[CS/DS 864] Machine Learning - 1 project

Photo Triage Benchmark

Darshan Bhat - MT2015038
Keerthan Pai K - MT2015053

December 21, 2016

1 Problem

People often take a series of nearly redundant pictures to capture a moment or scene. However, selecting photos to keep or share from a large collection is a painful chore. To address this problem, we seek a relative quality measure within a series of photos taken of the same scene, which can be used for automatic photo triage. Towards this end, a large dataset comprised of photo series distilled from personal photo albums have been gathered. By augmenting the dataset with ground truth human preferences among photos within each series, the problem is to establish a benchmark for measuring the effectiveness of algorithmic approaches to modeling human preferences.

2 Exploratory analysis of the dataset

The dataset contains 15,545 unedited photos distilled from personal photo albums. The photos are organized in 5,953 series. For each series, human preferences are collected by a crowd-sourced user study. The following figure shows several example series, annotated with human preferences.

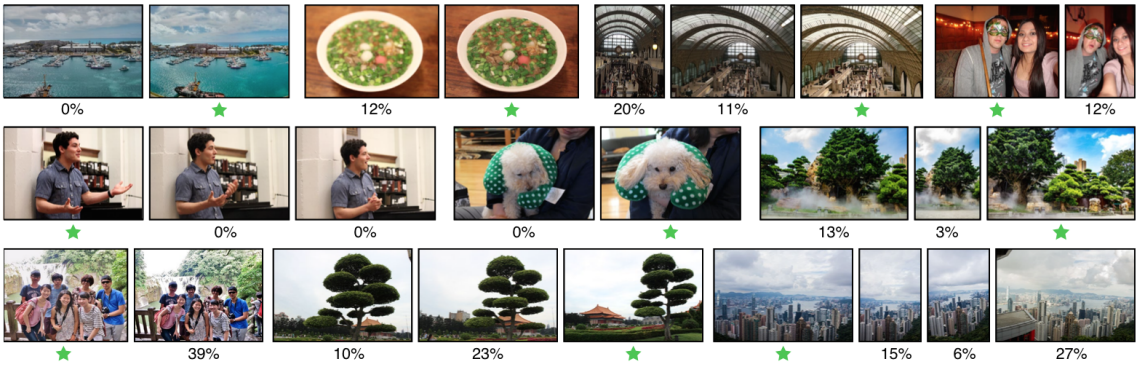


Photo Triage: The photo with the green star in each series is the one preferred by the majority of people, while the percentage below each other photo indicates what fraction of people would prefer that photo over the starred one in the same series.

Out of the 5,953 series, 4560 are randomly sampled for training, 195 for validation, and the remaining 967 for testing. The dataset includes the following folders and files:

- **train_pairlist.txt** lists the pairs in all training photo series.
  The format is "$\#SERIES\_ID\ \#PHOTO1\_IND\ \#PHOTO2\_IND$
  $\#PREFERENCE\_RATIO\_of\_PHOTO1\_OVER\_PHOTO2$
  $\#RANK\_of\_PHOTO1 RANK\_of\_PHOTO2$"

- **val_pairlist.txt** list all the pairs in all validation photo series. To test the performance of learning the human preferences offline, save the result of the predictor into a textfile and run test.m. The result could be either binary or float for the preferene of PHOTO1 over PHOTO2 for each pair.

- **train_val_series.mat** lists more information about the testing photo series, such as the Bradley-Terry scores modelled from human preferences.

- **train_val_imgs/** includes all the images which are resized in 800x800 with its aspect ratio preserved. The format is "$\#SERIES\_ID(\%06d)-\#PHOTO\_IND(\%02d).JPG$".

3 Algorithmic Approach

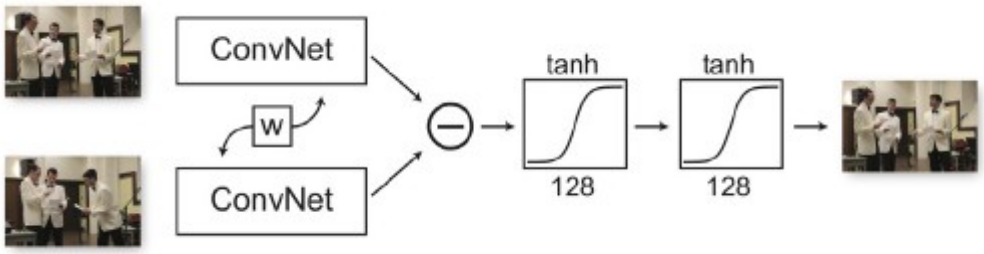
Many hand tuned features like color, lighting, composition, clarity, SIFT features can be considered for the problem. But as suggested in the website, feature extracted from pre-trained network like AlexNet, VGGNet will tend to outperform the hand tuned features. We will use pre-trained ConvNet to extract the features.

The training images belong to different categories like nature, selfie, indoor, outdoor etc. So the training using single image features will not make sense and will not lead to convergence. Rather we will take two images from the series of similar images and use the difference of their individual features as a new feature for training. Features are extracted from two exactly same ConvNets for two different images in a pair(each feature of around 4096 values). A new feature is formed by taking the difference of these two feature vector. This new feature is then used to train a new model with binary output denoting which image is the better of the two.

So we will process the images pairwise to predict the better image among the two. This result can be then clubbed to decide the overall best image of the series.

4 Reason behind choosing the DeepNet model:

Extensive work has been done on understanding the human preference over a pair of similar images[Nishiyama et al. 2011; Kaufman et al. 2012; Zhang et al. 2013]. The major factors which affects the preferences are found to be color and lightning, face position, composition of foreground and background and clarity. These are called the handheld features.
Many people have used these hand held features to train the model for this photo triage problem, but the DeepNet extracted features tend to outperform this. However the state of the art method for this problem uses the combination of both. In our project we have only confined to the DeepNet based features.
What is siamese architecture:
If we want to train a full deep neural network we need millions of images. In our data set we have only around 10,000 images for photo triage. But we had millions of images available for general classification task. Researchers had found that this can be used as a good feature extractor for the photo triage problem. But this particular problem is different from the classification problem. Classification model takes only one image as the input, where as here we should send a pair of images. Also the the result of the model should be same irrespective of the order of the input images.
In Siamese architecture [Bromley et al. 1993], two inputs are sent into two identical sub-networks independently, which share the same network parameters in both training and prediction phases.



The input of our model is a pair of images (I_1, I_2). We aim at learning a function p:IxI $\mapsto$ -1,1 where 1 means the first image is better and -1 means the opposite. The model p should be skew symmetric, that is, if the input images are flipped, the output should also flip, p(I_1,I_2) = -p(I_2,I_1)

At the second stage of learning f, a multi-layer perceptron classifies the features of the image pairs. In our perceptron, there are two hidden layers, each of which has a 128-dimension output. Each hidden layer comprises a linear fully connected layer and a nonlinear activation layer. Because f is supposed to be odd, we use tanh in activation layers. The outputs of the second hidden layer are fed to a two-way Softmax. The outputs of Softmax indicate which of the two input images is better.

5 Results:

With DeepNet extracted features as explained above and sending the difference of these features for a pair to a multi-layer perceptron classifier, we were able to classify the best image of the pair as 1 and the other as 0.

.

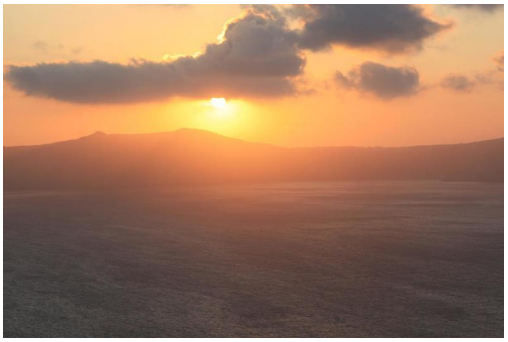


Figure 5.1

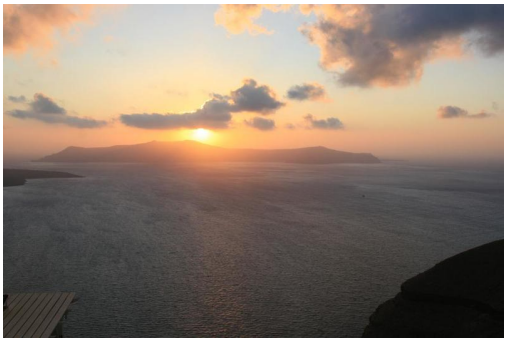


Figure 5.2

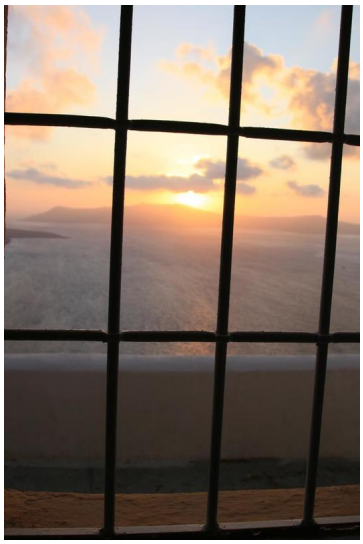


Figure 5.3

For one the test image series with three images above, we get the following predictions:

- **[5.1, 5.2] = [0, 1]** : denoting Figure 5.2 is better than Figure 5.1
- **[5.1, 5.3] = [1, 0]** : denoting Figure 5.1 is better than Figure 5.3
- **[5.2, 5.3] = [1, 0]** : denoting Figure 5.2 is better than Figure 5.3

Once we have these pairwise ranking, we can triage the images within a series to find the best image, in this case the best image of the three is Figure 5.2

Our model was trained with the sample of 1000 series of images totalling 2885 images and tested with 100 different pairs of images. The test accuracy was found to be 66%, where as the state of the art model is gives the accuracy around 78%.