Data Challenge - REGAIND

Leo Paillier
ENS Paris-Saclay
leo.paillier@gmail.com

Jules Scholler
ENS Paris-Saclay
jules.scholler@gmail.com

Abstract

This report is part of the MSc MVA Sparse Representations, Wavelets and Classification course. It presents our work on the data challenge proposed by REGAIND [1] *which aims at ranking portrait pictures on a scale of 0 to 24 based on their aesthetic qualities. All our code is freely available on GitHub* [2].

Introduction

The objective of this data challenge is to predict an aesthetic score for portrait photos. This problem is complex for at least three reasons: (i) images belongs to a very high-dimension space (several millions), (ii) what makes a picture beautiful can sometime be very subjective and finally (iii) portraits specificity where a minor change in the pan angle or a closed eye for instance can have a major impact on its perceived beauty.

Our work is mainly composed of three parts. First, we focus on building a baseline testing method using only the features provided by REGAIND. Then, combining art and science literature we explore new features and their impact. At the same time we try to understand why the algorithm fails for certain kind of portraits shedding light on new candidate features. Last but not least, we present our work on the regression optimization, that is, increase our accuracy with a fixed features data-set by testing several regression methods.

The second part deals mainly with image processing while the third part focuses on machine-learning optimization.

1. Baseline method

Starting such a wide challenge as predicting an aesthetic score for a portrait led us to carefully construct a baseline for our method evaluation. This will allow us to obtain a starting point for further improvements and to follow a logical path.

1.1. Data loading and shaping

We started by importing and converting the data provided by REGAIND into processable information. The data was composed of real numbers (e.g. position of the face) and attributes (e.g. right eye is open) and the first step was to convert the attributes into vectors (e.g. -1 for a closed eye and 1 for an opened eye). In the following, we refer to the raw processable data available for both training and testing dataset as D_{meta}.

1.2. Evaluation framework

We followed a typical k-fold evaluation framework following these steps:

1. Split the testing set into 10 random folds.
2. For $i = \{1, .., 10\}$, train the system with 9 folds (all folds except the i-th) and test it with the i-th fold. The i-th accuracy is given by the Spearman's rank correlation coefficient ρ_i.
3. The final method accuracy is given by averaging all ρ_i.

We used the Spearman's rank correlation coefficient as metric because it is the one used for the challenge. The baseline method takes D_{meta} as input, train a linear Support Vector Machine (SVM) using an automatically (with an heuristic on the Gram matrix) scaled Gaussian kernel on the standardized[3] D_{meta} feature vectors.

1.3. Results analysis

We obtained $\rho_{baseline} = 0.40$. The final accuracy is helpful for evaluating the different methods and tell whether a change led to an improvement or not. Nonetheless, the final accuracy does not carry information on where to find and correct the method errors/weaknesses. To do that we can construct other representations.

[1] https://challengedata.ens.fr

[2] https://github.com/JulesScholler/Regaind-ChallengesDATA

[3] Each feature vectors are centered and reduced in order to have 0 average and unit standard deviation

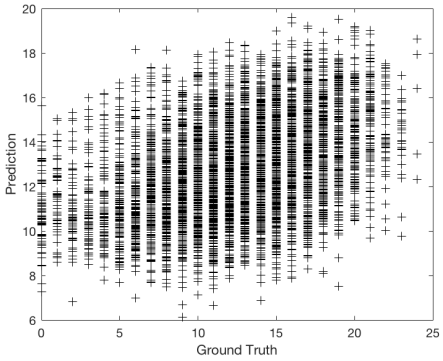

Figure 1. Predictions versus Ground truth for each data point

The predictions versus ground truth visualization enables us to easily interpret the performance of our method. An example is shown in Figure 1. From this we can derive a more synthetic representation displaying only the mean and standard deviation, see Figure 2. Finally we display score histograms for both predictions and ground truth to get a final insight on our method performance, see Figure 3.

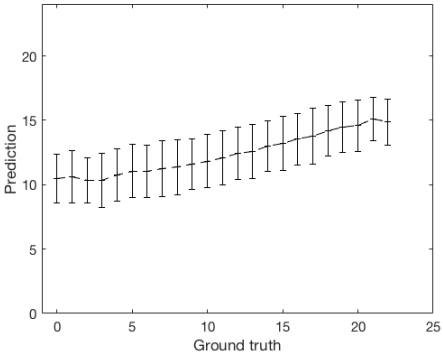

Figure 2. Predictions versus Ground truth with average and standard deviation

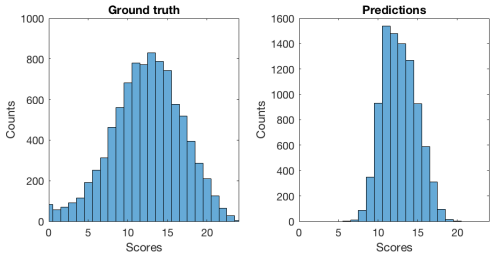

Figure 3. Scores histograms

These figures bring more information, e.g. here the method failed completely at predicting low scores, see Figures 1 and 2, besides the output distribution is not correct, see Figure 3.
Understanding the structure of the dataset is essential. For instance in Figure 3, one can observe that the ground truth scores seem to follow a Gaussian distribution. We can model this distribution by computing a Gaussian fit such as Equation 1 on the histogram. See Figure 4 for the plot, Table 1 for the parameters value and confidence bound and Table 2 for the fit error.

$$f : x \mapsto ae^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{1}$$

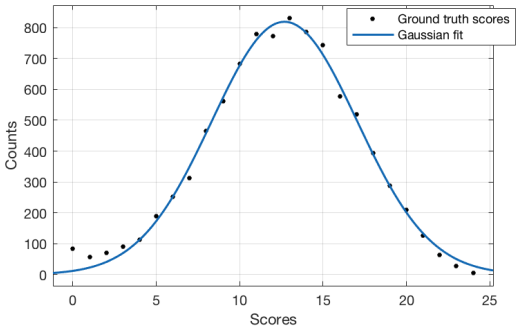

Figure 4. Scores histograms with Gaussian fit

Parameters	Value	Lower bound	Upper bound
a	818.9	794.7	843.2
μ	12.69	12.54	12.84
σ	4.363	4.214	4.513

Table 1. Coefficients with 95% confidence bounds

Error type	Value
SSE	1.55e+04
R-square	0.9921
Adjusted R-square	0.9914
RMSE	26.55

Table 2. Fit error

Except around 0 the scores distribution can be correctly approximated by a Gaussian distribution. In order to improve the method accuracy we need to understand how scores are given by adding more information. For example, the following Figures 5 and 6 present two images that were both ranked 0.

Figure 5. Bad lighting

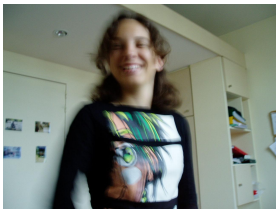

Figure 6. Motion blur

On can deduce from these figures that the lighting and the blur are important when it comes to predicting an aesthetic score and that they are of crucial importance to predict bad scores. As these attributes were not provided in the meta dataset we need to compute new features vector to improve the method accuracy.

2. New features

In this part we introduce the different features that we computed and added to the initial data-set. These features were designed by looking at bad prediction and by reading state-of-the-art literature. In the following we specify the article reference if it exists and denote by [#dim] the feature vectors dimension.

2.1. Definition

2.1.1 Intensity histogram [#10]

As presented in Figure 5, lighting can play a determining role when it comes to assessing the beauty of a portrait. Predicting dark portrait is fairly easy, indeed, by calculating the images intensity normalized histograms quantized on 10 values we can see that the dark portrait histogram is concentrated on the left, see Figure 7.

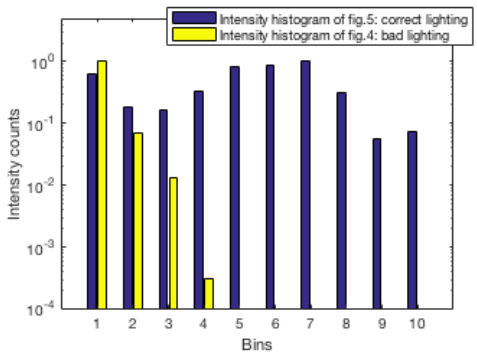

Figure 7. Quantized intensity histograms of Figures 5 and 6

2.1.2 HSV average [3] [#6]

The image is transformed from the RGB colorspace to HSV (Hue, Saturation and Brightness) and then the average Hue, Saturation and Brightness are computed on the whole image and the inner quadrant, which corresponds to the central region when the image is divided in a 3x3 grid.

2.1.3 Pleasure, Arousal, Dominance [3] [#3]

Based on a psychological study we computed these affective dimensions using Equations 2, 3 and 4.

$$Pleasure = 0.69\times V + 0.22\times S \quad (2)$$
$$Arousal = -0.31\times V + 0.60\times S \quad (3)$$
$$Dominance = 0.76\times V + 0.32\times S \quad (4)$$

2.1.4 Itten color histograms [5] [#20]

Histograms of H, S and V values quantized over 12, 3, and 5 bins respectively. Figure 8 displays the Itten color wheel developped by Johannes Itten in his book *The Art of Color*.

Figure 8. Johannes Itten color wheel, source:wikipedia

2.1.5 Itten Color Contrasts [5] [#3]

Standard deviation of the Itten Color Histograms distributions.

2.1.6 Contrast (Usual) [5] [#1]

The usual contrast, defined with Equation 5 considering that I is the grayscale intensity and $\bar{I}$ the average intensity on the image.

$$C_u = \frac{I_{max} + I_{min}}{\bar{I}} \quad (5)$$

2.1.7 Contrast (Michelson) [5] [#1]

The Michelson contrast, also called visibility, is defined according to Equation 6 considering that I is the grayscale intensity.

$$C_{Michelson} = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} \quad (6)$$

2.1.8 Contrast (Equalized) [5] [#1]

A contrast corresponding to the repartition of the different gray level using the formula in Equation 7 where u is the original image and u_{eq} the corresponding image but with an equalized histogram. See Figures 9 and 10 for example of histogram equalization.

$$C_{eq} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(u(i) - u_{eq}(i))^2} \tag{7}$$

Figure 9. Image 2 in gray scale

Figure 10. Image 2 histogram equalized

2.1.9 Spectral Saliency [1] [#9]

The method proposed in [1] requires to compute a radial average of the power spectrum. Such calculation is too expensive to be applied in reasonable times on the whole data-set. Our approach uses the same filtering method to compute the average power spectrum but we compute the L2 difference directly between the frequency domain without radial averaging:

Algorithm 1 Spectral saliency evaluation

Inputs: image u

Divide u into 9 sub-images $(v)_{1\leq i\leq 9}$ with the rule of thirds

for $i = 1$ **to** 9 **do**

Filter v_i with an averaging filter

Compute d_i the L^2 difference between u_i and v_i

end for

Compute $(\tilde{d})_{1\leq i\leq 9}$ the L^1 normalization of $(d)_{1\leq i\leq 9}$

Outputs: $(\tilde{d})_{1\leq i\leq 9}$

2.1.10 Motion face blur [6] [#10]

Detecting blur is not very difficult but assessing its origin is an harder task. Blur can be aesthetic in backgrounds (e.g. small field depth, velocity impression, etc.) but it can also be dramatic if it is a motion blur (e.g. the subject is moving during the exposition) as in Figure 6. Some papers [6] uses the wavelet transform and then try to recognize specific patterns. These methods are greedy and we tried to find a lighter algorithm for motion blur detection in order to process the whole data-set in reasonable times. Our algorithm performs as follows:

Algorithm 2 Motion face blur evaluation

Inputs: image u

Perform face detection on u and crop face into v

Compute grad(v)

Compute histogram of grad(v) into g quantized on 10 bins

Outputs: Blur evaluation g for the input image u

2.1.11 Sharpness [5] [#9]

In order to measure the sharpness of an image, we average the square norm of the two images obtained after applying a convolution between the image u and the following Sobolev gradient operators defined in Equation 8 and 9. This operation is done on the overall image plus the face, the background, the left and right eyes, eyebrows the mouth and the nose.

$$S_x = \frac{1}{4}\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \tag{8}$$

$$S_y = \frac{1}{4}\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \tag{9}$$

$$\text{Sharpness} = \frac{1}{n}\sum_{i=1}^{n} S_x * u(i)^2 + S_y * u(i)^2 \tag{10}$$

2.1.12 Exposure quality [5] [#1]

First we need to translate the image in the YCbCr color space and retrieve the luma component Y. Then we compute the skewness of the histogram of Y according to the Equation 11. The skewness is a good indicator of whether the luma is centred and symmetric or not.

$$\text{Skewness} = \frac{\mathbb{E}\left[(X - \mathbb{E}[X])^3\right]}{\mathbb{E}\left[(X - \mathbb{E}[X])^2\right]^{3/2}} \tag{11}$$

2.1.13 Color names [2] [#11]

We use the Discriminative Color Descriptors developed in [2] which is a clustering of all colors of the RGB space into 11 different categories chosen on their discriminative properties after studies with a panel. We then compute the proportion of each of those 11 clusters in the complete image.

2.1.14 Level of details [4] [#1]

A good representation of the level of details is the number of regions left after binarizing the image with a threshold of 0.4 and performing two image opening and closing with a 3x3 matrix of ones. See Figure 11 for an example.

Figure 11. Image after two opening and closing, 23 different regions detected.

2.1.15 Gray-level co-occurrence matrix [5] [#4]

The GLCM counts the number of adjacent pixels with the same value (using 8 bins). This provides information on the texture of the image. We therefore normalize the GLCM and then compute some statistics on this 8x8 matrix such as its entropy, contrast, energy and homogeneity, see Equations 12, 13, 14 and 15.

$$\text{Entropy} = -\sum_i p(i)\log p(i) \tag{12}$$

$$\text{Contrast} = \sum_{i,j} |i-j|^2 p(i,j) \tag{13}$$

$$\text{Energy} = \sum_{i,j} p(i,j)^2 \tag{14}$$

$$\text{Homogeneity} = \sum_{i,j} \frac{p(i,j)}{1+|i-j|} \tag{15}$$

2.1.16 Entropy [#1]

We retrieve the image order information by computing the entropy on the overall image.

2.1.17 Symmetry [5] [#1]

To compute the symmetry feature we first divide the original image into its left half and its flipped right half. We then retrieve the HOG descriptors from both images and return the mean of the difference between the two.

2.1.18 VGG-face [#1024]

We do a forward pass of the pre-trained convolutional neural network VGG-face[4] and retrieve the fc8 layer. Although proper care will have to be taken due to the dimension increase which might lead to overfitting (in any case these features were not helping for predicting aesthetic scores).

[4] http://www.robots.ox.ac.uk/~vgg/software/vgg_face/

2.2. Impact features

For the training set, REGAIND provides "impact features" which basically gives the influence (positive or negative) of the following: background, angle of the face (roll or pan angle), position of the face, sharpness of the face, face exposure, and face expression. These scores where given by humans and a positive impact is between 0 and 1, a negative is between -1 and 0. We tried to exploit this information by first learning a predictor of this score and then using it for the final predictor but this didn't lead to any improvement.

2.3. Results

We trained a SVM with a properly scaled Gaussian Kernel with each of the previous features in order to measure their ability to predict the portraits aesthetic score. We also looked at the noise standard deviation by measuring ρ on white gaussian noise and we obtained $\rho_{noise} \approx 0.01$. If a feature obtain a ρ sufficiently superior to ρ_{noise} then this feature can be considered as meaningful for predicting aesthetic scores.

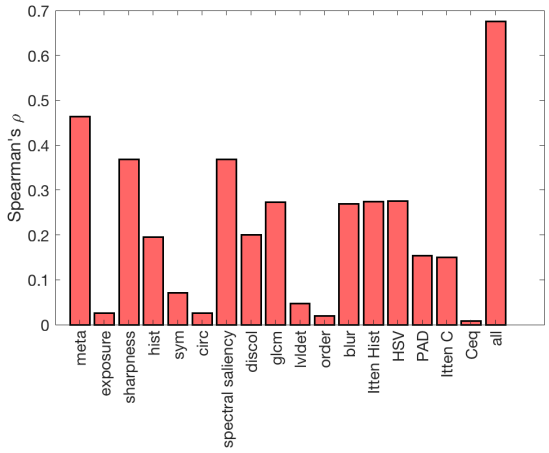

Figure 12. Features influence on Spearman's ρ

All features were more or less helping predicting aesthetic scores. Using all features we obtained $\rho = 0.69$. We can compare the original dispersion obtained by using only meta features and the one by using all features.

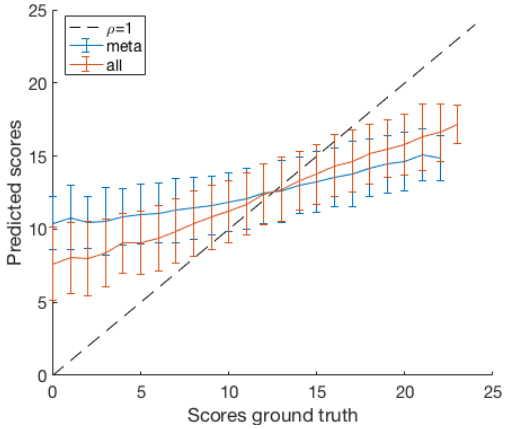

Figure 13. Prediction vs Ground truth with and without new features

On can see on Figure 13 that the introduced features help predicting low and high scores.

3. Learning optimization

In this section we describe our work on the learning part. With the provided and computed features we want to optimize the learning algorithm in order to increase the prediction accuracy.

3.1. SVM

The transformation from the raw images to the vector representation containing all the features can be viewed as an explicit embedding. For this reason we first focused on a linear SVM to test our baseline, testing whether a feature was usefull or not. Then we tried several kernels: linear, gaussian and polynomials. The results computed on the full feature dataset can be found in the table 3[5].

Kernel	Rho
Linear	0.608
Gaussian	0.678
Polynomial degree 1	0.609
Polynomial degree 2	0.635
Polynomial degree 3	0.634
Polynomial degree 4	0.632

Table 3. Kernel performance on full dataset

We thus decided to stick to the gaussian kernel.

3.2. Random forests

We briefly tried implementing random forest but the results were of an order of magnitude lower than a regular linear SVM so we chose not to pursue this implementation.

[5] As we used a random sampling the score may present small variations between each evaluation.

3.3. Neural Network approach

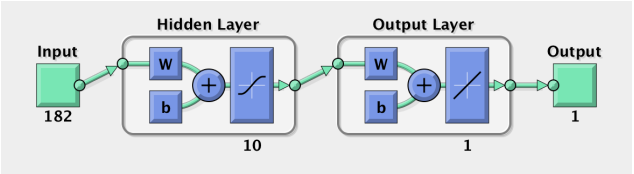

Figure 14. Neural network with 1 hidden layer and 10 neurons

We trained a single hidden layer neural network for regression. The neural network is presented on Fig.14. We divided the training data-set as follows: 70% training, 15% testing and 15% validation.

We used Levenberg-Marquardt backpropagating algorithm to train the neural network and obtained the results presented on Fig.15. Errors distribution are presented on Fig.16.

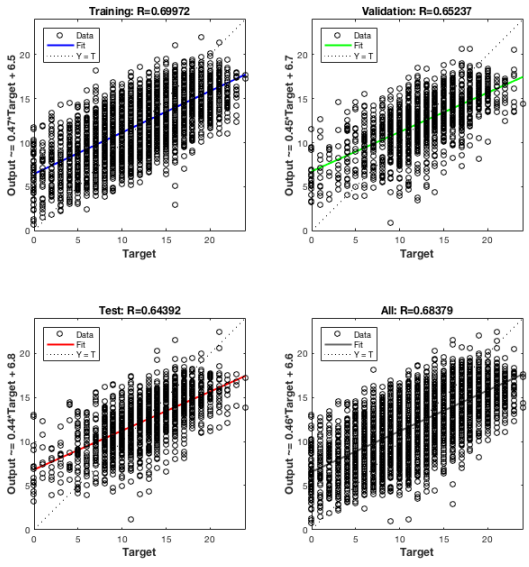

Figure 15. Regression results on the training set

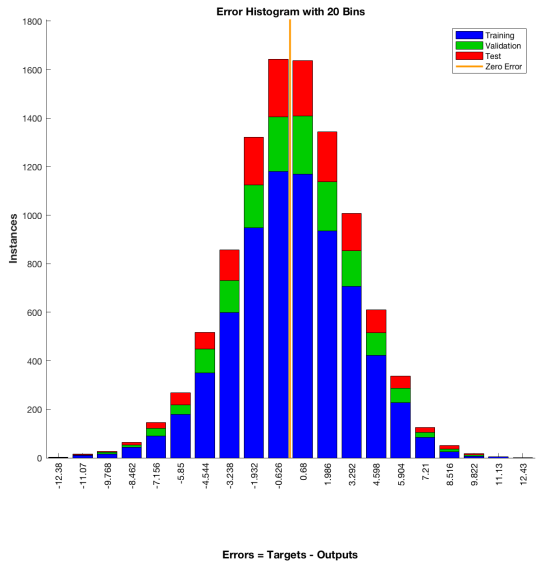

Figure 16. Error histogram results on the training set

We also tried to increase the number of neurons but the results on the validation fold were not better. As this approach did not lead to better accuracy, we decided to focus on simpler and more interpretable learning architectures.

Conclusion

One of the biggest challenge when facing a prediction task based on picture is to know which embedding to chose. The literature is swarming with possible feature giving information on texture, colors, interesting regions, etc. and choosing the relevant features is a challenge in itself.
Besides, even when the best features have been identified, the choice of the learning architecture, whether to implement an SVM, a neural network, or else, can become quite tricky.
Overall we observed several time during the challenge that it is better to start simple and improve from there.

References

[1] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[2] R. Khan, J. van de Weijer, F. S. Khan, D. Muselet, C. Ducottet, and C. Barat. Discriminative color descriptors. In *CVPR*, pages 2866–2873. IEEE Computer Society, 2013.

[3] J. Machajdik and A. Hanbury. Affective image classification using features inspired by psychology and art theory. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 83–92, New York, NY, USA, 2010. ACM.

[4] J. Machajdik and A. Hanbury. Affective image classification using features inspired by psychology and art theory. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 83–92, New York, NY, USA, 2010. ACM.

[5] M. Redi, N. Rasiwasia, G. Aggarwal, and A. Jaimes. The beauty of capturing faces: Rating the quality of digital portraits. *CoRR*, abs/1501.07304, 2015.

[6] H. Tong, M. Li, and C. Zhang. Blur detection for digital images using wavelet transform. *IEEE International Conference on*, 1, 2014.