

A Personalized Photo Enhancement Agent

Jason Tellis · Ling Ouyang · Mugdha Jain

Received: date / Accepted: date

Abstract An intelligent agent incorporating user's personal preference and feedback was developed for personalized photo enhancement. Most image enhancement techniques rely on heuristic algorithms with some static or probabilistic parameters. This agent learns the user's latent aesthetic preferences from their past images. Using this knowledge, it enhances and customizes new images to match the user's taste. The model was trained on user-clicked pictures and user provided labels such as 'liked' and 'disliked'. The agent learns from the training dataset and applies this to improve the user's new photos such that the image quality and style is tailored to the user's personal taste. To evaluate the performance, users were given a blind test to decide which they liked more between the original and the enhanced photo. The images mostly focus on having a human subject.

Keywords Image Classification · Image Enhancement

1 Introduction

In this era of being connected to everyone through our mobile devices, sharing photos has become an important form of communication. Everyone wants to look their best in photos to get the most attention from their social circle. Hence, photo enhancement plays an important role in our lives. In this project, we were interested in discovering how intuitive aesthetics of an

M. Jain
E-mail: mugdha.jain@sjsu.edu

L. Ouyang
E-mail: ling.ouyang@sjsu.edu

J. Tellis
E-mail: prashanthjason.tellis@sjsu.edu

image could be modeled to enhance photos instead of relying on heuristic/probabilistic approaches.

1.1 Related Work

A lot of research effort has gone into investigating how latent aesthetic features can be used to enhance images. A photo editing system, Creatism, that mimics landscape photographer's workflow to produce enhanced photos [1] was recently introduced by Google researchers. Another group [2] exploited professional photographs from the web to train their aesthetics model by examining pairs of views without having to apply any explicit photographic rules.

2 Materials & Methods

2.1 Environment

The Environment class operated on a set of images captured by users under various lighting conditions (sunny, daylight, twilight, night), scene types (portrait, landscape) and locations (indoor and outdoor). It presented the labelled images in the training set to the agent to build a model for classification.

2.2 Sensor

The sensor module passed the percepts to the agent function. The user interface for the sensor allows the user to select a folder of images for training the classifier.

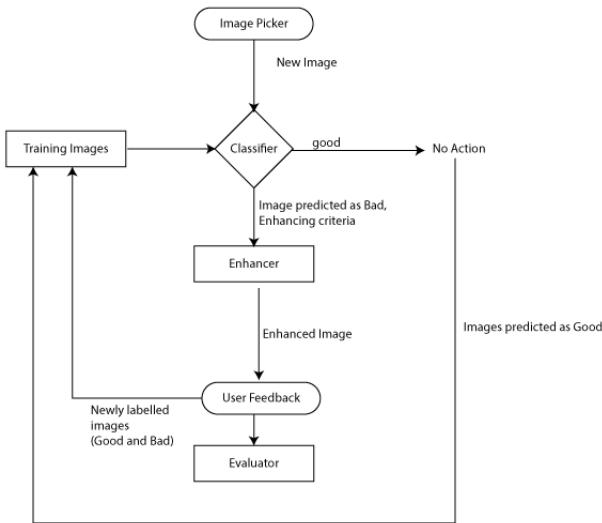


Fig. 1 High Level Design of the Agent

2.3 Percepts

The percepts are the images to be classified and enhanced by the agent. Examples of good and bad images are shown in Fig 2-3.



Fig. 2 Sample Good Image

2.4 Agent Function

The 3 tasks that the agent function performs are:

1. Feature Extraction
2. Image Classification
3. Enhancement of Images

2.4.1 Image Feature Extraction

Feature extraction was performed to retrieve meaningful information so that the agent learned what a good



Fig. 3 Bad Image

photo should look like. The following 7 features were used:

1. Number of Faces
2. Skin Probability: Possibility of each pixel being skin
3. Skin Colour
4. Face Sharpness
5. Face Smoothness
6. Image Entropy(Contrast and Brightness)
7. Global Edge Sharpness

2.4.2 Classifier

The classifier classifies a percept as good or bad depending on the classification model used(k-NN, SVM and Random Forest). The images classified as bad were sent to the enhancer module to be modified.

2.4.3 Enhancer

A bad image is enhanced based on specific enhancement criteria that is calculated by taking the average feature vector of all the good images in the training data set at each iteration.

2.5 Actuator

The actuator displayed all the processed images (unenhanced and enhanced) to the user and recorded their

feedback. Finally the images that the user tagged as good were sent back into the training dataset to be used the next time an image needs to be classified. This allowed the agent to incrementally learn and accumulate the user's preference of good images. The user interface for the actuator is a window that displays all the processed test images and allows the user to either 'Like' or 'Dislike' the image.

2.6 Performance Measures

The performance of the classifier and enhancer was measured separately.

Evaluating the Classifier: The accuracy of predictions obtained for each new set of images was used to evaluate the classifier.

Evaluating the Enhancer: The enhanced images are displayed to the user for their feedback. Once the enhanced images are displayed to the user for feedback, the enhancer effectiveness 'e' was calculated using the following formula:

$$e = \frac{w_1(\text{bad} \rightarrow \text{good}) + w_2(\text{good} \rightarrow \text{bad}) + w_3(\text{bad} \rightarrow \text{bad}) + w_4(\text{good} \rightarrow \text{good})}{w_1(\text{No.ofTestImages})}$$

where:

$$\begin{aligned} w_1 &= 2 \\ w_2 &= -2 \\ w_3 &= -1 \\ w_4 &= 0 \end{aligned}$$

The working principle is to keep a count of whether the enhancements are liked by the user as shown in the Table:

Table 1 Evaluation Parameters for Enhancer

Classification Label	Enhanced Image Label	Weight
Bad	Good	2
Good	Bad	-2
Bad	Bad	-1
Good	Good	0

The weighted sum of these parameters is normalized by dividing it by the number of test images to obtain a value between -1 and 1. Thus, a more positive score indicates that the enhancer performed well and a more negative score implies that the enhancer did not do well.

3 Experiments

3.1 Training Data Set

A training data set of 40 images per user was built manually and labelled 'good' or 'bad' based on the user's personal taste.

3.2 Validation Data Set

Validation data set was built using cross validation of the images in the training data set.

3.3 Test Data Set

To assess the performance of our agent, new images of the user were obtained.

4 Results

4.1 k Nearest Neighbors

The k Nearest Neighbors classifier produced the following confusion matrix for k=5 with an accuracy of 90.32%.

Table 2 Confusion Matrix for kNN

Class	Predicted Good	Predicted Bad
Actual Good	24	1
Actual Bad	2	4

4.2 SVM

The Support Vector Machine Classifier produced the following confusion matrix for linear kernel with an accuracy of 87.10%.

Table 3 Confusion Matrix for SVM

Class	Predicted Good	Predicted bad
Actual Good	24	1
Actual Bad	3	3

4.3 Random Forest

The Random Forest Classifier produced the following confusion matrix for 10 trees with an accuracy of 67.74%.

Table 4 Confusion Matrix for Random Forest

Class	Predicted Good	Predicted bad
Actual Good	19	6
Actual Bad	4	2

4.4 Enhancer Results

The following images demonstrate the enhanced image against the original image.



Fig. 4 Original Image



Fig. 5 Enhanced Image

There is a clear distinction between the enhanced and original image. For Fig. 4-5, the overall sharpness of the image and the skin tone has been enhanced. For Fig.6-7, the face has been brightened and smoothed and the global contrast has been enhanced.



Fig. 6 Original Image



Fig. 7 Enhanced Image

The enhancer is evaluated by calculating how effective its enhancements were on a test data set of 31 images.

Table 5 Confusion Matrix for Evaluating the Agent

Class	Predicted Good	Predicted bad
User-labelled Good	0	16
User-labelled Bad	8	7

The accuracy of the agent was found to be 22.58%. The effectiveness for this set of test data was found to be - 0.5968 indicating that the agent did not do well.

4.5 Future Work

The following tasks need to be improved/implemented in the future:

1. Improve the UI for the agent.
2. Automate fetching images from social media accounts of the user.
3. Feature Engineering to extensively personalize the features extracted for each individual user.
4. Automating annotation of training data by making use of social media metrics.

References

1. H. Fang, M. Zhang, CoRR **abs/1707.03491** (2017). URL <http://arxiv.org/abs/1707.03491>
2. Y. Chen, J. Klopp, M. Sun, S. Chien, K. Ma, CoRR **abs/1702.00503** (2017). URL <http://arxiv.org/abs/1702.00503>