CS5242 Project:
Search Real-world Fruit Images
For Fruit Sketches

Content

- *Introduction*
- *Data Collection*
- *Data Exploration*
- *Data Pre-processing*
- *Model Implementation & Analysis*
- *Conclusion*

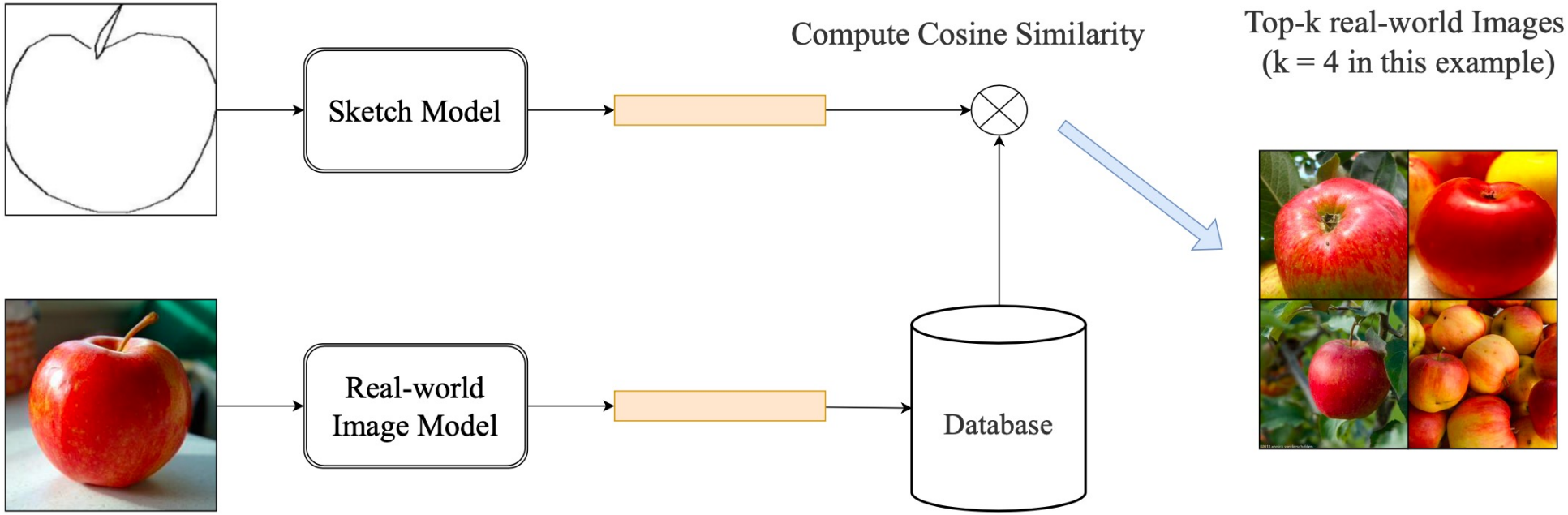
Introduction

Motivation and Problem Statement

Inspired by some AI-generated images from sketches, we realize that sketch images with few lines contain important content. It would be interesting and important to study how to utilize limited information in sketches.

Methodology

We aim to implement a sketch-based image search engine that can return *k* real-world fruit images for the user-input sketch. To match real-world images and sketches, we use models to extract image features and compute the similarities among their features. Search Engine would return images with highest similarity.

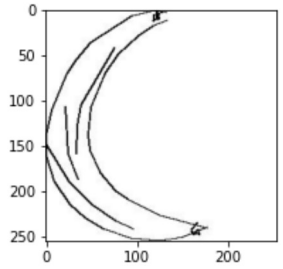


Introduction

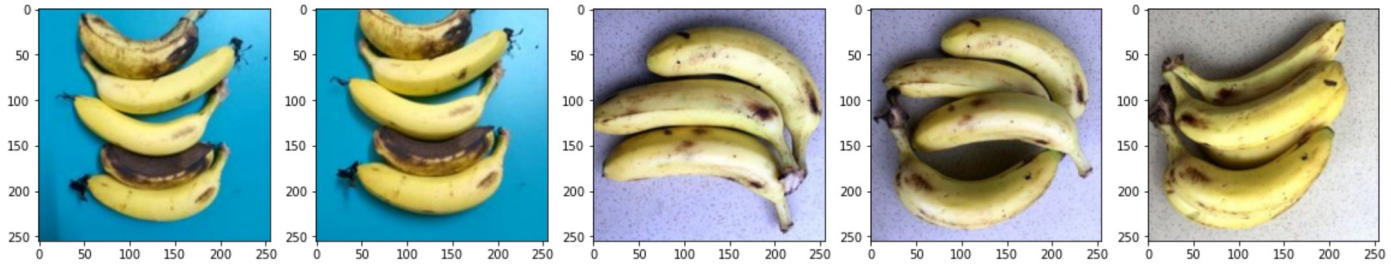
Motivation and Problem Statement

Inspired by some AI-generated images from sketches, we realize that sketch images with few lines contain important content. It would be interesting and important to study how to utilize limited information in sketches.

---------- User Input Sketch ----------



---------- Search Engine Outputs ----------



********** THE END **********

Methodology

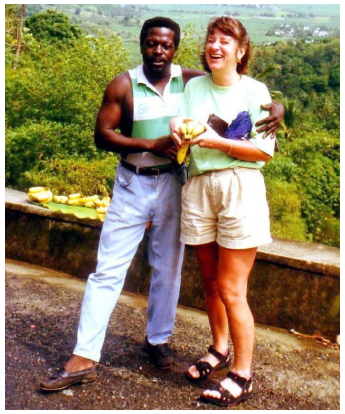
We aim to implement a sketch-based image search engine that can return *k* real-world fruit images for the user-input sketch. To match real-world images and sketches, we use models to extract image features and compute the similarities among their features. Search Engine would return images with highest similarity.

Data Scraping

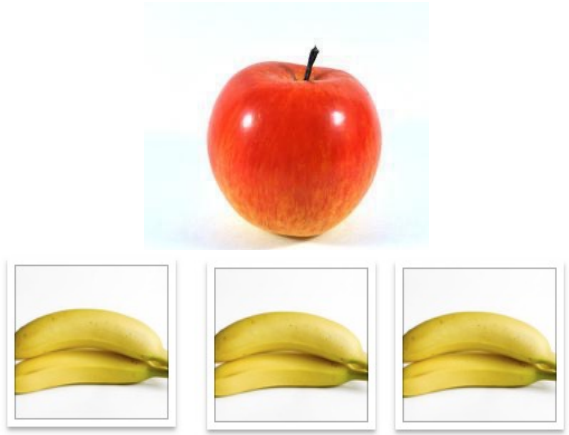
Sketch Image (QuickDraw)

- Method: Python quickdraw API
- Dataset size: 10,000 * 7
- Process:
  1. Download quickdraw package
  2. Using “QuickDrawDataGroup “ get Quickdraw objects
  3. Save quickdraw objects in image format

Real-world Image



Poor quality Image

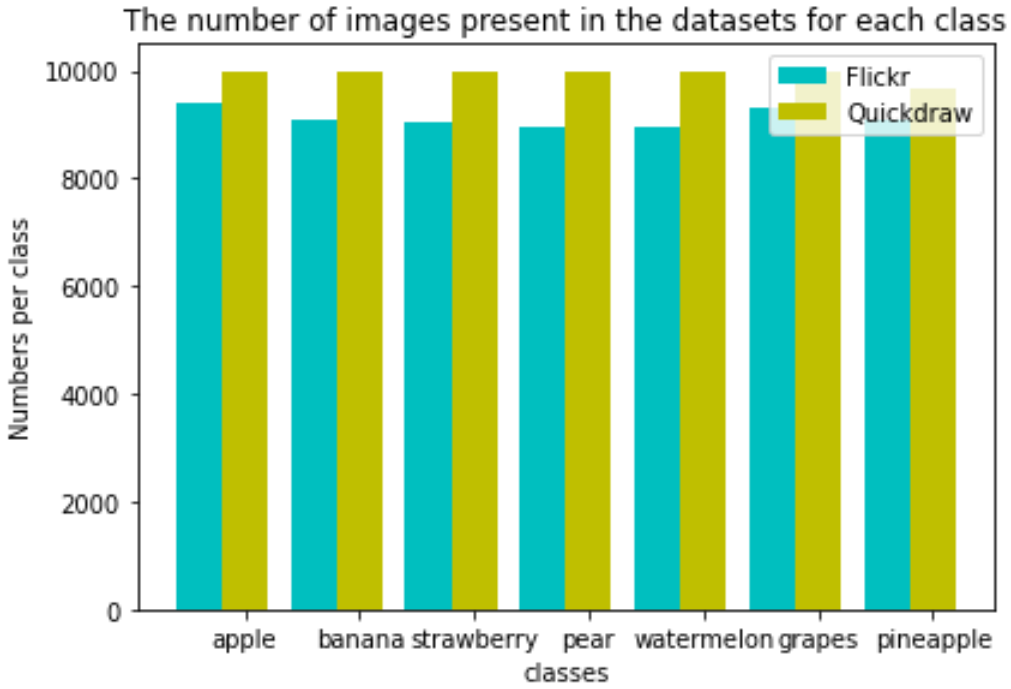


Repeated Image

- Method: Flickr API
- Dataset size: ~9,000*7
- Process:
  1. Specify search query
  2. Using “flickr.walk” to returns an iterable object
  3. Download certain size images and handle exception: download failed, image poor quality

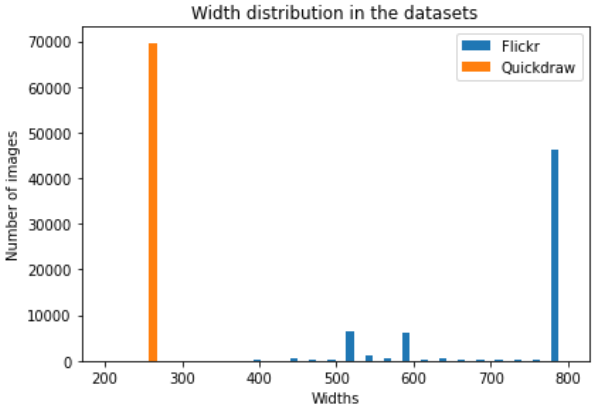
Data Exploration

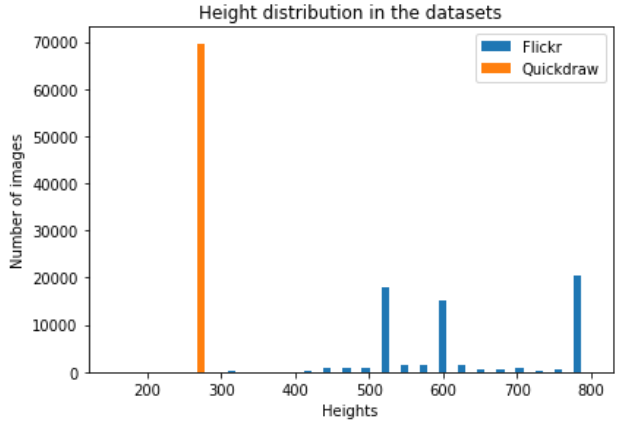
Statistics of Image Number



Insights: Images collected from quickdraw are all successfully downloaded; while some images collected by Flickr may be lost, some with poor quality.

Statistics of Image Shape





Insights: Images collected from quickdraw have the same size of (255, 255); while sizes of images collected from Flickr are diverse and should be scaled and clipped to the same size in the later preprocessing step.

Data Pre-processing

Step1: Data Cleaning

- Delete gray real-world fruit images
- Delete duplicated Images
- Manually delete some poor quality images

gray images of pears

poor quality images of pineapples

Step2: Data Normalization

- Crop and resize all images to 255*255
- Balance image numbers in different fruit categories.

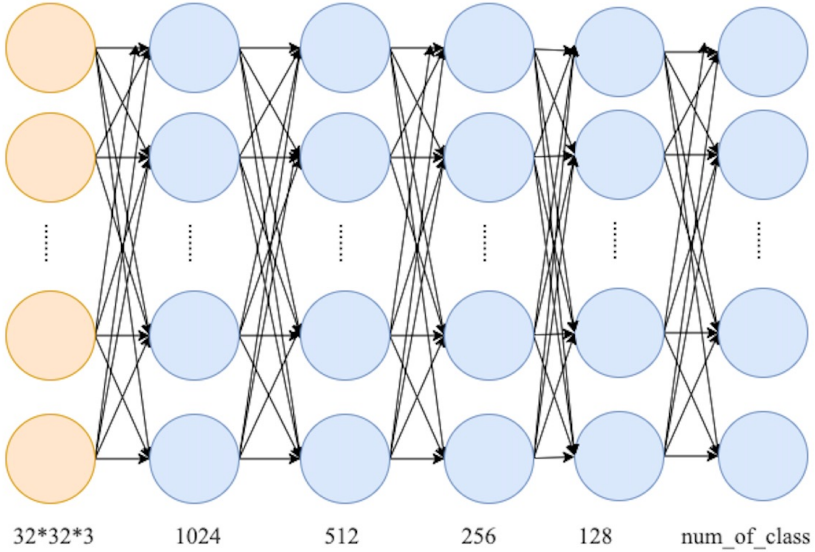
  Now we have 5k images for each kind of fruit

Image Shape (800, 533)

Image Shape (255, 255)

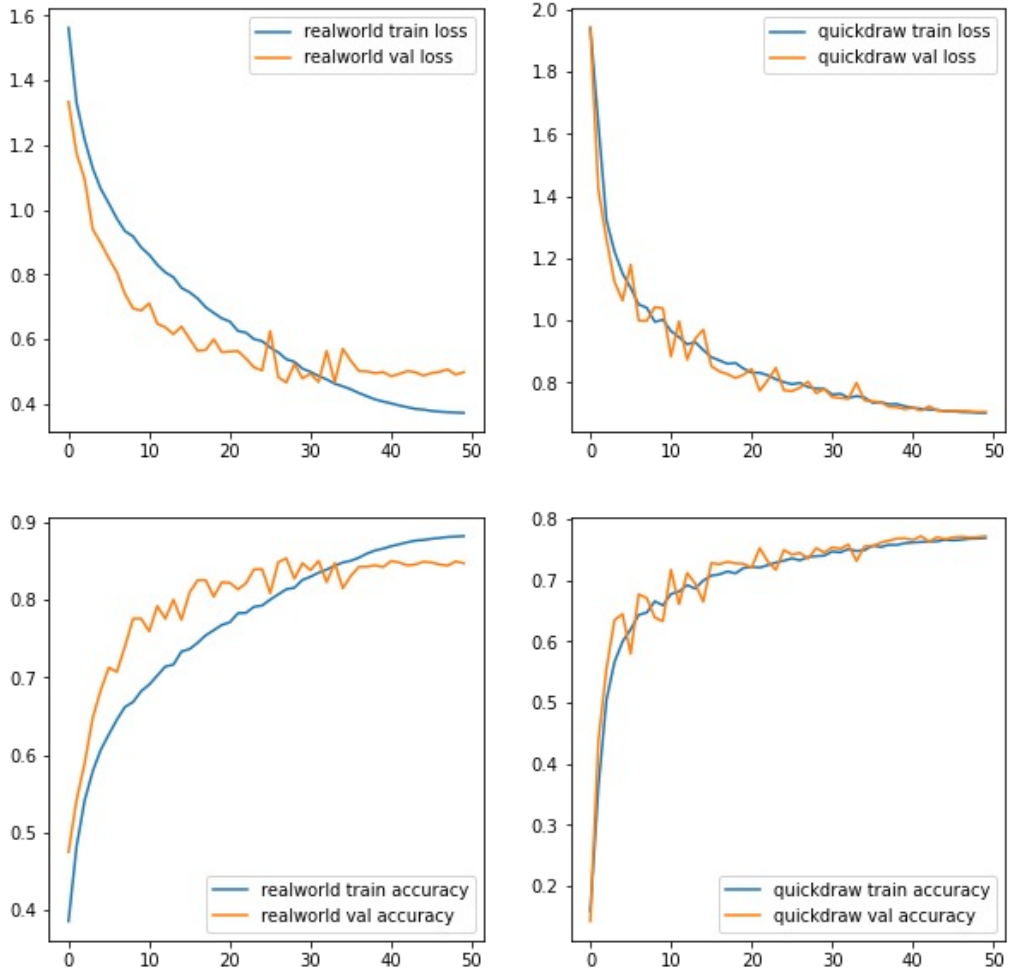
Multilayer Perception

Model Structure



Hyperparameter Tune

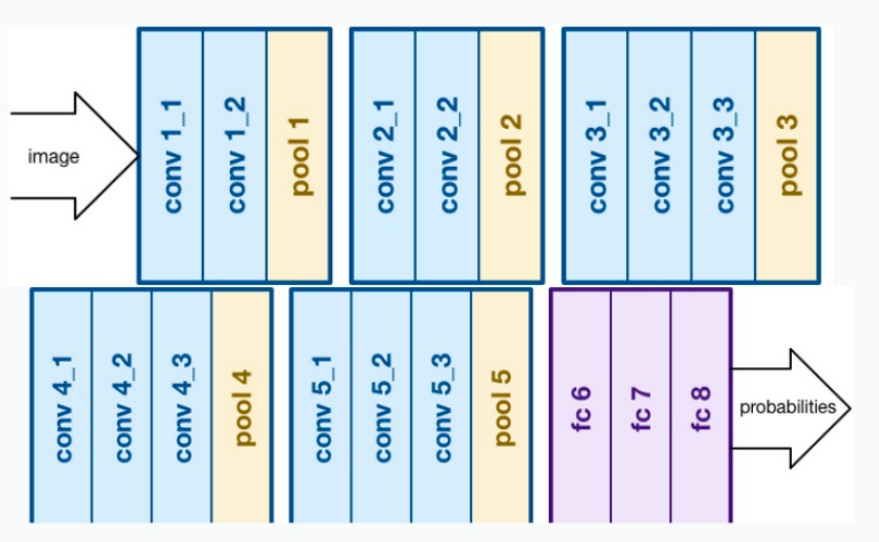
Linear Layer Number	3	4	5	6
Real-world Image Accuracy	82.7	84.3	84.7	84.0
Sketch Image Accuracy	72.74	75.7	77.2	76.3



After 50 epochs:
Real-world Image Classification Accuracy: 84.7%
Quickdraw Image Classification Accuracy: 77.2%.
Sketch-based Top-3 Image Search Accuracy: 61.4%.

VGG 16 *(focus on depth)*

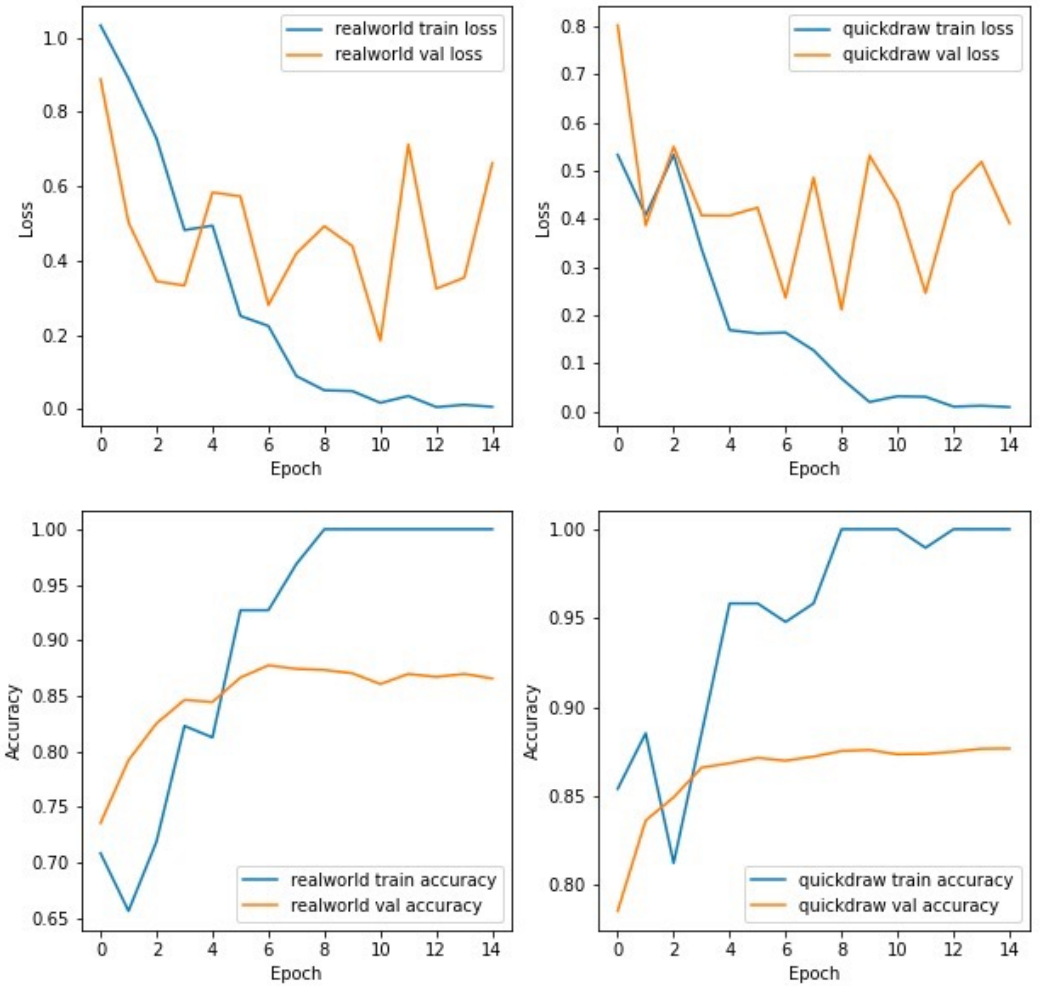
Model Structure



The structure of the VGG network is very consistent, we use a 3x3 convolution and a 2x2 max pooling from beginning to end.

Reduce Overfitting problem

Due to limited dataset and the complexity of model, we try lower learning rate, add dropout layers and weight decay.

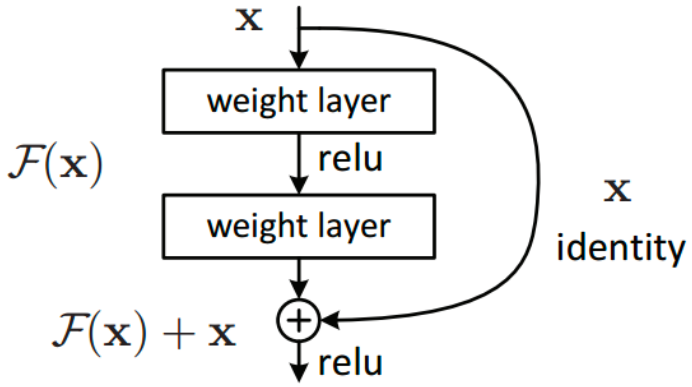


After 15 epochs:
Real-world Image Classification Accuracy: 85.9%
Quickdraw Image Classification Accuracy: 70.7 %.
Sketch-based Top-3 Image Search Accuracy: 13.7%.

ResNet 32

Model Structure

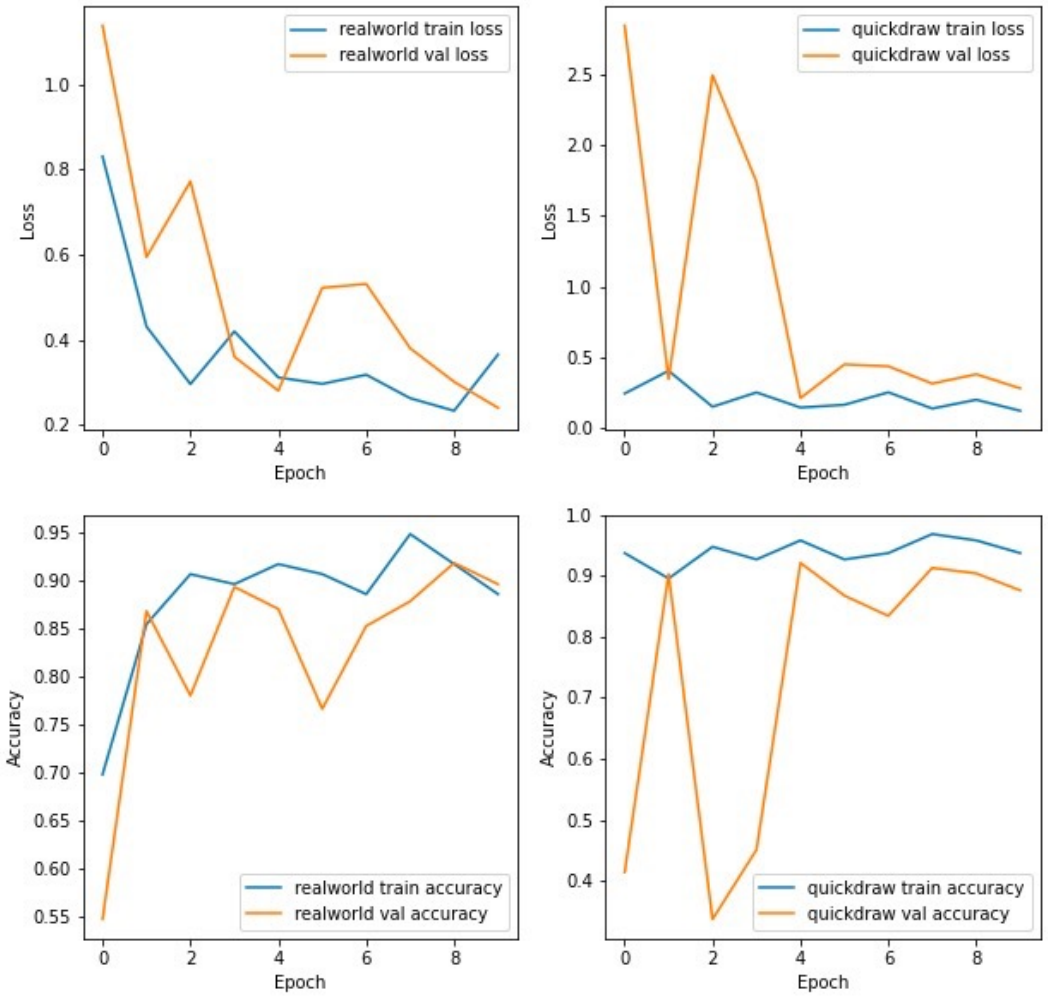
1. Create Residual Block



2. Build our ResNet with a helper function _make_layer to add the layers one by one along with the Residual Block.

Overfitting problem

ResNet architecture has better inference than VGG networks with similar training time in our case.

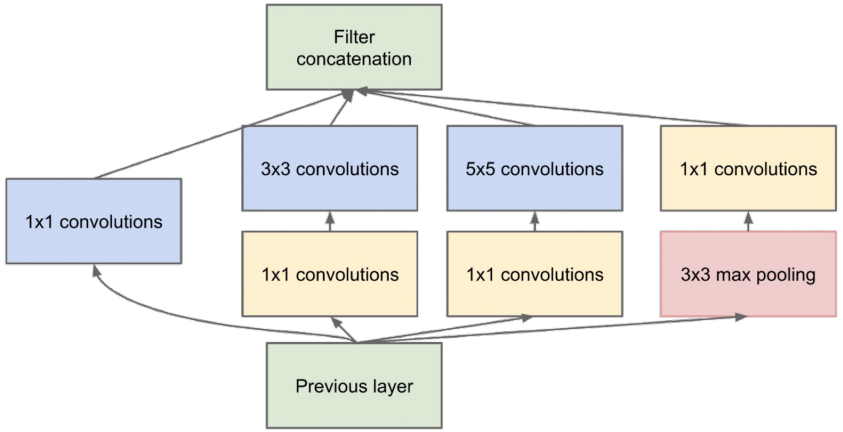


After 50 epochs:
Real-world Image Classification Accuracy: 84.7%
Quickdraw Image Classification Accuracy: 77.2%.
Sketch-based Top-3 Image Search Accuracy: 61.4%.

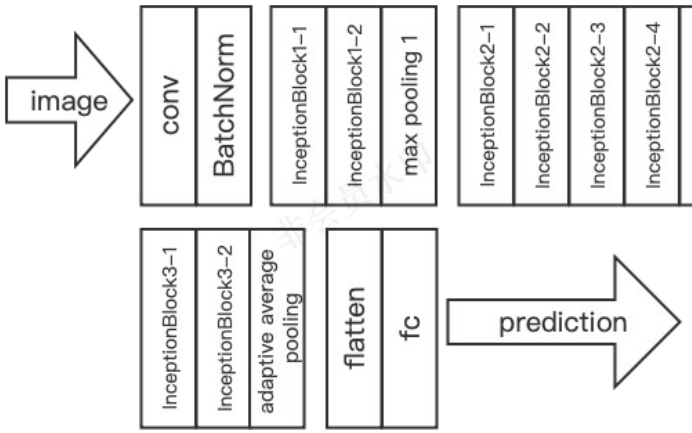
Inception

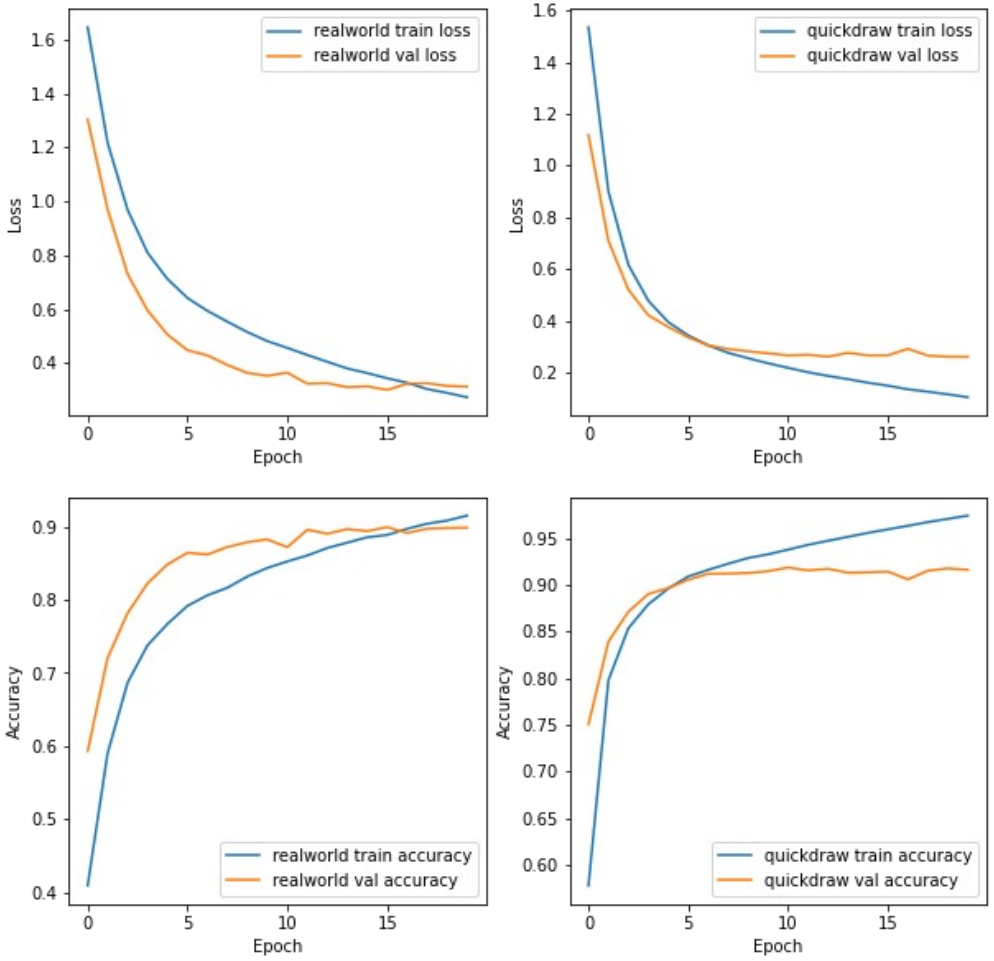
Model Structure

1. Create Inception Block



2. Build Inception Network

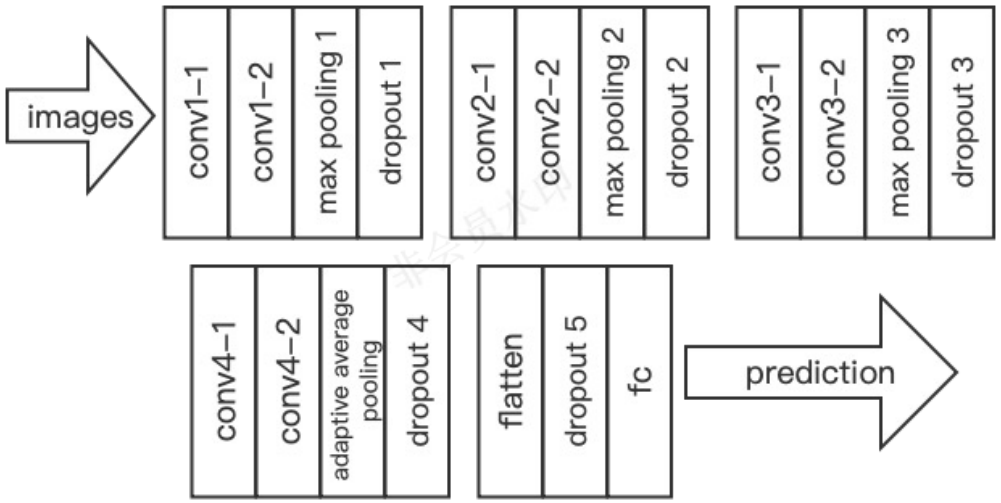


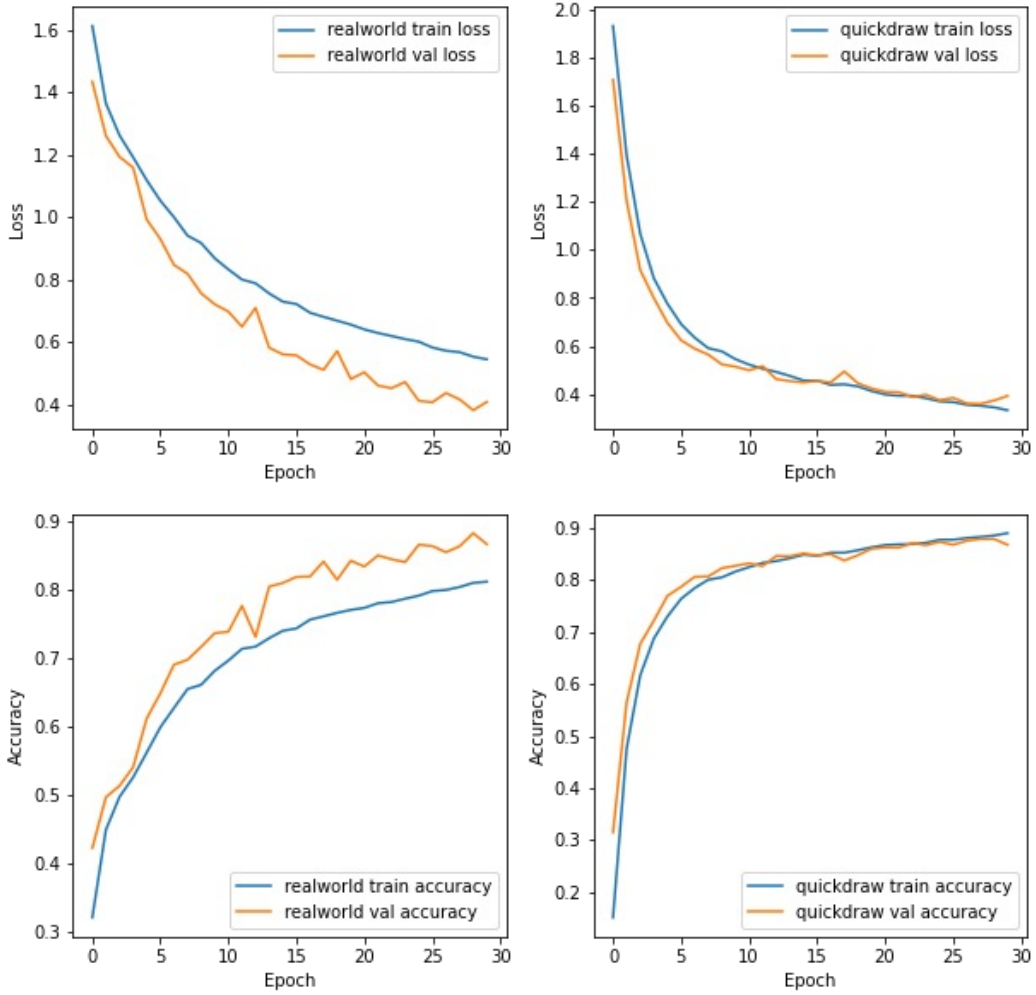


After 20 epochs:
Real-world Image Classification Accuracy: 89.0%
Quickdraw Image Classification Accuracy: 91.0%.
Sketch-based Top-3 Image Search Accuracy: 85.2%.

Convolutional Neuron Network

Model Structure





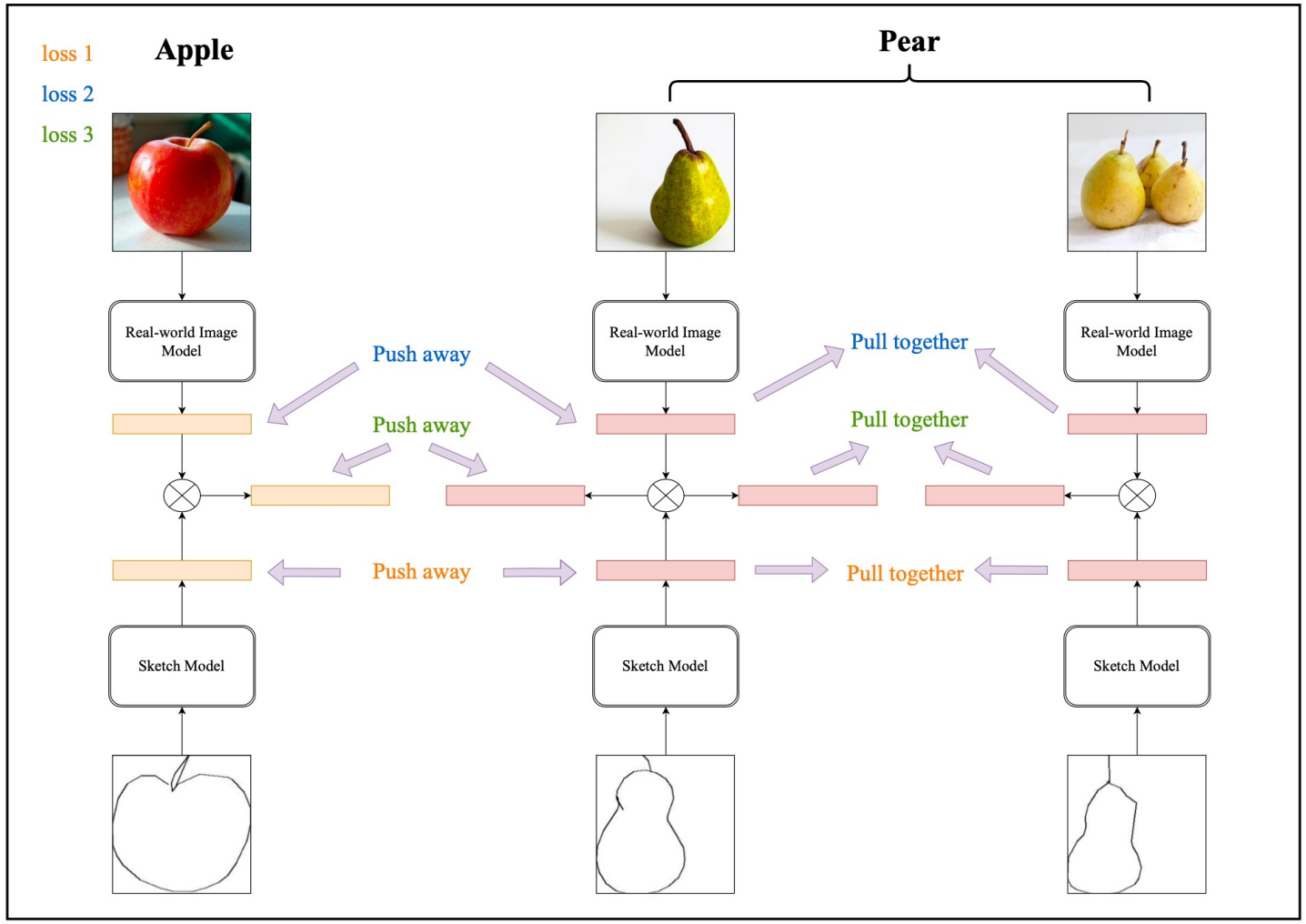
Hyperparameter Tune

Dropout Rate	0	0.1	0.2	0.3
Real-world Image Accuracy	86%	90%	85%	85%
Sketch Image Accuracy	86%	90%	89%	91%

After 30 epochs:
Real-world Image Classification Accuracy: 89.83%
Quickdraw Image Classification Accuracy: 90.10%.
Sketch-based Top-3 Image Search Accuracy: 75.2%.

CNN with Contrastive Loss

Model Structure



Contrastive Loss

$$loss = -\log \frac{\sum_{label_i=label_j} \exp(sim(f_i, f_j)/\tau)}{\sum_k \exp(sim(f_i, f_k)/\tau)}$$

where f represents image features and τ represents temperature in Contrastive loss formula

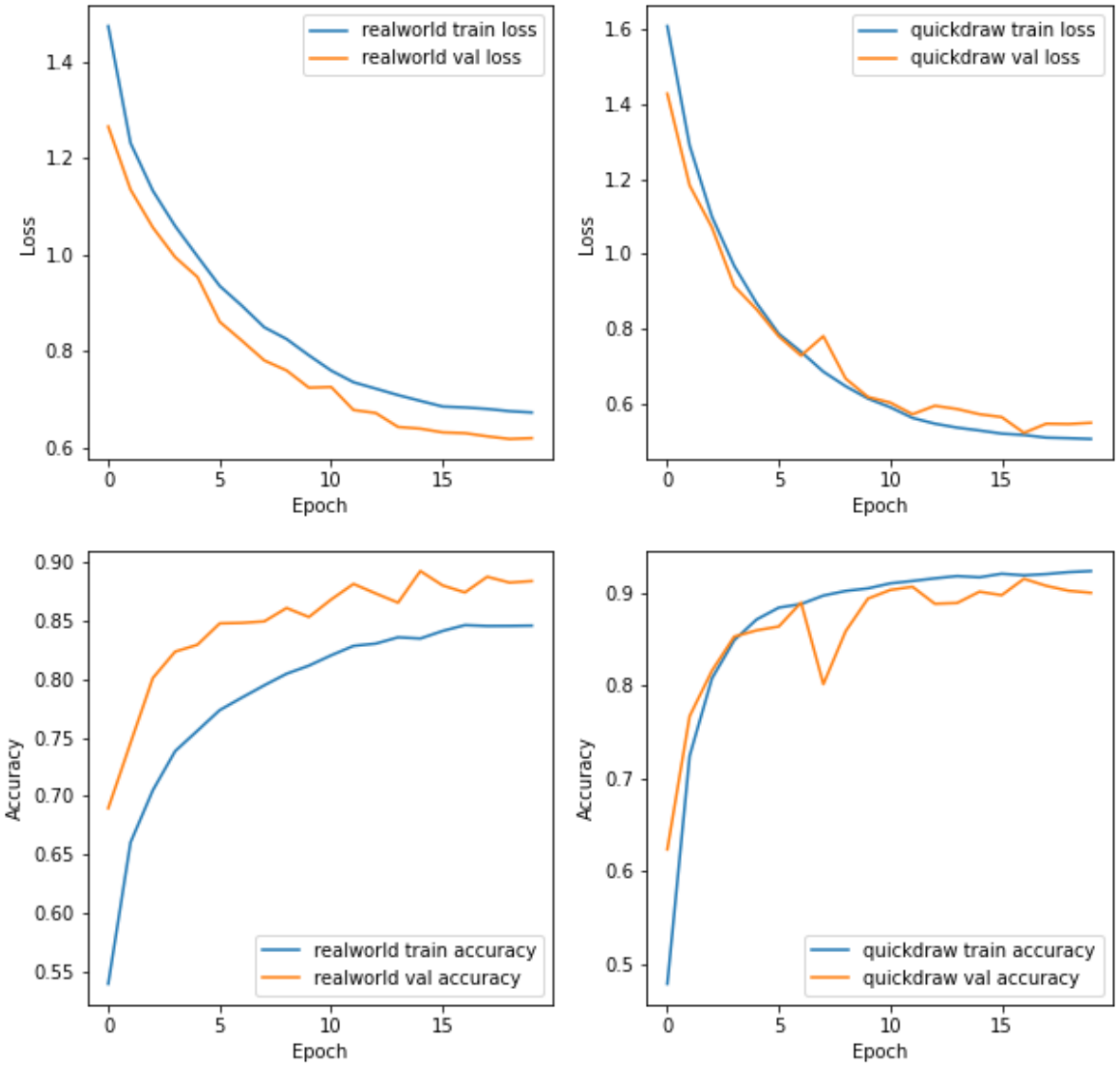
loss1: $f = f_{realworld} = M_{realworld}(x_{realworld})$

loss2: $f = f_{sketch} = M_{sketch}(x_{sketch})$

loss3: $f = f_{realworld} * f_{sketch}$

where f represents features, M represents Model and x represents input images

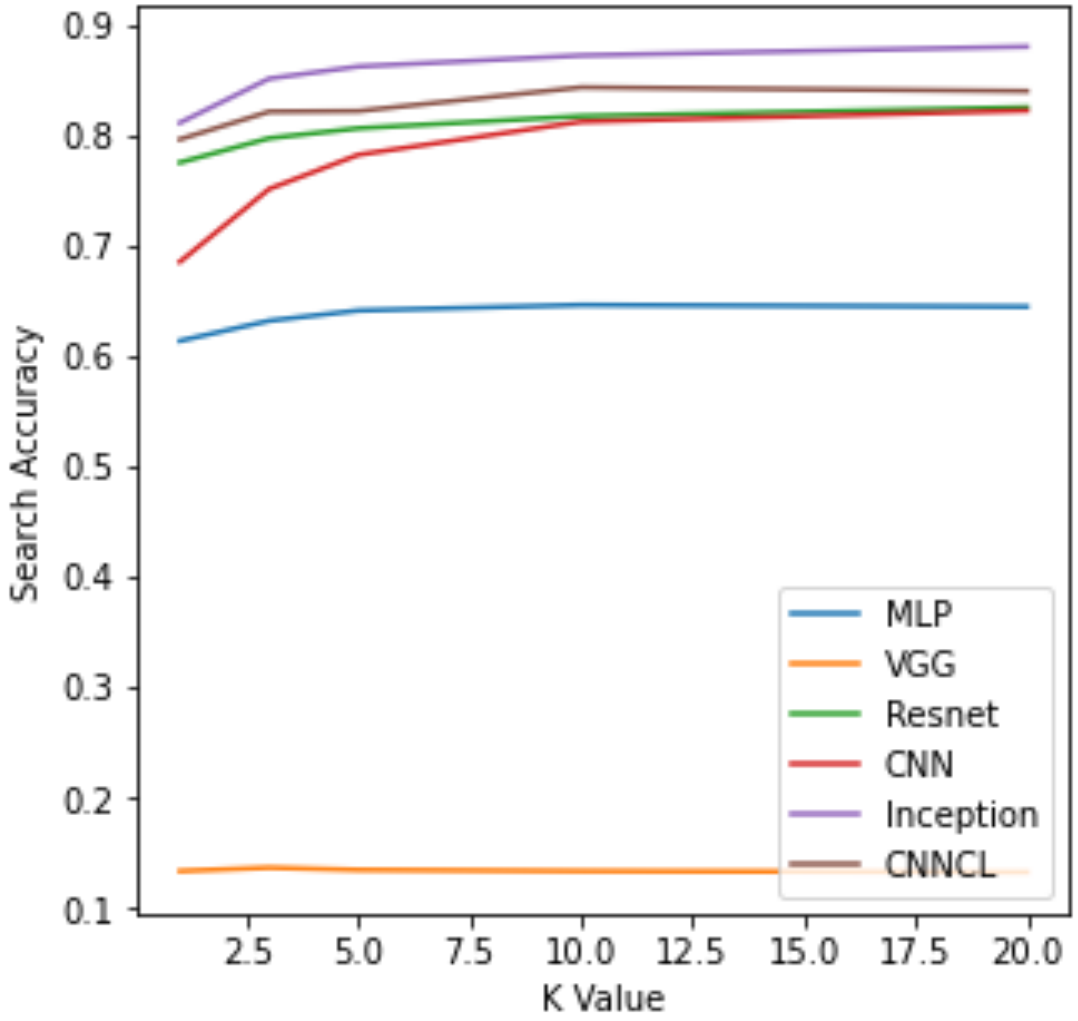
CNN with Contrastive Loss



After 20 epochs:
Real-world Image Classification Accuracy: 88.37%
Quickdraw Image Classification Accuracy: 89.96%.
Sketch-based Top-3 Real-world Image Search: 82.21%.

Top-k Real-world Images Accuracy	CNN	CNN with CL
1	56.70%	79.70%
3	59.20%	82.21%
5	61.60%	82.26%
10	63.70%	84.41%
20	66.40%	84.07%

Conclusion



Achievements

Based on all the results we get, the Inception model achieve the highest accuracy due to its well-organized inception blocks, which allows the network to look at the same data with different receptive fields.

ResNet bottleneck layers encourages our model to compress alient features for the target variable to best fit in the available space in our case. are more obvious.

Our proposed CNN with Contrastive Loss gains significant improvement compared to other baselines.

Future Improvement

We plan to increase the dataset size and expand our model to be applied on more image classes.

Thank you for listening