Image Aesthetics Assessment

Jens Laufer

2018-12-21

I. Definition

Project Overview

The "A picture is worth a thousand words" stresses how important images are in the modern world. The quality of images e.g. influences our decisions in different domains. Especially in eCommerce, where we cannot touch things they are very important. They have therefore a big influence on our product purchasing decisions.

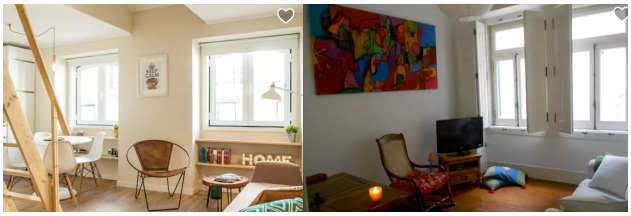


Figure 1: Which room would you book?

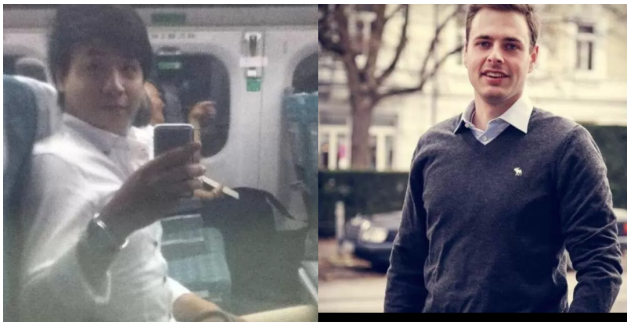


Figure 2: Which guy to date?

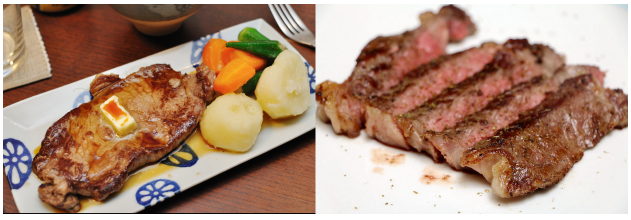


Figure 3: Which guy to date?

The goal of this project is to create a predictor that is able to quantify the aesthetics of images.

Problem Statement

The quantification of image quality is an old problem in computer vision. There are objective and subjective methods to assess image quality. With objective methods different algorithms quantify the distortions and degradations in an image. Subjective methods are based on human perception. The methods often dont't correlate with each other. Objective methods involve traditional rule-based programming, Subjective methods are not solvable this way.

The goal of this project is to develop an subjective method of image quality assessment. As mentioned before this problem cannot be solved with classical programming. But it seems that supervised machine learning is a perfect candidate for solving the problem as this approach learns from examples and it is a way to quantify the ineffeble. A dataset with image quality annotations is a requirement for learning from samples.

Within the machine learning ecosystem Convolutional Neural Networks (CNN) are a category of Neural Networks that have proven very effective in areas such as image recognition and classification. They are inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the human visual cortex.

The subjective quality predictor will be implemented with a Convolutional Neural Network as it seems a good fit to tackle the problem.

To solve the problem these steps are needed:

1. Find a dataset with images with quality annotations
2. Exploratory Data Analysis (EDA) of the dataset, to evaluate the characteristics and suitabilty for the problem space
3. Cleanup and Preprocessing of the dataset
4. Design a architecture for the CNN
5. Training of the CNN
6. Test the model against Benchmarks
7. Analysis of the results

There will be several iterations for the steps 4.-7.

Metrics

The distribution of user ratings will be predicted in the project. From there you are able to predict both a quantitative mean rating, but also a qualitive rating bucket. To capture this three metrics will be used.

Earth Mover's distance (EMD)

The **Earth Mover's Distance (EMD)** is a method to evaluate dissimilarity between two multi-dimensional distributions in some feature space where a distance measure between single features, which we call the ground distance is given. The EMD 'lifts' this distance from individual features to full distributions. It's assumed that a well performing CNN should predict class distributions such that classes closer to the ground truth class should have higher predicted probabilities than classes that are further away. For the image quality ratings, the scores 4, 5, and 6 are more related than 1, 5, and 10, i.e. the goal is to punish a prediction of 4 more if the true score is 10 than when the true score is 5. The EMD is defined as the minimum cost to transport the mass of one distribution (histogram) to the other. (Hou, Yu, and Samaras 2016)(Rubner, Tomasi, and Guibas 2000)(Talebi and Milanfar 2018)

$$EMD(P,Q) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}d_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}}$$

Accuracy

To compare qualitative results the **Accuracy** is used. The Accuracy is the ratio of correct predictions. In this case the ground-truth and predicted mean scores using a threshold of 5, as is standard practice for AVA dataset.

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

$$TP : TruePositives, TN : TrueNegatives, FN : FalseNegatives, FP : FalsePositive$$

II. Analysis

Data Exploration

The AVA (Aesthetic Visual Analysis) image dataset which was introduced by (Murray, Marchesotti, and Perronnin 2012a), (Murray, Marchesotti, and Perronnin 2012b) is the reference dataset for all kind of image aesthetics. The dataset contains 255508 images, along with a wide range of aestehetic semantic and photographic style annotations. The images were collected from www.dpchallenge.com.

In the data exploration phase the data several new technical variables were created to check the dataset for anomalities: image size, filesize, width, height, resolution, aspect ratio. The variables were analysed with a univariate and a bivariate analysis. The Exploratory Data Analysis AVA

Sample rows

image.id	1	2	3	4	5	6	7	8	9	10	rating.mean	rating.sd	rating.mean.bucket
340753	3	2	5	43	100	80	23	10	3	0	5.360595	1.225537	5-6
674342	0	2	4	9	39	56	31	21	15	6	6.355191	1.595610	6-7
737669	8	16	29	55	81	18	6	0	0	0	4.234742	1.300529	4-5
16606	0	1	13	24	46	55	40	14	5	2	5.770000	1.478885	5-6
344449	1	6	17	52	91	47	25	6	1	0	5.044715	1.285485	5-6

Sample images

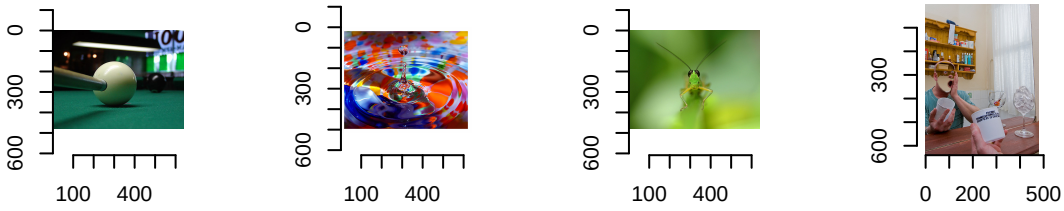


Figure 4: Best rated images

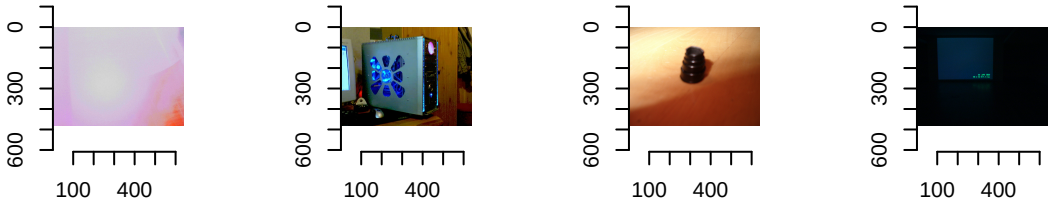


Figure 5: Worst rated images

Desciptive Statistics of number of ratings

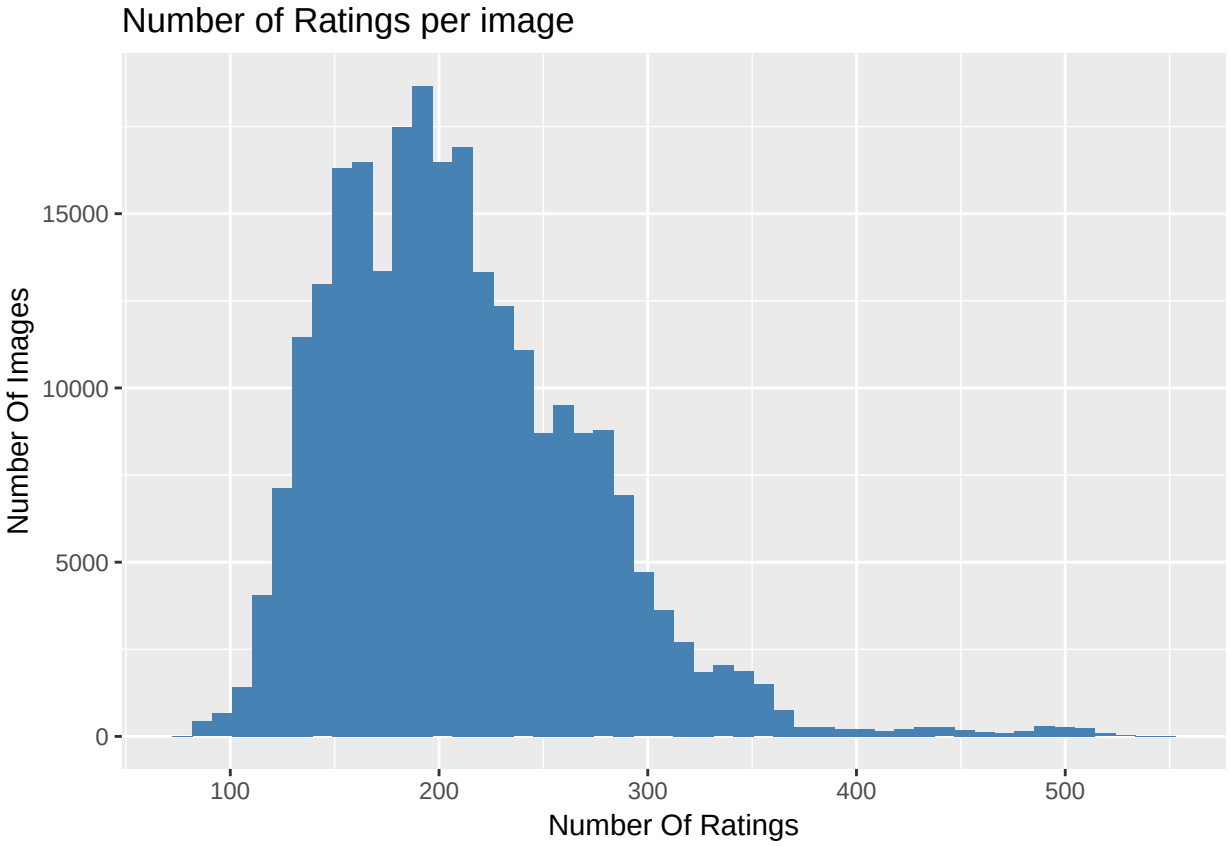
	number
Mean	210.14
Std.Dev	61.51
Min	78.00
Q1	164.00
Median	201.00
Q3	247.00
Max	549.00

Desciptive Statistics of rating.mean

	rating.mean
Mean	5.38
Std.Dev	0.73
Min	1.81
Q1	4.91
Median	5.39
Q3	5.87
Max	8.60

Exploratory Visualization

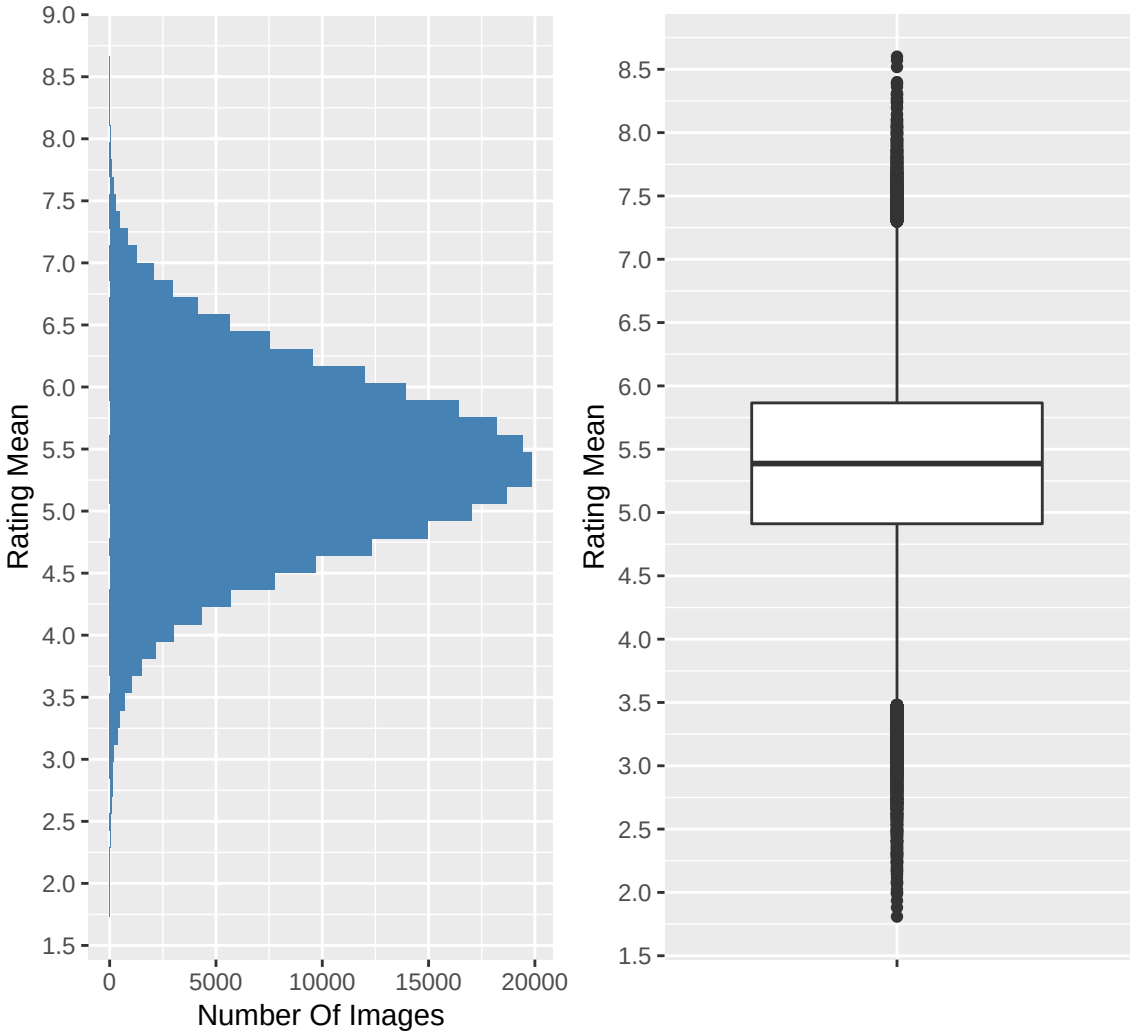
Distribution of number of Ratings



The number of ratings for the images ranges from 78 to 549 with an average of 210 on a scale from 1 to 10.

You can see that all images are rated by a high numbers of raters. This is very import as rating an image by it's aesthetics is very subjective. To level out outliers ratings, a high number of raters is needed.

Distribution of Mean Ratings



It can be seen from the distribution and the descriptive statistics that 50% of images has a rating mean within 4.9 and 5.9 and about 85% are between 3.9 and 6.8. From the boxplot it can be seen that rating means above 7.2 and below 3.5 are outliers in the way that these values are very rare.

This is problematic that our model performance might not sufficient for images with very good and bad quality.

Algorithms and Techniques

Transfer learning involves the approach in which knowledge learned in one task is transferred and used to improve the learning of a related target task.

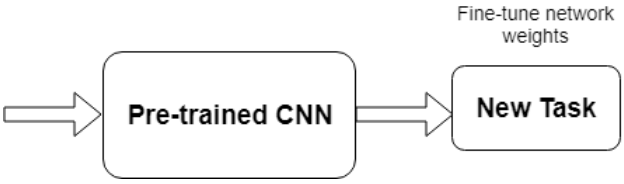


Figure 6: Transfer learning

This approach is used to train the predictor instead of designing a Convolutional Neural Network (CNN) from the scratch, as it saves compute power. The compute power within this project is restricted to a one GPU AWS cloud instances (p2.xlarge) and the dataset is quiet big. Several papers reported that it took days to teach the predictor on their own CNN architecture. The goal in this project is rather to restrict the time to teach the network to a few hours to save cost and try out different options.

Several state-of-the-art image classification applications are based on the transfer learning solutions (He et al. 2016), (Szegedy et al. 2016) Google reported in it's NIMA (Neural Image Assessment) paper high performance on transfer learning based predictors (Talebi and Milanfar 2018)

The goal the is to use the MobileNet architecture with ImageNet weights, and the replacement of the last dense layer in MobileNet with a dense layer that outputs to 10 classes (scores 1 to 10), which form together the rating distribution as suggested by (Talebi and Milanfar 2018)

Benchmark

Accuracies of different models on the AVA dataset are reported in different papers. These accuracies are used for benchmarking the models which are created in this work. The goal is to achieve at least an accuracy of 68% which is above the lower boundary of the relevant papers for image aesthetics.

Model	Reference	Accuracy (2 classes)	EMD
Murray	(Murray, Marchesotti, and Perronnin 2012b)	68.00%	–
Reg	(Kong et al. 2016)	72.04%	–
DCNN	(Lu et al. 2014)	73.25%	–
DMA	(Lu et al. 2015)	74.46%	–
Schwarz	(Schwarz, Wieschollek, and Lensch 2018)	75.83%	–
NIMA(MobileNet)	(Talebi and Milanfar 2018)	80.36%	0.081
NIMA(Inception-v2)	(Talebi and Milanfar 2018)	81.51%	0.050

III. Methodology

Data Preprocessing

The data preprocessing can be devided into two parts: The first part was done during the exploratory data analysis. In this step the following checks and cleanings were performed:

1. Removal of images
   - Several images had to be removed from meta data as they did not exist.
   - Several corrupted images were identified with a script. The corrupted images were deleted from the meta data.
2. Technical image properties were engineered to check image anomalities

   Several technnical image properties (file size, resolution, aspect ratio) were engineered and checked for anomalies. No abnormal images could be identified here with these properties.

The second preprocessing step is performed during training:

1. Splitting of the data into training, validation and test set

   The test data is extracted from the meta data (and from the training data a validation set is extracted

2. Basemodel specific preprocessing were performed

   Each basemodel provided by Keras offers a preprocessing function with specific preprocessing steps for this model. This preprocessing step is applied to ImageGenerator which loads the images for training and model evaluation.

3. Normalization of distribution

   The rating distribution was normalized, because each image was rated by a diffrent number of people.

4. Image resizing and random cropping

   The training images are rescaled to 256 x 256 px and afterwards a randomly randomly performed performed crop of 224 x 224 px is extr In training, input images are rescaled to 256 ï¿½ 256, and then a crop of size 224 ï¿½ 224 is randomly extracted. This is reported to reduce overfitting issues. (**???**)

5. Undersampling of the data

   For earlier tainings sessions the number of images are reduced by cutting the data in 10 rating binsand taking the top n samples of each bin. This is done because of two reasons: As the compute power is limited this reduces training time. Another reason is that the data is unbalanced. There are just a few images with very low and high ratings. It was expected that this reduces the effect of overfitting to the images around the most common ratings.

Implementation

The goal was to create a clear training script which can be parameterized from outsite for triggering the different trainings. To reduce the lines of code of this training script it orchestrates the building blocks of the training with a pipeline script.

1. All needed libraries are identified and put into a requirements.txt
2. An internal library to download the AVA images and the meta data are implemented.
3. A training script was created with building blocks for training (loading data, preparing data, train, evaluate)
4. Building blocks of the training script are moved to pipeline script. The scripts saves different artifacts: Model architecture, Model weights, training history, time for training, training visualization
5. A model class is created, which encapsulates the basemodel and top model and offers helper functions to change optimizer and freeze layers on the fly
6. The EMD loss function is created
7. The image generator is created for loading the images and perform the preprocessing of the images
8. Several helper functions for model evaluation

The actual training is performed in 2 Steps:

1. Base model weights are freezen and just the top model is trained with a higher learning rate
2. Base model weights are unfreezen and the full network is trained with a lower learning rate

Model design of the CNN

The model consists as mentioned before of two parts. The base model is unchanged apart from the first layers which is removed. The model is initialized with the ImageNet weights. The ImageNet project is a large visual database designed for use in visual object recognition software research. The weights for this dataset is used as the images are similiar to the ones in the AVA dataset. For the base model the MobileNet architecture as this network is smaller to other networks and suitable for mobile and embedded based vision applications where there is lack of power. (Howard et al. 2017)

The top model consists of two layers. The first layer is a dropout layer to reduce overfitting, followed by dense layer with a output size of 10 with a softmax activation to predict the distribution of ratings. A Adam optimizer with different learning rates and learning rate decays is used for training.

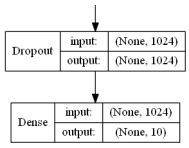


Figure 7: Design of top model: Dropout Layer for avoiding overfitting, Dense layer with 10 output classes

Refinement

Several parameters were used for model refinement:

- Learning rate for dense layers and all layers
- Learning rate decay for dense layers and all layers
- Number of epochs for for dense layers and all layers
- Number of epochs for for dense layers and all layers
- Number of images per rating bin used for training
- Dropout ratio in dropout layer in top model

The training is done in iterative way: First the model is trained with a very few samples and the default values for the parameters above. Then the model is trained with more samples and fined tuned of the parameters. After the model is trained the loss value and the accuracy are calculated. The accuracy is then compared against the accuracy scores from the paper (see section Benchmarks) till a sufficient model accuracy was reached.

The training process is evaluated with plots of the loss on the training and validation set, to check if everything works well and to optimize the learning process.

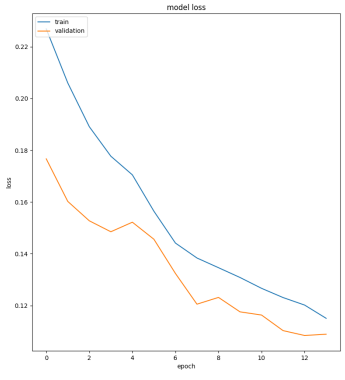


Figure 8: The plots for training history is used to find the best number of epochs for the two learning phases. During phase 1 validation loss flattens at epoch 5 (4 in plot) and in phase 2 the val loss flattens at epoch 8 (12 in plot)

IV. Results

Model Evaluation and Validation

From the different models was chosen as it's EMD loss value is the lowest and it's accuracy is the highest among all models on the test set. The results are trustful, as the test set is the "official" test set for AVA and the model never saw these images during test or validation.

model	acc	emd
model8	75.22	0.094
model6	74.89	0.117
model9	74.85	0.095
model5	73.94	0.121
model7	70.42	0.105

The best model is based on the MobileNet architecture and the these parameters are used. All these parameters seem reasonable:

Dropout	n training samples	lr(dense)	lr(all)	lr decay(dense)	lr decay(all)	Epochs (dense)	Epochs (all)
0.75	13914	0.001	3e-05	0	2.3e-05	5	9

It can be seen from the figure below, that the distribution of the ground truth mean ratings and the predicted mean ratings are very similiar for the best model. The model works well for mean ratings between 3.5 and 7.5. Ratings below or above these boundaries are not covered by model. This due the fact, that there are not many images with very high and low ratings. So model is not capable to rate these extreme outliers correctly, because of the lack of examples.

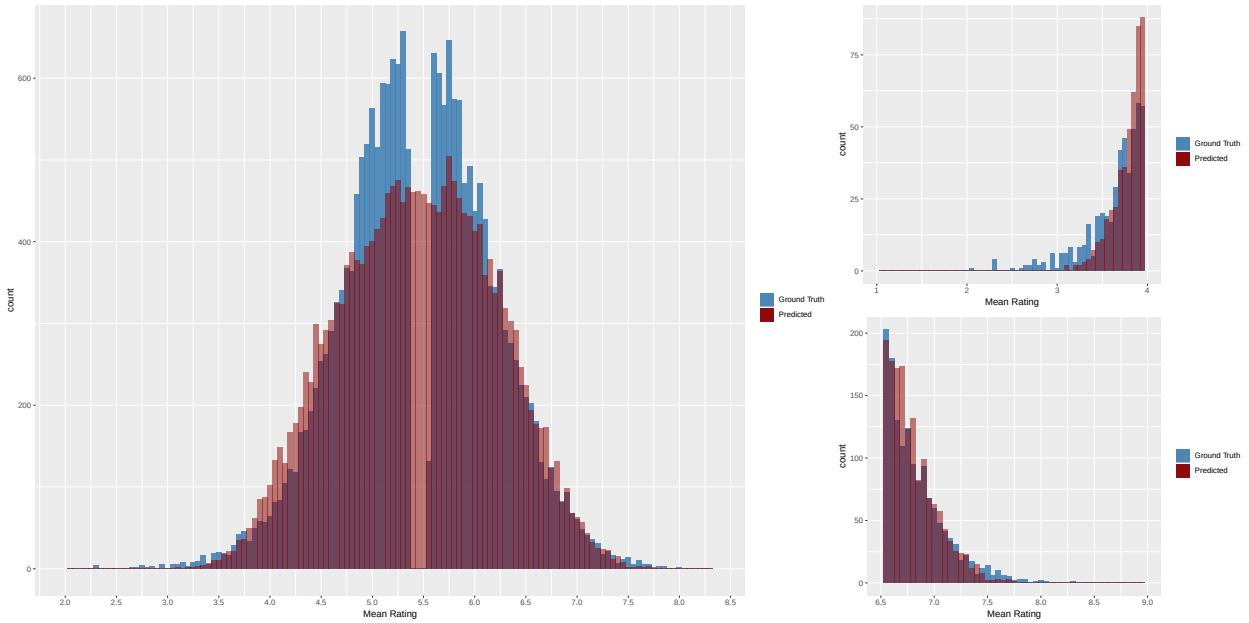


Figure 9: Big figure: Distribution of pedicted mean ratings and ground truth rating on test set. Small figures: Distribution on lower and upper end on test set.

Justification

In comparison to the benchmarks the model shows an moderate accuracy on the reference test set for AVA which is used throughout all models from the papers.

The result are quite impressive, as the model was trained with just 13914 images. The models in the papers were trained with the full training set.

Model	Reference	Accuracy (2 classes)	EMD
Murray	(Murray, Marchesotti, and Perronnin 2012b)	68.00%	–
Reg	(Kong et al. 2016)	72.04%	–
DCNN	(Lu et al. 2014)	73.25%	–
DMA	(Lu et al. 2015)	74.46%	–
My Model	–	**75.22%**	**0.094**
Schwarz	(Schwarz, Wieschollek, and Lensch 2018)	75.83%	–
NIMA(MobileNet)	(Talebi and Milanfar 2018)	80.36%	0.081
NIMA(Inception-v2)	(Talebi and Milanfar 2018)	81.51%	0.050

V. Conclusion

Free-Form Visualization

For a final quick and dirty test the images, which are not part of any dataset from the beginning are rated with the model.

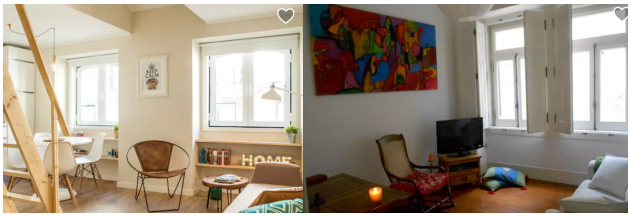


Figure 10: Left Image: 4.23 Right image: 3.91

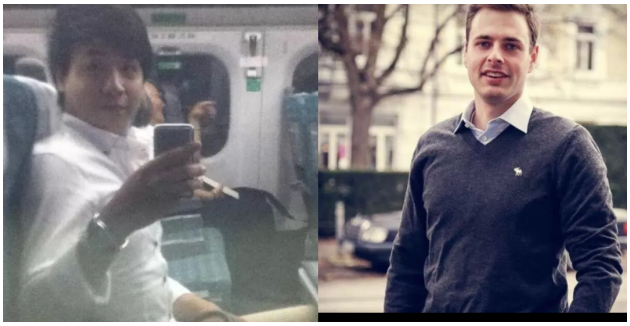


Figure 11: Left Image: 3.27 Right image: 4.00

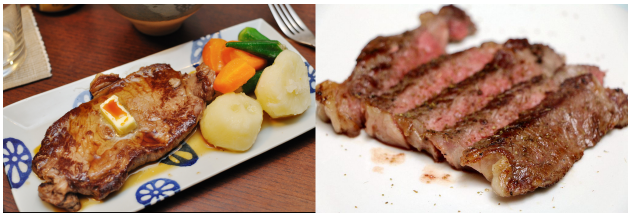


Figure 12: Left Image: 3.98 Right image: 4.67

It can bes seen, that the images which we as a human beeing would rate better are rated better by the model, although the food images are almost the same quality.

Reflection

The process used for this project can be summarized using the following steps

1. An relevant problem was found
2. A research for relevant papers was done
3. Datasets for the problem were researched, analyzed and the best suitable dataset was selected
4. The dataset was cleaned
5. Model benchmarks were extracted from papers
6. The technical infrastructure for the project was set up
7. Models were trained and finetuned and checked against the benchmarks, till a good enough model was found, that solves the problem

The project was very challanging for me as I had limited compute power and the dataset is very large. Till the end I was not able to train the models on the full training set as there were always problems like running out of memory and Keras and Tensorflow specific problems, I was at some point stuck, as the models performed badly. After doing an additional research round I found the the Nima paper from Google, which was so brandnew that it wasn't published when I started the project in July. The insights from the paper were a

breakthrough, especially the usage of the Earth Movers Loss and the usage of the MobileNet archtitecture for the base model. I am very proud that I could get a accuracy which was within the boundaries of the relevant papers and mastered a topic that is very hot in the moment. The project was big fun to always switch from R, which I prefer for data analysis and Python which I prefer for the rest.

Improvement

It's very interesting that I did achieve a accuracy within the boundaries with my undersampling strategy, which was half born out of need. Even after doing the undersampling of the data the distribution of the ratings is unbalanced.

A strategy to even perform better would be to do image augementation on the underrepresented rated images. This is not so easy, as not every kind of image augmentation can be used e.g darkening an image may effect the aesthetics of the image. Another interesting approach would be to generate images with very high rating with GANs (generative-adversarial-networks).

For sure having a high performing environment with that you were able to do training on the whole dataset would be a improvement for further optimization.

Another improvement for the project would be to containerize the whole process with Docker and Docker NVIDIA. The goal would be to have a docker image that automatically downloads the data, does the preprocessing of it, does the training and stops the container after training. Within this project this is done with anaconda environments, which is less than ideal in my eyes. I had to always switch from my local environment to the AWS cloud instance, lost time as the environments are not the same. A Docker environment could be also optimized with reusable elements for other Deep Learning projects.

VI. References

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. "Deep Residual Learning for Image Recognition." In *Proceedings of the Ieee Conference on Computer Vision and Pattern Recognition*, 770–78.

Hou, Le, Chen-Ping Yu, and Dimitris Samaras. 2016. "Squared Earth Mover's Distance-Based Loss for Training Deep Neural Networks." *arXiv Preprint arXiv:1611.05916*.

Howard, Andrew G, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. "Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications." *arXiv Preprint arXiv:1704.04861*.

Kong, Shu, Xiaohui Shen, Zhe Lin, Radomir Mech, and Charless Fowlkes. 2016. "Photo Aesthetics Ranking Network with Attributes and Content Adaptation." In *European Conference on Computer Vision*, 662–79. Springer.

Lu, Xin, Zhe Lin, Hailin Jin, Jianchao Yang, and James Z Wang. 2014. "Rapid: Rating Pictorial Aesthetics Using Deep Learning." In *Proceedings of the 22nd Acm International Conference on Multimedia*, 457–66. ACM.

Lu, Xin, Zhe Lin, Xiaohui Shen, Radomir Mech, and James Z Wang. 2015. "Deep Multi-Patch Aggregation Network for Image Style, Aesthetics, and Quality Estimation." In *Proceedings of the Ieee International Conference on Computer Vision*, 990–98.

Murray, Naila, Luca Marchesotti, and Florent Perronnin. 2012a. "AVA: A Large-Scale Database for Aesthetic Visual Analysis." https://github.com/mtobeiyf/ava_downloader.

———. 2012b. "AVA: A Large-Scale Database for Aesthetic Visual Analysis." In *Computer Vision and Pattern Recognition (Cvpr), 2012 Ieee Conference on*, 2408–15. IEEE.

Rubner, Yossi, Carlo Tomasi, and Leonidas J Guibas. 2000. "The Earth Mover's Distance as a Metric for Image Retrieval." *International Journal of Computer Vision* 40 (2). Springer: 99–121.

Schwarz, Katharina, Patrick Wieschollek, and Hendrik PA Lensch. 2018. "Will People Like Your Image?

Learning the Aesthetic Space." In *Applications of Computer Vision (Wacv), 2018 Ieee Winter Conference on*, 2048–57. IEEE.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. "Rethinking the Inception Architecture for Computer Vision." In *Proceedings of the Ieee Conference on Computer Vision and Pattern Recognition*, 2818–26.

Talebi, Hossein, and Peyman Milanfar. 2018. "Nima: Neural Image Assessment." *IEEE Transactions on Image Processing* 27 (8). IEEE: 3998–4011.