Personalized Image Aesthetic Prediction

An T. Nguyen Josh Kelle
Department of Computer Science
University of Texas at Austin

Abstract

Computing the aesthetic quality of an image has gained attention in recent years. One application is mining good quality images to show users. However, people have different opinions on image quality and aesthetic. We design a probabilistic graphical model that takes into account personal taste to predict personalized image aesthetic scores on a scale of 1 to 5. We experiment with different image features and evaluate the system using a large real-world dataset and find strong improvement over competitive baselines. We also show that our approach is particularly useful to users who annotate only a small number of image aesthetic scores.

1. Introduction

The number of images on the internet is massive and growing rapidly. Many internet companies such as Pinterest and Facebook are becoming more reliant on using images to engage users. It is imperative to automatically identify images that users will find appealing. Recent work has focused on building recommendation systems that predict image quality. However, users don't always find the same images equally aesthetically pleasing. Therefore it's important to design a personalized model that takes into account users' unique tastes.

Kong et al. (2016) [7] have developed the Abstract Aesthetics and Attribute Database (AADB) to foster research in exactly this area. In this dataset, annotators provide a 1-5 start rating of the overall aesthetic score for thousands of real-world images. Additionally, workers annotate the presence or absence of certain image attributes which have historically been used to judge image quality, such as good lighting, rule of thirds, etc. Figure 2 shows some examples of images and their aesthetic ratings.

Multiple workers annotate each image. It is often the case that workers give different scores to the same image. This highlights the importance of personalization in the field of image aesthetic prediction. Our goal is to design a system that learns users' unique taste. A challenge that comes with personalization is the fact that many users only annotate a small number of images. Figure 1 shows the distribution of the number of images rated versus worker index. The ideal prediction system will require only minimal input from the user.

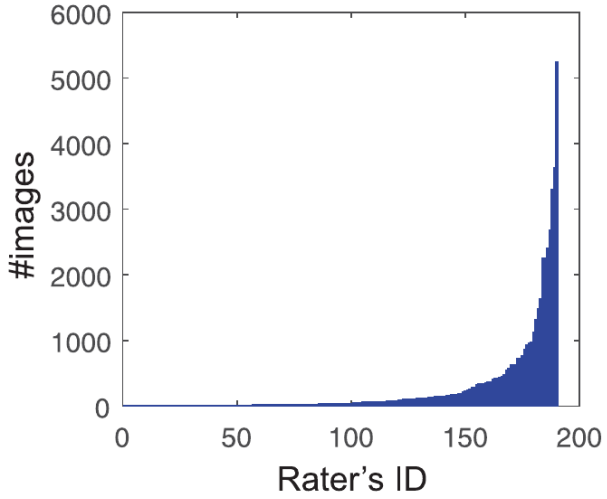

Figure 1: This plot shows the number of images each worker annotated. There is a long tail of workers who annotated very few images (based on Figure 6 in Kong et al. (2016)).

Our method requires (1) appropriate image features and (2) a good prediction model. We extract features from fully connected layers of a convolutional neural network pretrained on ImageNet and fine tuned on AADB. The architecture is described in more detail in Section 3.1. These features are used with a probabilistic graphical model that effectively clusters users into topics. This model allows us to predict the overall aesthetic score for a given image and user, even if the image has not been rated by other users before. This model also allows us to achieve better results than state of the art methods, especially for users with few ground truth ratings to train on. Section 3.2 describes this graphical model in more detail.

(a) Ratings $\in (2, 3, 3, 4, 4)$

(b) Ratings $\in (5, 5, 5, 5, 5)$

(c) Ratings $\in (1, 1, 3, 4, 5)$

(d) Ratings $\in (1, 1, 1, 2, 2)$

Figure 2: Some images in the AADB dataset with their five ratings.

2. Related Work

Kong et al. (2016) [7] propose a method for predicting image aesthetics from content and certain photometric image attributes. They created the AADB dataset to train and evaluate their model. Their approach is to fine tune a modified version of AlexNet [10] (see Figure 3) on AADB to predict overall image aesthetic rating. However, this prediction is global, i.e. does not depend on the user.

Lu et al. (2015) [11] use CNNs to classify real-world images into two categories, either low-aesthetic or high-aesthetic images. Dhar et al. (2011) [2] also train a CNN to classify images into these two categories, but they instead use high-level human describable image attributes. While we also use CNN features, our model performs regression to predict a score in the range of 1-to-5.

There is also previous work on the closely related problem of image recommendation. Fan et al. (2009) [3] develop an interactive interface for users to explore and search images from Flickr. Geng et al. (2015) [4] propose a deep model that learns a unified feature representation for both users and images. These two works are similar to our work in that we share personalization as a goal. However, our work is different because our goal is to predict image aesthetic ratings.

With respect to personalized predictions, Kovashka and Grauman (2013) [8] propose an adaptive SVM and Rank SVM approach to predict user attribute labels and apply to image search. Their following work [9] improve on that by clustering users into 'schools of thought' and adapt the predictions to each school. We use this as a baseline to compare our method against.

Modeling annotators is a core problem in crowdsourcing. Most previous work in this space focuses on improving the aggregated labels [18, 14, 17], assuming that the difference in labels for different users is 'noise'. This is

in contrast with our work where we model individual user. Our method builds on the model proposed by Nguyen et al. (2016) [12] for identifying 'unreliable ratings' and 'unreliable crowd workers'. Here, we assume that all ratings and users are reliable and try to predict the personalized ratings for unseen images. This is a reasonable assumption since Kong et al. (2016) has experiments showing that the ratings they collected are highly reliable. Technically, we generalize Nguyen et al. (2016)'s Beta-Bernoulli model for two topics to a Dirichlet-Categorical model for an arbitrary number of topics.

3. Method

As mentioned, our method is two-fold. First we find an appropriate image representation, then we use this representation along with per-user ratings to train a graphical model. Section 3.1 discusses feature extraction in detail. Section 3.2 describes the graphical model in detail.

3.1. Features Extraction

We experiment with several different image features, all of which are extracted from fully connected layers of convolutional neural networks. We used off-the-shelf models of the popular CNNs VGG16 [15], ResNet152 [5], and GoogLeNet [16], all pre-trained on ImageNet. We also test features from these networks after fine tuning on AADB star ratings. Finally, we extract features from a network provided by Kong et al.

We used Caffe [6] to fine tune VGG16 and ResNet152 to predict average aesthetic rating for each image. We use regression loss

$$Loss_i = \|y_i - f(x_i)\|_2^2$$

We reduced the learning rate by a factor of 10 for later convolution layers. For earlier convolution layers, we dropped the learning rate to zero to prevent these weights from changing. We also reduced the dropout percentage in final layers.

Finally, we tried extracting features from a network provided by Kong et al. This network, shown in Figure 3, is based on AlexNet but has some important additional layers. There are parallel fully connected (fc8) and classification layers for each of the binary image attributes. The output vectors of all fc8 layers are then concatenated and fed to a final fully connected layer (fc10). The output of this final fc10 layer is used to predict a single overall aesthetic score.

3.2. Rating Prediction

Given the extracted features for each image, we next build a model that predicts the ratings that a particular user will give to some new images. Two obvious baselines are: (1) A single model for all users and (2) A model for each user. In the former, we ignore the difference in the preference of the users to train the model on all the rating data. In the latter, for each user, we train a model on his (or her) ratings, ignoring the ratings of other users. Although the model here can be any regression method, we consider Linear Regression (LR) without features engineering techniques such as kernels. Since the features are extracted from complex neural network architectures, we expect that further features engineering is not necessary and LR to be sufficient. The two baselines are two extremes and a 'middle ground' is often desired. An easy way to do that is to use the idea of domain adaptation: consider all ratings to be the source domain and each user's ratings to be a target domain. We first train a single model for all users then adapt that model to each user. There are a large number of domain adaptation algorithms. Here, we consider 'Adaptation by Weighting', a simple, efficient and well-performed method in our experiments. In this method, for each user, we train a weighted predictive model with higher weights for ratings given by that user. The weights are set using the Validation set. We will use 'Adaptation by Weighting' as a baseline and also as a component in our proposed method, which we now present.

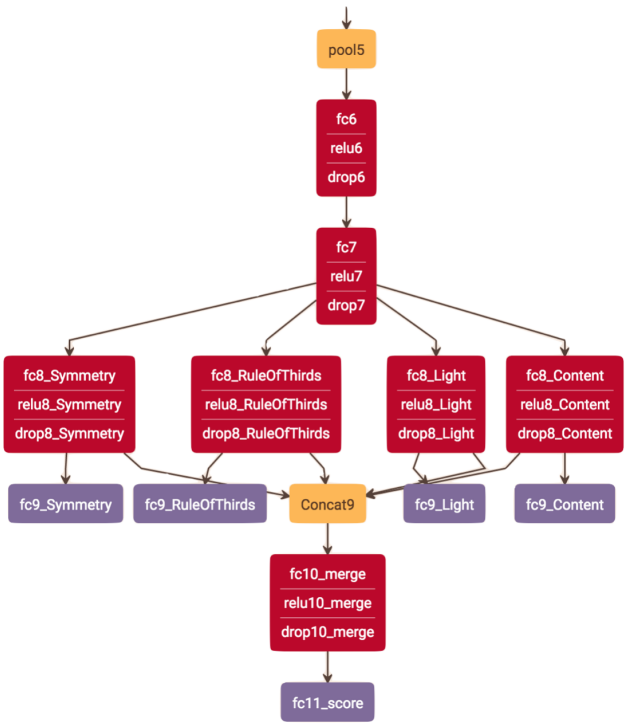

Figure 3: Network architecture provided by Kong et al. Each photographic attribute has a fully connected layer. Some attribute layers have been removed for simplicity. The top of the network is also not shown for simplicity. It is the same as AlexNet.

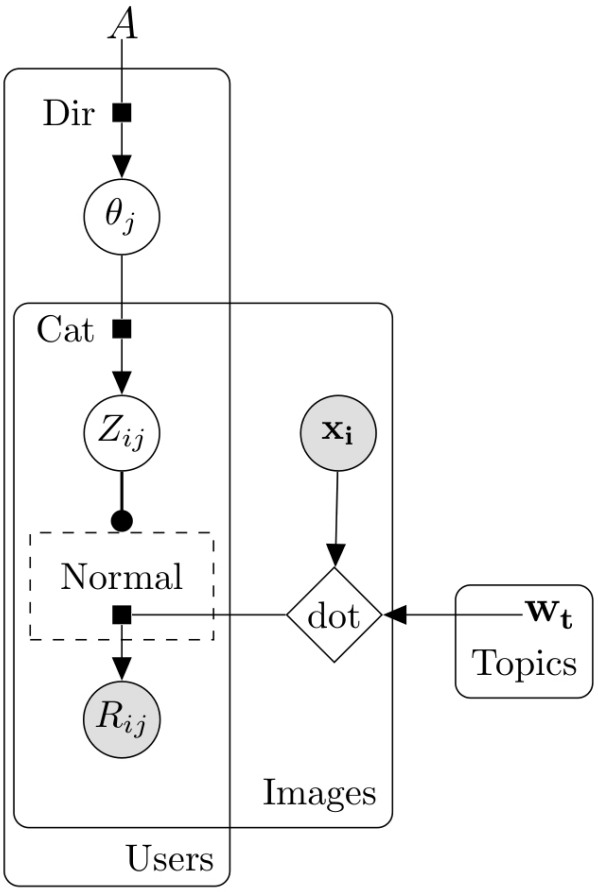

Figure 4: The factor graph for our method. Circles represent random variables where shaded ones are observed. Plates represent repetitions. The dotted plate is a gate where the value of Z_{ij} selects which Normal distribution will generate R_{ij}. Note that we do not have the ratings for all user-image pairs.

Model Definition: The method we propose is a Probabilistic Graphical Model, which defines a joint distribution over some variables of interest. Let R_{ij} be the rating for image i given by user j, $\mathbf{x_i}$ be the features of image i. We assume that for each rating R_{ij}, there is a latent variable Z_{ij} indicating the topic that this rating belongs to, where topics are LR models. Specifically, let $\mathbf{w_t}$ be the parameter for topic t, we have:

$$R_{ij}|Z_{ij} = t \sim \text{Normal}(\mathbf{w_t} \cdot \mathbf{x_i}, \sigma_t^2) \tag{1}$$

which means that given Z_{ij} indicates the topic t, R_{ij} are generated from a Normal distributions with mean $\mathbf{w_t} \cdot \mathbf{x_i}$ and variance σ_t^2. The topic indicator Z_{ij} are generated from a Categorical distribution with parameters θ_j. θ_j are then the topic distribution for user j and has a conjugate Dirichlet Prior:

$$Z_{ij} \sim \text{Cat}(\theta_j) \tag{2}$$
$$\theta_j \sim \text{Dir}(A) \tag{3}$$

where θ_j and A are both vectors of T dimensions, where T is the number of topics. Figure 4 is the factor graph for our method.

Learning: We learn the parameters $\mathbf{w_t}$ using the Expectation Maximization (EM) algorithm [1]. This algorithm iterates between the E step and the M step until convergence (or reaching the maximum number of iterations). In the E step, it infers the posterior distribution over the hidden variables. In the M step, it fits the parameters $\mathbf{w_t}$ by maximizing the likelihood under the expectation of the posterior inferred in the last E step. For efficiency, we treat only Z_{ij} as hidden variables and $\theta_j, \mathbf{w_t}$ and σ_t^2 as parameters. The prior parameters A is used as a smoothing constant for estimating θ_j.

In the E step, we evaluate the posterior:

$$p_{ijt} = \Pr(Z_{ij} = t|R) \propto \text{Cat}(Z_{ij} = t|\theta_j) \tag{4}$$
$$\text{Normal}(R_{ij}|\mathbf{w_t} \cdot \mathbf{x_i}, \sigma_t^2) \tag{5}$$

In the M step, we first estimate θ as by normalizing the posterior p_{ijt}, smoothed by the prior A.

$$\theta_{jt} = \frac{\sum_i p_{ijt} + A_t}{\sum_{i,t} p_{ijt} + A_t} \tag{6}$$

To estimate $\mathbf{w_t}$ and σ_t^2, we simple train a weighted LR model for each topic using the posterior p_{ijt} as the weights. Our implementation trains the LR model using Stochastic Gradient Descent, provided by the package Scikit-Learn [13].

Since the EM algorithm only provides locally optimized solutions, a good initialization is important. We initialize the topics distribution for each user θ_j using the following heuristic. For each user j with more than K ratings, we create a topic and set θ_j to be a one-hot vector (with 0 everywhere and 1 in the index of the topic). The idea is that for each user with 'enough' ratings, we create a topic specifically for learning his (or her) preference. For each user j with less than K ratings, we set the vector θ_j to be uniform over the topics. The value of K is set using the validation set. We also have an experiment to analyze the sensitivity of our method to the choice of K.

4. Evaluation

What experiments did you run to evaluate the idea? What is the main purpose of each experiment, and what can

you conclude from the results? Can you make any comparisons with alternative approaches? Provide figures and examples as appropriate. Also comment briefly on what software, libraries, datasets, etc. you used. The analysis and your interpretation of the results are most important for this part of the paper. Be sure to answer not only what you did, but also why, and what the outcomes indicate.

4.1. Setup

Data: The AADB dataset released by Kong et al. (2016)[7] contains nearly 10,000 images with 5 ratings each. We split the data into 60% Train, 20% Validation and 20% Test. We use the Validation set to develop our method and set some hyper-parameters. We fix all the decisions and parameters before running on the Test set (which we did only once).

Metric: Each method that we evaluate is given the images, the ratings, and the user identities. At test time, the method is given a set of new images and user IDs and must predict the personalized ratings. We compare the predicted ratings to the true ratings and use the Root Mean Squared Error (RMSE) metric.

Baselines: We compare our method to the following baselines. (1) A single LR model for all users (Single). (2) An LR model for each user (User). (3) Adapt the Single LR model to each user by weighting the ratings for the users (Adapt by weight). Ratings by the user we want to adapt to have higher weights, which are set by the Validation set. (4) Shades of Meaning [9]. For the last, we follow the authors to implement the method using the same Bayesian Matrix Factorization package by Xiong et al. (2010) [19] with the same parameters and cluster the users using K-means with K selected by silhouette coefficient to discover 'schools of thoughts', as described in the paper. For adapting to those 'schools of thoughts', since our task is regression, we use Linear Regression with adaptation by weight (similar to the baseline (3)), instead of the adaptive support vector machine method [20] used by the authors.

All the baselines use the best features that we get in the following comparison.

4.2. Comparison of the extracted features

In the first experiment, we compare the images features that we extract using different neural network architectures. The purpose of this is to identify the best features for our task. The first in our comparison is the CNN provided by Kong et al. (2016) [7], which is an AlexNet [10] pre-trained on ImageNet and fine-tuned on AADB. The fine-tuning is done with both the 1-to-5 rating and the image aesthetic attributes (lighting, color, etc). Although AlexNet has achieved good performance on many vision tasks, recent work has proposed other CNN architectures with favorable results, including GoogleNet [16], VGG [15] and ResNet [5]. We compare these networks using both the original version (pre-trained on ImageNet) and the those fine-tuned on the AADB dataset. Our fine-tuning uses only the 1-to-5 ratings.

Features	RMSE
AlexNet fine-tuned	0.831
GoogleNet	0.986
VGG	0.980
VGG fine-tuned	0.965
ResNet	0.998
ResNet fine-tuned	1.039

Table 1: Root Mean Squared Error (RMSE) of the different features with predictions by 'Adapt by weight'. The results are on the Validation set.

Method	mean RMSE	std RMSE
Single	0.9369	0.0007
User	1.0487	0.0026
Adapt by weight	0.8392	0.0012
Shades of Meaning	0.9376	0.0048
Ours	0.8355	0.0012

Table 2: Comparison of our method against baselines by Root Mean Squared Error (RMSE) on the Test set. We report the mean and standard deviation in 10 bootstrap resamples of the training data.

To compare the features, we use 'Adapt by weight' to train a predictive model for the user ratings. The choice of 'Adapt by weight' is due to its good performance in our preliminary experiments. The results, in RMSE in the Validation set, are in Table 1. AlexNet performs surprisingly well, with the lowest RMSE, which suggest that using aesthetic attributes for fine-tuning is important. Our fine-tuning, using only the 1-to-5 ratings, improve VGG but not ResNet. We hypothesize that the very deep structure of ResNet is a difficulty. In the next section, we use the features extracted from the fine-tuned AlexNet for comparing all the prediction models.

4.3. Comparison of the prediction methods

In Table 2, we compare our prediction method to the baselines. The results are on the Test set, which we run after fixing all hyper-parameters. We report the means and standard deviations in 10 bootstrap re-samples of the training data to assess whether the difference is significant. For the first two baselines, we see that Single performs better than User. Single has the advantage of training on all the ratings but its predictions are not personalized. User is personalized but has very little data to train models for the users who provided a few ratings, which probably explains its poor performance. 'Adapt by weight' is a simple

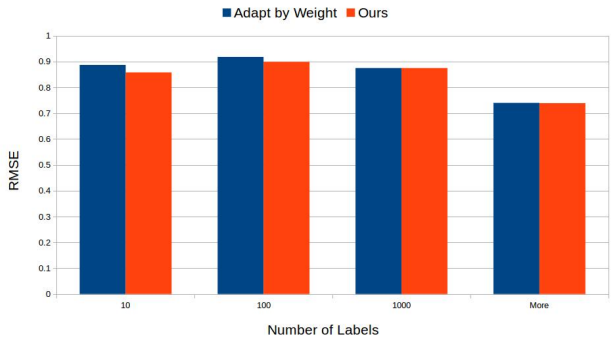

Figure 5: RMSE for Adapt by Weight and our method in four groups of users: (1) those with 10 ratings or less, (2) those with more than 10 but at most 100 ratings, (3) those with more than 100 but at most 1000 ratings and (4) those with more than 1000 ratings.

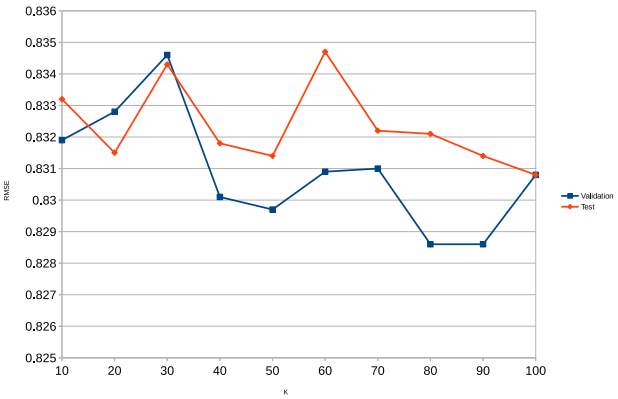

Figure 6: RMSE of our method with varying K in the Validation and Test set. We see that the numbers vary in a range of less than 0.007

way of getting the best of both: train on all the data and personalize to each user. This baseline achieves significant improvement. 'Shades of Meaning' has some improvement over User but does not seem to work well. We hypothesize that this method has clustered users with different preference into the same 'school of thought' and therefore fails to provide a more personalized prediction. The clustering is done by K-means using user features inferred by Bayesian Matrix Factorization. For users with very few ratings (e.g. 10 or less), a feature vector with $D = 50$ dimensions probably contains more noise than signal. Furthermore, this user clustering is based only on their ratings and is independent of the image features.

We find that our method has the best performance. The small values of the standard deviations suggest that this improvement is significant: our mean RMSE is more than 3 standard deviations away from the second-best Adapt by weight. To further understand the improvement, we plot the RMSE for our method and the second-best baseline 'Adapt by weight' with a varying number of ratings the users have provided in Figure 5(see the caption). We observe that most of the improvement is for users with very few labels (100 to 10 or less). One may ask how our method can learn their preference with so few ratings. Recall that in the E step, for each rating, we calculate the posterior distribution over the topics. These posterior distributions are used to discover the preference for each user, by estimating his (or her) topic distribution. Our flexible approach of using a 'soft clustering' for both users and ratings is responsible for the improvement, in our opinion.

In an additional experiment, we study the sensitivity of our method with respect to the hyper-parameter K (the ratings threshold for our heuristic in initializing the EM algorithm). In Figure 6, we report our performance in RMSE with varying Ks. We can see relatively small variations, suggesting that our method is not sensitive to K.

5. Conclusion

We have presented our approach to predicting personalized image aesthetic ratings. The approach consists of features extraction using neural network architectures and prediction model that discovers user preferences and represents them in terms of topics.

Experiments in a real dataset of aesthetic ratings by users in Amazon Mechanical Turk show our strong improvement over competitive baselines, especially for users with few ratings. Other experiments also suggest that the improvement is significant and that our method is not sensitive to an important hyper-parameter. These are encouraging results and we expect them to generalize to other settings and datasets.

In future work, we plan to use the same approach for personalized attributes classification and investigate active learning strategies in picking images for users to rate as well as visualizing the learned topics.

References

[1] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[2] S. Dhar, V. Ordonez, and T. L. Berg. High level describable attributes for predicting aesthetics and interestingness. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1657–1664. IEEE, 2011.

[3] J. Fan, D. A. Keim, Y. Gao, H. Luo, and Z. Li. Justclick: personalized image recommendation via exploratory search from large-scale flickr images. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(2):273–288, 2009.

[4] X. Geng, H. Zhang, J. Bian, and T.-S. Chua. Learning image and user features for recommendation in social networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4274–4282, 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[7] S. Kong, X. Shen, Z. Lin, R. Mech, and C. Fowlkes. Photo aesthetics ranking network with attributes and content adaptation. *arXiv preprint arXiv:1606.01621*, 2016.

[8] A. Kovashka and K. Grauman. Attribute adaptation for personalized image search. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3432–3439, 2013.

[9] A. Kovashka and K. Grauman. Discovering attribute shades of meaning with the crowd. *International Journal of Computer Vision*, 114(1):56–73, 2015.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] X. Lu, Z. Lin, X. Shen, R. Mech, and J. Z. Wang. Deep multi-patch aggregation network for image style, aesthetics, and quality estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 990–998, 2015.

[12] A. T. Nguyen, M. Halpern, B. C. Wallace, and M. Lease. Probabilistic modeling for crowdsourcing partially-subjective ratings. In *Proceedings of The Conference on Human Computation and Crowdsourcing (HCOMP)*, 2016.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[14] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322, 2010.

[15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[17] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*, pages 155–164. ACM, 2014.

[18] P. Welinder, S. Branson, P. Perona, and S. J. Belongie. The multidimensional wisdom of crowds. In *Advances in neural information processing systems*, pages 2424–2432, 2010.

[19] L. Xiong, X. Chen, T. kuo Huang, J. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization.

[20] J. Yang, R. Yan, and A. G. Hauptmann. Adapting svm classifiers to data with shifted distributions. In *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*, pages 69–76. IEEE, 2007.