**Photo Rater**: Photographs Auto-Selector with Deep Learning

Wentao Guo, Charlie Ruan, Claire Zhou  
Department of Computer Science, Cornell University  
(Dated: November 2022)

*Photo Rater* is a computer vision project that uses neural networks to help photographers select the best photo among those that are taken based on the same scene. This process is usually referred to as “culling” in photography, and it can be tedious and time-consuming if done manually. *Photo Rater* utilizes three separate neural networks to complete such a task: one for general image quality assessment, one for classifying whether the photo is blurry (either due to unsteady hands or out-of-focusness), and one for assessing general aesthetics (including the composition of the photo, among others). After feeding the image through each neural network, *Photo Rater* outputs a final score for each image, ranking them based on this score and presenting it to the user.

I. BACKGROUND AND APPLICATION SETTING

When a photographer sees a beautiful scenery, he or she would take multiple pictures to guarantee that the best moment is captured. However, when the photographer returns home, he or she would have to pick the best photo from a group of images which are taken from the same scene repeatedly. The photographer has to go over such an image selection process for each scene, and this could be a very hard and tedious task.

Therefore, the goal of this project *Photo Rater* is to make a neural network learn what a good photograph is, while keeping in mind that each picture is taken in a similar setting. The *Photo Rater* would assist photographers in the image selection process by grading the images based on general quality and aesthetic assessment.

II. HIGH-LEVEL ARCHITECTURE

We approach this project by thinking about what differentiates a good photograph from a bad one under this setting. If a set of photos are taken consecutively on the same scenery, then there would not be too much difference in their color, lighting, or content. What should be emphasized is whether a photograph has any random noise during rendering due to the processing of the image, whether the photographer’s hands were shaking, whether the photo was out of focus, whether the photo has a good composition, and whether the photo is aesthetic in a generic sense.

With these goals in mind, we will train three neural networks separately, each with a specific task:

1. Image Quality Assessor: learns to differentiate a well-rendered image from one that has random noise
2. Blurriness Classifier: learns to differentiate a sharp photo from one that is either motion-blurred (due to shaking hands) or out-of-focus
3. Aesthetic Assessor (ReLIC): learns what a good composition is and other generic aesthetics

Figure 1: General architecture of our neural network.

Note that we will refer to these sub-NN’s as NN1, NN2, and NN3 in the rest of the paper.

Finally, after training all three neural networks, which each will produce a score for an input, we will train a final neural network, which we call the Combined NN, that learns the best coefficients to combine the three scores, producing an ultimate score. With that being said, the overall architecture of our model is as shown in Figure 1.

In Section IV, we will explain each sub-NN and the Combined neural network in details. We first cover the datasets we used for training in the next section.

III. DATASET

A. TID2013

NN1 is trained on a dataset called TID2013. It is suitable for quality assessment, as it has 24 types of distortions that simulate sources of noise such as image acquisition noises and JPEG compression noises. The distortion levels in TID2013 are determined by human
participants’ pairwise selection results with a reference image (i.e., an image without any distortion). Therefore, TID2013 is a good fit for non-reference image quality assessment tasks, and the labels represent human observable ratings.

TID2013 has 25 reference images, and each image is distorted in 24 types of noises, with 5 levels for each (image, distortion type). In total, we have 3000 distorted images (25 × 24 × 5). All the reference images are obtained by cropping from Kodak Lossless True Color Image Suite.

B. Aesthetic Visual Analysis (AVA)

NN2 and NN3 are trained based on a dataset called Aesthetic Visual Analysis (AVA) [2]. This dataset consists of 250,000 photos from the website dpchallenge, where users upload their own photographs to be scored by other users. We picked this dataset because of its realisticness, wide coverage of types of photos, and the fact that the dataset is under the context of photography (hence the emphasis on aesthetics).

On top of the images, the AVA dataset also consists of the specific type of the photo (e.g., macro, sports) and its corresponding scorings (from 1 to 10, with 10 being the highest) from the users. Figure 2 is an example of the data.

IV. IMPLEMENTATION DETAILS

A. Neural Network 1: Image Quality Assessor

We believe that one major source of a “bad” image is the noises added to the image during rendering. Therefore, it would be helpful for the photographers if we can develop a neural network that scores the images based on their general quality. We found that this task is actually a well-studied field called Non-reference Image Quality Assessment (NR-IQA) in which people try to assess the quality of an image without a reference image, and it is a good fit in our photo-selection scenario: suppose the photographer has taken 20 images, but we do not know what the reference image (i.e., an “ideal photo”) looks like. In other words, we have to make a clear quality assessment without an actual target.

For the architecture, we adapt the model from [3] for IQA as shown in Figure 3. This model consists of a backbone network (a ResNet in its actual implementation), and a few more fully-connected layers. We first adopt the pretrained weights from the author’s github, and we follow this paper’s procedure for finetuning on TID2013 as shown in Figure 4.

For training, we select distortion types 1, 2, 19, and 23 from the TID2013 dataset which represent noises from image acquisition step (described in Table I and shown in Figure 5). However, we only use them for testing our model, but for training we still use the entire TID2013 dataset since TID2013 is a small dataset and we found images in other distortion types still suitable for training.
Figure 5: An example in the TID2013 dataset. For each type of distortion we selected, distortion level 1 and 5 are included.

our model. More importantly, we want more examples in training to generalize our model.

| Type | Description                      |
|------|----------------------------------|
| 1    | Additive Gaussian Noise          |
| 2    | Additive Noise on Color          |
| 19   | Multiplicative Gaussian Noise    |
| 23   | Chromatic Aberrations            |

Table I: Description for each type of distortion in TID2013 that we include. We choose the most representative noise types that one may encounter in a photography setting.

During training, we use Adam optimizer (lr=1e-3, eps=1e-8, beta1, beta2=0.9, 0.999, no weight decay or damping) with a ReducedLROnPlateau type learning rate scheduler to stabilize our losses [4]. For data augmentation, we apply random vertical and horizontal flips and rotations as we believe a transformed image still preserves the general image qualities.

B. Neural Network 2: Blurriness Classifier

For a set of pictures that are taken consecutively, one of the key differences that separate a good one from a bad one is whether the picture is “sharp”. There can be two main reasons for a picture to not be “sharp”, namely the photographer’s hands are shaking instead of being steady, and the picture is simply out of focus. Therefore, the Blurriness Classifier is dedicated to learning these two types of blur simultaneously.

Since we would like to make this neural network a classifier, we will include the original AVA image as a “sharp” image, and two corresponding manually distorted AVA images as blurry images. We used relatively straightforward computer vision methods to simulate motion blur and out-of-focusness. Figure 6 demonstrates the two types of manual distortions. Note that the major difference, among others, between the two distortions is that motion blur has a randomly picked direction associated with the blurriness, since motion blur simulates the hand-shaking of a photographer, which has an associated direction. The good data labeled as 0 is the original image from AVA, and the bad data labeled as 1 is the same image, but manually distorted by us.

Since this neural network model deals with a relatively straightforward task of image classification, we decided to use DenseNet121 from [5], connected with a linear layer that outputs a score. Since the output represents the probability that an image is blurry, we will take that into account when outputting the score to the final Combined NN.

C. Neural Network 3: Aesthetic Assessor (ReLIC)

Neural network 3 determines the aesthetic quality of the pictures. We wanted to not only focus on the colors and brightness, but also take composition features into account, predicting its general aesthetic value. ReLIC is an architecture we found that could perform this task well [6].

ReLIC first extracts visual features through MobileNetV2 as the backbone network [7]. It then learns compositional and overall aesthetic features using a fully
connected graph. In this graph, edge attributes (\(e\)) represent image composition, and global attributes (\(u\)) represent miscellaneous aesthetic features. The model then combines the features together in a fully connected layer to produce prediction scores, as shown in Figure 7.

We finetuned the existing model and changed its training loss function to cross entropy loss. The model was then trained on the entire original AVA dataset, converging after about 900 steps.

**D. Combined Neural Network**

Finally, after training each of the three neural networks independently, with each of them outputting a score, we need to figure out a way to combine the three scores, hence holistically rating the image with everything taken into account.

There are several options to do this. We could simply make the final score the sum of the three scores, but this does not guarantee the best information to be presented since the weight of each score might not be the same. We could also simply return the three scores as a tuple to the user, but this still leaves the user with too much work. Therefore, we decide to use back-propagation to learn the best set of coefficients:

\[ \lambda_1 s_1 + \lambda_2 s_2 + \lambda_3 s_3 \]

where \(s_i\) is the output score from the \(i^{th}\) sub-NN and \(\lambda_i\) is the corresponding coefficient.

The input and the label for this Combined NN might be the tricky part here for training. We use the AVA dataset for this task. For each image, we computed the average score from the users, which we will call the original score. In our training data, we will first include the original AVA image, whose labels will simply be the original score. However, in order to assess the importance of NN1 and NN2, we will manually distort the images again using the six methods: additive Gaussian noise, multiplicative Gaussian noise, chromatic aberration, and salt-pepper noise from NN1’s training; motion blur and out-of-focus from NN2’s training.

Therefore, for each original image from AVA, we will create six different kinds of distorted images. The corresponding label (i.e. score) will be the original score divided by three, penalizing for the poor image quality. The goal is to make the Combined NN learn that a distorted image will have a much lower score compared to its original aesthetic score.

The Combined NN not only serves as a model that learns the most suitable coefficients, but also serves as the main script of our pipeline (i.e. the “main.py”). Referring to Figure 1 after the input image is fed into the model, we pass it to the three sub-NNs separately, each producing a score. We then connect the three outputs with the final output using a linear layer, where the final output will be the score returned to the user. The three sub-NN’s will each be a pretrained model, and hence are frozen and not trainable. On the other hand, the linear layer’s parameters will be trainable. Therefore, during back-propagation, we only compute the gradient with respect to \(\lambda_1, \lambda_2, \lambda_3\).

**V. EVALUATION AND OUTCOME**

For evaluation, we want to know how well each of the sub-NNs performs, how well the Combined NN performs, how the Combined NN’s scoring on AVA differs from the scoring of the users, and whether the Combined NN’s results under a real application setting make sense. Each question will be answered by a subsection below.

**A. Neural Network 1 Evaluation**

We use a (0.95, 0.05) train-test split to assess our IQA. The results are not too satisfactory as shown in Table II, as in training we found the spearman rank correlation generally fluctuates between \(-0.1\) to \(0.1\), and we believe that it is because our dataset is still really small. It would improve if we could find better datasets to perform a few rounds of pre-training and then apply the fine-tuning task to obtain the optimal results.

| Metric                        | Result Based on Training |
|-------------------------------|--------------------------|
| Spearman ranking correlation  | -0.326                   |
| Pearson correlation           | -0.243                   |
| root mean squared error       | 1.587                    |
| mean absolute error           | 2.519                    |

**Table II:** The training outcomes of neural network 1 Image Quality Assessor.

**B. Neural Network 2 Evaluation**

We also use a (0.95, 0.05) train-test split to assess our Blurriness Classifier. NN2’s training was relatively successful, converging to an accuracy of around 85% on the testing dataset, as shown in Figure 8. Since our accuracy is relatively high using our manually distorted dataset, we believe that the evaluation result is successful for this specific neural network.
C. Neural Network 3 Evaluation

The model ReLIC converged after about 900 steps. We recorded the Pearson linear correlation coefficient (plcc), Spearman’s rank-order correlation coefficient (srocc) and classification accuracy for the last step, as shown in Table III. For classification accuracy, images with a score higher than 5 are set to be Good, and those with a score lower than 5 are set to be Bad. These indices were around 75 to 80, indicating a successful learning.

| Metric                  | Result Based on Training |
|-------------------------|--------------------------|
| Spearman ranking correlation | 0.749                    |
| Pearson correlation      | 0.748                    |
| classification accuracy  | 0.818                    |

Table III: The training outcomes of neural network 3 Aesthetic Assessor, measured by various metrics.

D. Combined Neural Network vs. AVA

The evaluation of the combined NN is basically the evaluation of our entire pipeline, since the Combined NN is essentially the “main.py” for our project.

The first step of evaluating the Combined NN is to simply forward-pass the original images from the AVA dataset without any distortion. We then compare the score outputted by the Combined NN with the original score provided by AVA. Since there is not any manually distortion added, we expect the two scores to be similar, as both our pipeline and the users will holistically evaluate the images. Figure 9 depicts the comparison of two sets of scores: while the blue line represents the AVA’s original average score for each picture, the orange line represents the scores outputted by Combined NN for each picture. The overall result is not ideal, as we are expecting that the orange line will exhibit a similar pattern as the blue line. The Combined NN’s scores have an extremely small variance, only producing scores between 2 and 4, while the scale is supposed to be 0 and 10.

We believe that there are several potential reasons that lead to this result. First, as mentioned in Section IV, when training the Combined NN, we have to distort the original AVA images to let our Image Quality Assessor and Blurriness Classifier play a role. However, the main issue arises when we need to assign a label (i.e. score) to such manually-distorted images, since there are no ground-truth scores available. We simply assign them with the original score in AVA divided by 3, a scale determined arbitrarily. Besides, the ratio between original images and distorted images in our dataset is determined arbitrarily as well, which may not reflect the correct distribution of good and bad images in real life. Therefore, during training, the Combined NN probably “cheats” by lowering all scores (including the undistorted images) in order to minimize the empirical loss. Besides, another apparent reason is that our NN1 (Image Quality Assessor) does not perform quite well during its training as mentioned earlier, which damages the performance of Combined NN as well.

However, looking at the bright side of the result, based on Figure 9 we witness an upward trend in the orange line. This means that our Combined NN is still able to tell the difference between a good image and a bad image. This gives us hope to further evaluate our Combined NN.

E. Combined Neural Network on Real Use Cases

The main downside of the previous evaluation is that it does not test our pipeline under the context of scoring images that are taken in similar settings, which is the pri-
Figure 10: Rankings of two sets of photos based on the scores outputted by Combined NN.

mary prerequisite for Photo Rater’s usage. Therefore, we will specifically test out the Combined NN’s performance under such settings in this subsection.

As shown in Figure 10, we took two sets of images, one set on the forest (macro) and one set on a guitar (micro). Within each set, there are four different images. “Best Fit” represents the most aesthetic image based on common sense, sharp and without any shaking. “Wrong View” represents a relatively unaesthetic image due to a bad composition, though still has sharp quality. “Shaking” represents an image taken when the photographer’s hands are not steady, and “Out of Focus” simply represents an out-of-focus image. After feeding each image to the Combined NN, we ranked the results based on their scores. For both cases, the “Best Fit” is ranked as the best image, and “Out Of Focus” is ranked as the worst image, which makes complete sense, since we almost never want an image that is out of focus. The ranking of “Wrong View” and “Shaking” varies, which is understandable as well, since “bad composition” is relatively subjective.

Therefore, based on the evaluation of the real-world applications, we can conclude that the Combined NN is able to produce meaningful results despite the low-variance scoring observed in the previous subsection.

VI. CONCLUSION

Our project Photo Rater: Photographs Auto-Selection with Deep Learning aims to help photographers select the best photo among a set of photos taken in a similar setting. The pipeline determines the best photo based on its overall image quality assessed by our Image Quality Assessor, its sharpness assessed by our Blurriness Classifier, and its composition and general aesthetics based on ReLIC. The three aspects are weighted correspondingly based on the training result of our Combined NN, the forward pass to which can be viewed as the “main.py” of our project.

The evaluation of our project is successful to some extent. Though the scoring produced by the Combined NN exhibits a low variance, a photographer is still able to receive useful information if we compare a set of images taken in a similar setting – the exact intentional use case of our pipeline.

Some possible areas to improve are: a better model for our Image Quality Assessor (NN1), a better method to create a dataset for training the Combined NN, and a more robust way of considering the distribution of good and bad images. After such improvements, if our model performs well, it can be potentially implemented in products like cameras, car dash cams, and GoPro, providing photography enthusiasts auto-selected aesthetic pictures efficiently.

[1] N. Ponomarenko, L. Jin, O. Jeremeiev, V. Lukin, K. Egiazarian, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti, and C.-C. J. Kuo, Signal Processing: Image Communication 30, 57–77 (2015).
[2] N. Murray, L. Marchesotti, and F. Perronnin, in 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012) pp. 2408–2415.
[3] H. Zhu, L. Li, J. Wu, W. Dong, and G. Shi, “Metaiq: Deep meta-learning for no-reference image quality assessment,” (2020).
[4] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” (2014).
[5] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) pp. 2261–2269.
[6] L. Zhao, M. Shang, F. Gao, R. Li, F. Huang, and J. Yu, Computer Vision and Image Understanding 199, 103024 (2020).
[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, (2018). 10.48550/ARXIV.1801.04381