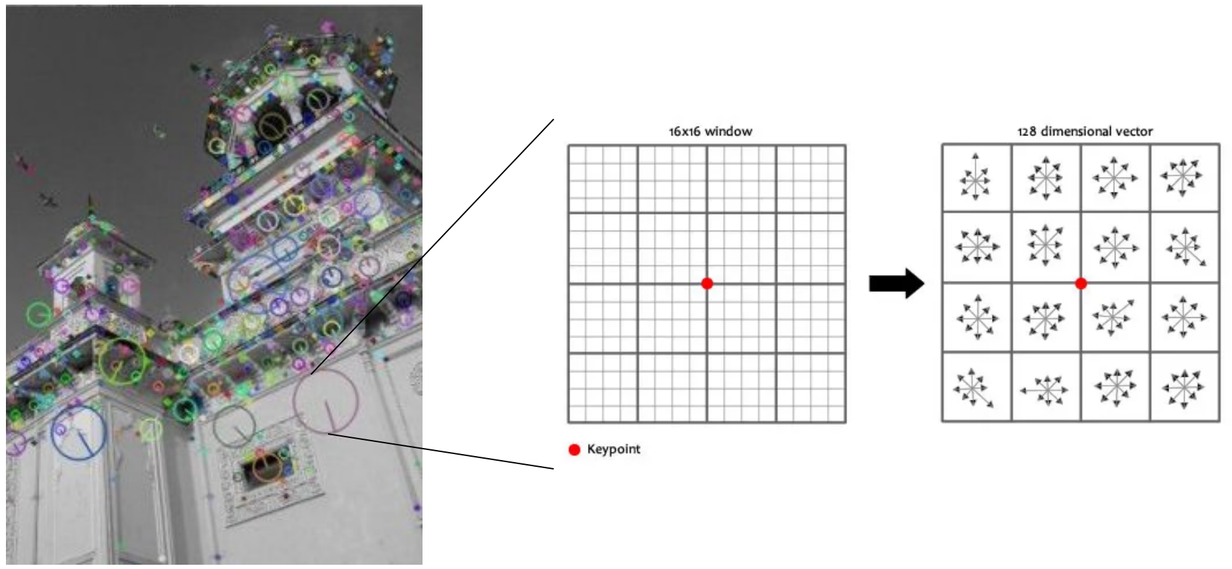
$$tf\_idf_{t,d} = tf_{t,d} * idf_{t,d} = tf_{t,d} * log\left(\frac{N}{df_t}\right)$$

SIFT: Scale Invariant Feature Transform

Steps

- Scale-space Extrema Detection: introduces DoG, efficient approximation of NLoG to find local extrema.
- Keypoints localization: eliminates any low-contrast keypoints and edge keypoints and what remains is strong interest points.
- Orientation assignment: creates keypoints with same location and scale, but different directions.
- Keypoint descriptors: invariant to rotation, scale, and brightness.

SIFT Descriptors



Lowe, David G.. “Distinctive Image Features from Scale-Invariant Keypoints.” International Journal of Computer Vision 60 (2004): 91-110.