

Learning to Detect Multiple Photographic Defects

Ning Yu¹ Xiaohui Shen² Zhe Lin³ Radomír Měch² Connelly Barnes¹
¹University of Virginia ²Adobe Research
 {ny4kt, connelly}@cs.virginia.edu {xshen, zlin, rmech}@adobe.com

Abstract

In this paper, we introduce the problem of simultaneously detecting multiple photographic defects. **We aim at detecting the existence, severity, and potential locations of common photographic defects related to color, noise, blur and composition.** The automatic detection of such defects could be used to provide users with suggestions for how to improve photos without the need to laboriously try various correction methods. Defect detection could also help users select photos of higher quality while filtering out those with severe defects in photo curation and summarization.

To investigate this problem, we collected a large-scale dataset of user annotations on seven common photographic defects, which allows us to evaluate algorithms by measuring their consistency with human judgments. Our new dataset enables us to formulate the problem as a multi-task learning problem and train a multi-column deep convolutional neural network (CNN) to simultaneously predict the severity of all the defects. Unlike some existing single-defect estimation methods that rely on low-level statistics and may fail in many cases on natural photographs, our model is able to understand image contents and quality at a higher level. As a result, in our experiments, we show that our model has predictions with much higher consistency with human judgments than low-level methods as well as several baseline CNN models. Our model also performs better than an average human from our user study.

1. Introduction

Many natural photos suffer from certain types of photographic defects, e.g., bad exposure, severe noise, and camera shake, due to imperfect capture conditions or limited expertise of the photographer. To improve those images, various manual tools in image editing software (e.g., Adobe Photoshop[®]) and automatic adjustment methods in the research community have been developed to fix specific types of defects [30, 8, 13, 36, 11]. Because many factors affect image quality and there are abundant corrections available for each factor, it becomes difficult for a user without much photographic expertise to understand the defects in

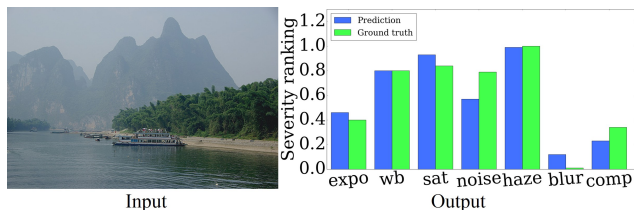


Figure 1. An illustration of detecting multiple photographic defects. For each defect (from left to right: *bad exposure*, *bad white balance*, *over/under saturation*, *noise*, *haze*, *undesired blur*, *bad composition*), we report the relative ranking of a severity score in percentage, compared to all the other photos in a testing set. Higher numbers indicate more severe defects. Our prediction rankings (blue) are consistent with the human judgment (green).

an image and choose proper correction methods. Moreover, with the explosion in the number of photos in one’s personal collection, it is also very tedious, if not impractical, for a user to go through all the photos and choose different corrections. It is therefore desirable to have a tool that can quickly identify the common defects in an image, suggest corresponding tools or auto-correction methods, and guide users to improve their photos. Furthermore, such a technology can also be applied in photo curation and collage to suggest good photos while filtering out bad ones.

To this end, we introduce the problem of simultaneously detecting multiple photographic defects. That is, we detect the existence and severity of a number of common photographic defects. By consulting professional photographers and analyzing a large amount of image editing data, we identified the seven most common defects, namely, *bad exposure*, *bad white balance*, *over/under saturation*, *noise*, *haze*, *undesired blur*, and *bad composition*. Given a natural photo, we would like to predict the severity of these seven defects at the same time, as illustrated in Figure 1. We note that although there is research on estimating the degree of specific defects (e.g., noise level [23] or blur amount [3]), to our knowledge, there is no prior work addressing the problem of simultaneous detection of multiple defects.

To facilitate the research on this problem, we collected a dataset containing 12,853 natural images, each with detailed user annotations on the severity of all the seven de-

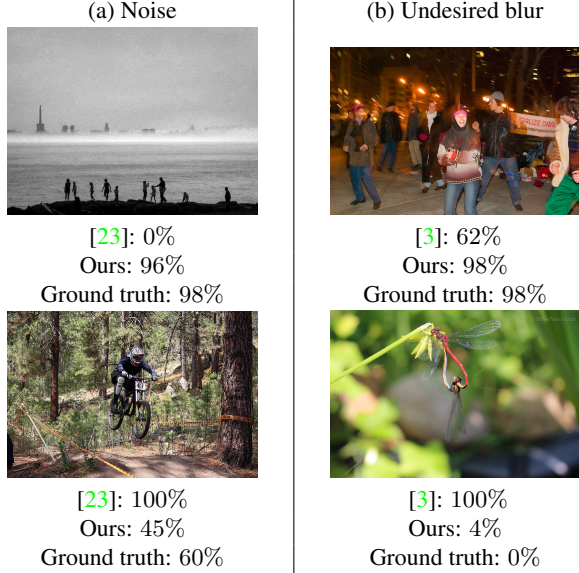


Figure 2. Failure cases of the two previous defect estimation methods on *noise* [23] and *undesired blur* [3], respectively. Previous methods fail to detect the defects in the first row, and are confused by highly textured areas or desired depth-of-field in the second row. Our predictions are more consistent with ground truth. The percentage numbers measure the ranking of the image in terms of defect severity compared to other photos in a testing set.

fects. This dataset allows us to train and evaluate algorithms based on human judgments, which is a distinct difference from previous methods that use synthesized artifacts as ground-truth defects [23, 18]. Synthetic defects are usually generated under certain assumptions regarding the defect patterns (e.g., Gaussian noise, or uniformly darkening the images). As a result, the methods developed on top of such data cannot cover the much more diverse defect patterns present in natural photos. For example in Figure 2 (a), the noise level estimation method [23] is easily fooled by real noise or highly textured areas. Moreover, human judgments of image defects also heavily rely on understanding the important content that was intended to be captured. Without such higher-level understanding, the blur analysis method [3] in Figure 2 (b) cannot differentiate between undesired motion blur and a desired depth-of-field effect.

By contrast, leveraging the newly collected dataset, we formulate the problem as a multi-task prediction and learn a multi-column deep convolutional neural network (CNN) to simultaneously predict the severity of all the defects. By taking both the entire image and local patches as input, the learned model can better understand the image content while still being able to focus on local statistics, and have more accurate predictions, as shown in Figure 2.

The contributions of this paper are therefore: (1) We introduce a new problem of detecting multiple photographic defects, which is important for applications in image editing and photo organization. (2) We collect a new large-scale

dataset with detailed human judgments on seven common defects, which will be released to facilitate the research on this problem. (3) We make a first attempt to approach this problem by training multi-column neural networks that consider both the global image and local statistics. We show in our experiments that our model achieves higher consistency with human judgments than previous single-defect estimation methods as well as baseline CNN models, and performs better than an average user.

Code and dataset are publicly available at <https://github.com/ningyul991/DefectDetection>.

2. Related Work

We discuss previous work, as grouped into three areas.

Single defect estimation and correction. There have been many efforts focused on fixing a specific type of photographic defect, e.g., exposure correction [33, 41], haze removal [13, 42, 32], denoising [31, 8, 1], deblurring [12, 35] and image cropping [22, 40, 11]. However, most of these methods directly generate an improved image without explicitly estimating the existence or severity of the defect. The level of white Gaussian noise in an image is explicitly estimated in [23], while Chakrabarti *et al.* [3] analyze the amount of spatially-varying blur. Both of these methods rely on low-level statistics under the assumption that such defects already exist in the image, and may not work very well given an arbitrary natural photo. More importantly, none of those previous works tackles the detection of all the common defects at the same time, as in our study.

Deep convolutional neural networks (CNN). Deep convolutional neural networks [37, 14] have shown tremendous success in capturing high-level image content, and have achieved state-of-the-art in various computer vision tasks [16, 4, 10]. Previous papers have demonstrated that multi-column networks can have improved performance over single-column networks [7, 25, 26], by leveraging the information from multiple related tasks, or taking inputs with different scales [29, 5, 26]. Inspired by these results, we formulate the multiple defects estimation problem as a multi-task prediction, and design an end-to-end multi-column network that shares weights in earlier stages and splits out columns in the later stages for each defect.

Image quality assessment. The conventional no-reference image quality assessment (NR-IQA) evaluates visual distortions including JPEG compression, additive white Gaussian noise, Gaussian blur, etc. [21, 6, 18]. In these tasks, distortions are synthetically added and uniformly distributed over the entire image. On the contrary, our problem focuses on common defects found in photos in the wild, which exist mainly due to limitations at capture time. Our problem does not have any assumptions regarding the existence, types, and locations of the defects,

and involves high-level image content understanding driven by human judgment, and is therefore a significantly different problem. The recent work of deep photo aesthetics assessment [27] directly classifies query images into high or low aesthetics, which is also different from our problem.

3. Photographic Defect Severity Dataset

Because our problem is new and involves human judgments, we need to run a user study to collect human judgments, and also define a suitable evaluation metric. We first discuss in Section 3.1 our new dataset with detailed human annotations on natural photos. Next, in Section 3.2, we introduce an evaluation metric that is well-suited for our problem. Then, in Section 3.3, we provide user consistency analysis based on the proposed evaluation protocol. We will release our dataset and evaluation protocol to promote research on this problem.

3.1. Data Collection

To determine the most common and important defects, we consulted professional photographers and analyzed a large amount of image editing data. In the end, we selected seven types of photographic defects: *bad exposure*, *bad white balance*, *over/under saturation*, *noise*, *haze*, *undesired blur*, and *bad composition*.

We then randomly sampled 12,853 natural photos from the Yahoo Flickr Creative Commons 100M dataset [38], and obtained the annotations of severity scores for each defect through Amazon Mechanical Turk (AMT)¹. Specifically, for the *over/under saturation* defect, we provided five levels of severity for users to choose from: {*severely under-saturated*, *mildly under-saturated*, *normal saturation*, *mildly over-saturated*, *severely over-saturated*}, which map to a score set of $\{-1.0, -0.5, 0.0, 0.5, 1.0\}$. For all the other defects, we provided three levels of severity: {*none*, *mild*, *severe*}, which map to a score set of $\{0.0, 0.5, 1.0\}$.

When collecting the annotations, we randomly inserted a small set of “sanity check” images with known defect severity levels. Most of these images have obviously severe defects or are defect-free, so a careful user will do a very good job on these images. We can thus filter out bad user annotations by measuring users’ performance on those images. More details about the data collection process, such as the user interface, the qualification test, and the quality control procedure, are included in the supplementary material.

In the end, each image has five valid user annotations for each defect. We calculate the final ground-truth severity scores as a weighted average over the five user annotations, in which the weights are proportional to users’ accuracy on the “sanity check” images and normalized among the five users. We found that such a weighted averaging

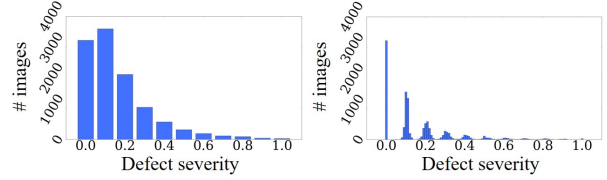


Figure 3. The histogram of the ground-truth severity scores for the *bad exposure* defect, shown with a coarse bin size (left) and with a fine bin size (right). Each bin in the left figure corresponds to a peak in the right figure. The histogram has 11 peaks due to the discrete levels used in user annotation.

process can significantly reduce annotation noise, and generate quite consistent ground-truth scores. More analysis regarding users’ consistency is described in Section 3.3.

Figure 3 (left) shows the distribution with a coarse bin size of the ground-truth severity scores for the *bad exposure* defect. The distribution has a long tail, with most images containing no or mild exposure problems. This is expected since the images are randomly sampled from a large collection of photos and follow a natural distribution in terms of image quality. Another interesting observation is that the scores, even if plotted with a fine bin size (as shown at right in Figure 3), form 11 peaks. That is because the annotations given by each user have three levels: 0.0 (*none*), 0.5 (*mild*), and 1.0 (*severe*). An equal weighted average over five user annotations would result in 11 discrete levels from 0.0 to 1.0 with a step of 0.1. When the averaging is weighted by users’ accuracy, the scores become a little more dispersed but still form 11 peaks around those discrete levels. Each peak in the right histogram corresponds to a coarse bin in the left histogram. We observe a similar distribution for the scores of other defects except for saturation, whose score distribution has 21 peaks, because its annotations by each user have five different levels instead of three.

Finally for experimental evaluation, we randomly split the dataset into a training set with 11,313 images and a testing set with 1,540 images.

3.2. Evaluation Metric

In order to fairly evaluate the performance of users and algorithms, it is important to have an evaluation metric that is suitable for our problem and dataset. We first discuss three key properties that a metric should have, then discuss limitations of some simple evaluation metrics, and next discuss our preferred metric.

The three key properties that we desire from an evaluation metric are: (1) *Balance*: the metric should give a roughly equal contribution to the final cost for images that fall under each severity of defect. This is because there are much more defect-free images than defective ones in our dataset, as shown in Figure 3, so the evaluation metric should perform a rebalancing to account for this; (2) *Proportionality*: the metric should consider slight errors in

¹www.mturk.com

Table 1. The mean cross-class ρ and Kendall’s W among user annotations for each defect.

	Bad exposure	Bad white balance	Over/under saturation	Noise	Haze	Undesired blur	Bad composition	Mean
Cross-class ρ	0.7691	0.7498	0.7944	0.8236	0.8530	0.8528	0.6168	0.7799
Kendall’s W	0.5247	0.4863	0.5435	0.5470	0.6208	0.6388	0.4203	0.5402

prediction as better than extreme errors in prediction. For example, if we have a defect-free image (class 1) and we predict that it is slightly defective (class 2), this should be better than predicting that the same image is highly defective (class 11); and (3) *Ranking*: a ranking-based metric that considers only the order of the predictions from the model is preferable to an absolute metric. This is especially true for applications where the relative ranking is important, such as photo ranking or curation [20], and for comparisons with previous work, where the scores output by an algorithm may not be directly comparable to the user severity scale.

We now discuss how a few simple metrics fall short of the key properties. The L^2 loss does not satisfy key properties of balance (1) and ranking (3). The overall classification accuracy could be computed by quantizing the defect scores into 11 or 21 classes as discussed in Section 3.1. However, accuracy does not satisfy any of the key properties. The classification accuracy given varying class bias tolerances is generalized to satisfy proportionality (2) but still does not satisfy the other properties. A Spearman Rank Correlation Coefficient [28] could be computed by forming two ranked image lists based on the prediction and ground-truth scores, respectively. However, the Spearman Rank Correlation does not satisfy the key property of balance (1) and proportionality (2). In particular, it fails at proportionality (2), because two sets of images that all fall into a given class such as slightly defective (class 2) can still have quite different rankings.

In order to satisfy all three key properties, in this work we propose a new evaluation metric, the **cross-class ranking correlation** (cross-class ρ). Specifically, we assign the test images to one of the 11 classes according to their ground-truth defect severity scores (21 classes for saturation). The classes (the bins in Figure 3 left) naturally fit the peaked distribution of our dataset, which is shown in Figure 3 right. During evaluation, we randomly sample one image from each class from the ground truth. Those images compose an ordered list based on the severity levels of classes they are sampled from. When a prediction is made for the defect scores of those images, we can also sort the images according to their predicted scores and form another ordered list. We then calculate the regular Spearman Rank Correlation Coefficient ρ [28] between the two lists, yielding a score within $[-1, 1]$. A larger Spearman coefficient indicates the orders in the two lists are more similar, and the predictions are more consistent with the ground truth. To obtain a robust evaluation, we repeat the random sam-

pling and correlation calculation many times (15, 000) and use the mean as our final cross-class ρ .

With the cross-class ranking correlation, we achieve all three key properties of an evaluation metric. During image sampling, each class only contributes one image to the list, so this acts to rebalance the dataset, and satisfy the balance property (1). The property of proportionality (2) is satisfied because no penalty is applied if two images fall within the same class, and if images are within a different class, the correlation decreases as the classes become further apart. The ranking property (3) is trivially satisfied.

3.3. User Consistency Analysis

After specifying our evaluation metric, we are able to examine the consistency of AMT users’ annotations. We conduct consistency analysis on each group of five users who annotated the same batch of images. We compare the annotations from two users against the other three annotations. Specifically, we calculate the mean annotations among the two subgroups separately and utilize the cross-class ρ to evaluate the consistency. For each batch, we average the correlations over all possible two-against-three splits. We additionally estimate the p-value of a t-test for each correlation, which measures the statistical significance of the correlation relative to a null hypothesis of uncorrelated response. We use the Benjamini-Hochberg procedure [2] to control the false discovery rate (FDR) for multiple correlation hypotheses. At an FDR level of 0.05, we calculate the percentage of batches with significant agreement among users. The average cross-class ρ for each defect are listed in Table 1. They are all above 0.6, and mostly around 0.8, where the valid range for ρ is $[-1, 1]$. The percentage of significant batches is at least 99.55% for all the defects.

We further evaluate the annotation consistency with Kendall’s Coefficient of Concordance (W) [19], which directly calculates the agreement among multiple users, and accounts for tied ranks. Kendall’s W ranges from 0.0 (no agreement) to 1.0 (complete agreement). We estimate the p-value of a Chi-squared test to evaluate the statistical significance. We use the same Benjamini-Hochberg procedure to measure the percentage of batches with the significant agreement. Kendall’s W values for each defect are listed in Table 1. These show a similar trend as cross-class ρ , and also have a percentage of batches with significant agreement of at least 99.55%. Both measures demonstrate the consistency across AMT users and indicate that the annotations are reliable for scientific research.

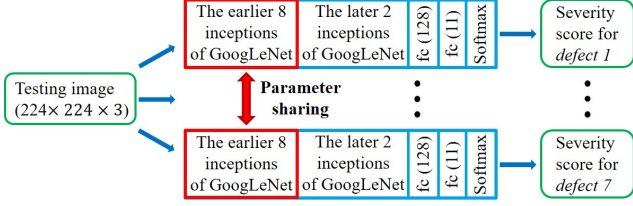


Figure 4. This diagram shows the multi-column GoogLeNet [16] architecture for multi-task predictions. Here “ $fc(x)$ ” represents a fully connected layer with x hidden neurons. Red frames indicate the layers with shared parameters.

4. Simultaneous Detection of Multiple Defects

The availability of this new dataset enables us to train a deep convolutional neural network (CNN) to directly learn high-level understanding of photographic defects from human judgments. In this section, we describe the details of CNN training, including the architecture, pre-processing of input images, loss functions, and the data augmentation process to rebalance our skewed training data.

4.1. Multi-Column Network Architecture

Our goal is to predict the severity of seven defects at the same time. These defects are related to low-level photo properties such as color, exposure, noise, and blur, and high-level properties such as faces, humans, and compositional balance. We note that both low- and high-level content features may be useful. Therefore, we use a multi-column CNN, in which the earlier layers of the network are shared across all the defects to learn defect-agnostic features, and in later layers, a separate branch is dedicated to each defect to capture defect-specific information. Figure 4 shows our architecture. We build upon GoogLeNet [37, 16], which contains convolutional modules called inceptions. We select GoogLeNet rather than other prevalent architectures, e.g., VGG nets [34] or ResNet [15], because its lighter memory requirement enables multi-column training with a larger batch size. We use the first 8 inceptions of GoogLeNet [16] as shared layers, and then dedicate a separate branch for each defect with two inceptions and fully-connected layers.

We also tried two other baseline models: (1) A single-column network that directly predicts seven defect scores, and (2) Seven separate networks, each predicting one defect. The comparisons in Section 5.1 show that our branching architecture is better than these alternatives.

4.2. Network Input

In order to detect the defects, we need both a global view of the entire image and a focus on the statistics in local image areas. Therefore, we prepare two different types of inputs for the network: (1) downsized holistic images, which contains complete image content, and (2) patches randomly sampled from the images at the original resolution, which

retains high-frequency statistics especially useful for the detection of some defects such as noise and blur. We make the simplifying assumption that we can assign each local patch the same holistic severity score. This sometimes introduces noise when a defect appears only locally. But by aggregating over many patches the network can learn to ignore this noise (this can be observed in Table 2 in the outperformance of the patch model for certain defects). We also tried a weakly supervised architecture by estimating an attention map (similar to [39]) which uses different weights for each patch, but the results show it did not help in our case.

Mixing the two input types would confuse the network during training. Accordingly, we train a separate network in Figure 4 for each type of input, with a goal that each network can capture complementary information.

The network with patch inputs does not predict the score regarding bad composition, because image composition should be solely considered over the entire image. For all the other six defects, we combine the predicted scores from the patch and holistic networks. We found that a simple average of scores using equal weights achieves good results, and outperforms both of the two individual models, as shown in Section 5.1. This demonstrates that the global and local information captured in the two networks is complementary for this problem. We also tried to optimize the weights using quadratic programming on a separate validation set, but did not observe much improvement from this.

4.3. Loss Functions

Due to the distribution of our ground truth annotations as shown in Figure 3, where the scores are mostly distributed around discrete peaks, we found that it works better to formulate our loss to involve classification rather than regression. However, the standard cross-entropy loss used in classification ignores the relations between the classes, as discussed in Section 3.2. In other words, all misclassifications are treated equally. In our case, we should impose more penalty if we misclassify an example in class 1 (no defect) to class 11 (severe defect), compared with a misclassification from class 1 (no defect) to class 2 (very mild defect). This is the property of proportionality from Section 3.2.

To accommodate such requirement, we use the infogain multinomial logistic loss² to measure the classification errors. The infogain loss E is mathematically formulated as

$$E = -\frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K H_{l_n, k} \log(p_{n, k}), \quad (1)$$

where N is the number of image samples; K is the number of classes; l_n is the class ground truth of the n^{th} sample; $p_{n, k}$ is the probability of the n^{th} sample classified to the

²http://caffe.berkeleyvision.org/doxygen/classcaffe_1_1InfogainLossLayer.html

k^{th} class, which is the output after the softmax layer satisfying $\sum_k^K p_{n,k} = 1$ and $p_{n,k} \geq 0$. Finally, $H_{l_n,k}$ is the infogain weight for the n^{th} sample with ground truth l_n to be classified to class k . The higher the weight, the greater the reward for that classification result. Therefore, we can assign higher weights between similar classes. We derive our defect-specific infogain matrices from a naive conditional independence assumption and statistics of individual AMT users’ case-by-case annotations. The details are included in the supplementary materials. During testing, once we obtain the classification probabilities over all the classes for an image, since each class is associated with a severity score, we can use the probabilities as weights to obtain an averaged severity score. That score is treated as our final prediction regarding the defect severity of the image.

Note that we use infogain loss only for training and prefer the cross-class ρ for evaluation. This is because cross-class ρ is a metric that uses ranking, and for applications such as photo curation, we care more about relative rankings than absolute scores.

We experimentally show in Section 5.1 that training using the infogain class achieves better performance than using the standard cross-entropy loss. We also tried formulating the prediction as a regression task, and use L^2 loss compared with ground truth scores. The results are reasonable, but not as good as using the infogain loss.

4.4. Data Augmentation

As discussed in Section 3.1, our training data is heavily unbalanced with a high percent of defect-free images. In order to prevent the training from being dominated by defect-free images, we augment more training data on images with severe defects. This also better satisfies the property of balance from Section 3.2. We augment samples in inverse proportion to class member counts but clamp the minimum and maximum sample counts to 5 and 50, respectively. The augmentation operation for the holistic input is random cropping (at half the receptive field) and warping, and for the patch input, random cropping. More details and the histograms before and after augmentation are shown in the supplementary material. We experimentally validate in Section 5.1 that our data rebalancing is crucial to the results.

4.5. Implementation Details

The network is initialized from the GoogLeNet model [37] trained for ImageNet classification [9]. We made some slight modifications on the architecture to make the model more compact and efficient: (1) we remove the two auxiliary classifier branches *loss1* and *loss2*, (2) we trim off the 3×3 convolution branch in *inception_5b*; and (3) in *inception_5b*, we reduce the number of output features of the 1×1 , 3×3 double, and the pooling projection layers to 88, 56, and 32, respectively. The output feature dimension

of *inception_5b* is thus reduced from 1,024 in the original network to 176.

During training, the batch size is 32. The initial learning rate is 0.0001 for the parameter-shared layers and is 10 times larger for the defect-specific layers. All learning rates are multiplied by 0.96 after every 6,400 iterations. We set weight decay as 0.0002 and momentum as 0.9. We implement the training and testing in Caffe [17].

During testing, to obtain the patch model predictions, we crop K random patches from each image and average the scores from the patch networks. We set $K = 10$, which gives a good trade-off between testing time and robustness.

5. Experiments

To predict all the seven defects, the testing time of the proposed model on our Intel i7-6950X CPU (3.00GHz) is 3.6 sec. The average testing time on our NVIDIA Titan X GPU is about 0.5 sec. The holistic model requires 108 MB of memory and the patch model requires 97 MB. Two additional qualitative results are presented in Figure 5.

5.1. Ablation Study

Network Input. The first three rows in Table 2 show the cross-class ρ of the multi-column network with holistic image input, the network with patch input, and the combination of the two, respectively. The mean cross-class ρ in the last column is obtained by averaging the values over all the seven defects. We can see that after combination, the results improve on almost all the defects. This shows the complementarity between the holistic and patch model.

In all subsequent ablation studies, we use the same combined inputs for all the CNN models. That is, we separately train two networks for holistic images and local patches, and average the predictions with equal weight.

Network architectures. To investigate the necessity of having separate branches for each defect, we train a single-column network, in which the parameters are all shared for the defects except the last output. To have a fair comparison in terms of model capacity, we increase the numbers of feature channels in the last two inceptions in the single-column network, to make the number of parameters for this model similar to our model. The results of the single-column network are reported in the 4th row in Table 2. We can see that the results on most defects become worse, as does the mean cross-class ρ . The performance on the composition defect has the biggest decrease, probably because understanding image composition requires higher-level features than color or texture, which are more important for other defects.

On the other hand, one can train a separate network for each single defect, without sharing any parameters. To investigate this, we train a GoogLeNet for each defect separately and report the results in the 5th row in Table 2. We note that by unsharing the parameters, the number of overall

Table 2. Comparison with baseline CNNs in terms of the cross-class ρ on our testing dataset. **Bold** indicates the best performance. Underline indicates the second best.

	Bad exposure	Bad white balance	Over/under saturation	Noise	Haze	Undesired blur	Bad composition	Mean
Multi-column (holistic)	0.7529	0.7614	0.8996	0.6736	0.8346	0.6032	0.7123	0.7482
Multi-column (patch)	0.7825	0.8000	0.8923	0.8197	0.7759	0.6696	-	-
Multi-column (combined)	0.8008	<u>0.8249</u>	0.9098	<u>0.8174</u>	0.8490	0.6867	0.7123	0.8001
Single-column architecture	<u>0.8063</u>	0.8201	0.8817	0.7246	0.7778	0.7447	0.5969	0.7646
Separate networks	0.7972	0.7925	<u>0.9039</u>	0.7403	0.8315	<u>0.7209</u>	0.6656	0.7788
Regression L^2 loss	0.8145	0.8323	0.8995	0.8118	<u>0.8394</u>	0.7008	0.6169	<u>0.7879</u>
Classification loss	0.7850	0.7938	0.8969	0.7426	<u>0.7867</u>	0.7205	<u>0.6929</u>	<u>0.7740</u>
Without augmentation	0.7864	0.7675	0.8907	0.7096	0.8076	0.6349	0.5383	0.7336

Table 3. Comparison with previous methods in terms of the cross-class ρ on our testing dataset. **Bold** indicates the best performance.

	Noise	Haze	Undesired blur
Previous	0.4199 [23]	0.6615 [13]	0.4864 [3]
Ours	0.8174	0.8490	0.6867

trainable parameters in this model is much higher than the one in our multi-column network, resulting in much larger model size and longer testing time. However, our model has better performance on all defects except blur. We investigated the gaps between training and testing performance, and found that the separate networks for single defects are more prone to over-fitting, whereas the shared layers in our network act as a regularizer to improve generalizability.

After comparing with these baseline CNN models, we find that our multi-column architecture gives a good trade-off between performance, compactness, and efficiency.

Loss functions. The results of the networks trained using L^2 loss as regression, and using cross-entropy loss as classification, are shown in the 6th and 7th rows in Table 2, respectively. This shows that using these two losses can also result in reasonably good performance. However, the info-gain loss outperforms these two losses on the predictions for several defects as well as the overall mean cross-class ρ .

We realize we can reach the second best performance when we train using L^2 loss. Therefore, in order to show the significance of the outperformance of our best model, we calculate the p value of the two-tailed Students t-test between the two networks. Here p is 0 to within the double precision accuracy, which indicates that training with the info-gain loss significantly outperforms training with L^2 loss.

Data Augmentation Finally, we show that it is important to perform rebalancing for the training set to achieve good performance. The results without data augmentation (last row in Table 2) are significantly worse, due to imbalance.

5.2. Comparison with Previous Methods

We are not aware of any previous work for simultaneously detecting multiple defects. However, there is previous work for estimating the degree of a single defect, e.g., noise level [23] or blur amount [3]. The method in [23] can

directly predict an overall noise level. The blur estimation method [3] generates a pixel-wise spatially-variant prediction map. We made our best efforts to obtain an overall blur severity assessment from the prediction map, by experimenting with taking different percentiles or the mean. We found that the mean gives the best performance.

In addition, for some adjustment methods, e.g., haze removal [13], we can calculate the adjustment amount for each pixel in the image, where a higher adjustment indicates a more severe haze defect in the original image. We can then obtain an overall haze amount estimation by taking the mean adjustment amount over the entire image. Similar to before, we also experimented with various percentiles, but found the mean performed best.

A comparison of our model with these three methods is presented in Table 3. The cross-rank ρ metric is especially useful here, because the score ranges of these methods are different and not calibrated with our ground-truth scores, but the relative rankings of the test images are comparable among different methods. We can see that the improvement of our model over these methods is substantial.

5.3. Comparison with Human Performance

We further compare our predictions to individual users’ annotations on the test set. To fairly evaluate a user’s performance, we use the mean of the other four users’ annotations as ground truth instead of the mean over all the five, to remove the influence of the given user on the ground truth. The same ground truth is then also used to measure the performance of our model, so that the comparison between our model and that user is fair. We show in Table 4 the comparison between our results and the averaged user performance. We can see that our model performs better than an average user on most of the defects.

5.4. Evaluation on Synthetic Data

Although our model was trained on our dataset of defective images in the wild, we can also validate our trained model on an easier dataset of synthetically generated global defects. We separately generate defective images for under exposure, over exposure, over/under saturation, Gaussian

Table 4. The performance of our model compared to individual users. The 1st and 2nd rows indicate, for different defects, the average performance of users and our combined multi-column CNN, respectively. **Bold** indicates best performance over the first two rows. The 3rd row gives a percentage indicating what fraction of users our model’s predictions outperform.

	Bad exposure	Bad white balance	Over/under saturation	Noise	Haze	Undesired blur	Bad composition	Mean
User cross-class ρ	0.6307	0.4953	0.6906	0.5652	0.5755	0.6378	0.5348	0.5900
Our cross-class ρ	0.7572	0.7217	0.8688	0.6750	0.7391	0.6320	0.5990	0.7133
Our ranking	75%	89%	100%	72%	78%	44%	87%	78%

Table 5. Our model’s performance on five synthesized defects.

Under exposure	Over exposure	Over/under saturation	Gaussian noise	Motion blur
0.9560	0.8440	0.9968	0.9573	0.8986

noise, and spatially invariant motion blur. We first select for each defect all of the defect-free testing images (there are between 420 and 940 such images). For each such image, we synthesize a sequence of defective images with either 11 or 21 different levels of a global parameter, where the number of levels is chosen to be consistent with the class structure in our user dataset discussed in Section 3.1. We then measure the ranking correlation between the predicted scores and the parameter choices. This can be viewed as a simplification of our cross-class ρ , which preserves the three key properties for this task, but does not require random sampling. The mean result over each dataset is listed in Table 5. Note that our model performs better in the synthetic datasets than in the real dataset, which implies that the synthetic task is easier because the defects are global and require less high level information to detect. The result also demonstrates the generalizability of our model. Please see the supplemental material for more details.

5.5. Photographic Defect Localization

We also experimented with our well-trained patch model to localize photographic defects. No re-training is required. To do this, we converted the architecture to fully-convolutional [24], by removing the last 7×7 pooling layer and replacing the fully connected layers with convolutional layers with 1×1 spatial kernels. We then added an upsampling layer (bilinear interpolation) afterward. The resulting network accepts an image with arbitrary size and outputs a spatially-variant defect map with the same size. Figure 6 shows two examples of defect maps. We see our model can roughly localize the defective image areas. Although we have only obtained preliminary results for this, such spatially varying maps could open an avenue for future work such as applications in spatially-variant image corrections or guidance. It could also be a promising future work to collect spatial annotations for defect severity from a user study, and then train a defect localization model specifically.

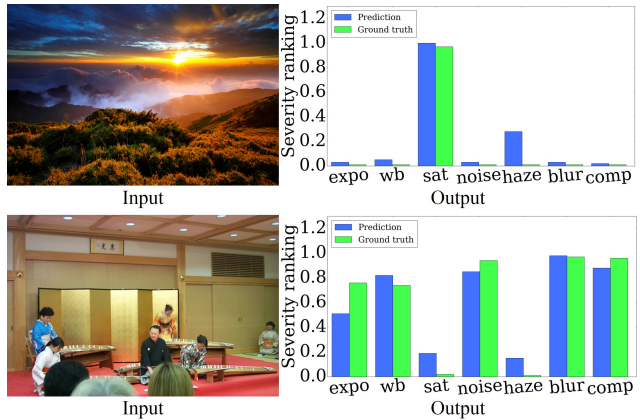


Figure 5. Two visual results of our defect detection. For each defect (from left to right: *bad exposure*, *bad white balance*, *over/under saturation*, *noise*, *haze*, *undesired blur*, *bad composition*), we report the relative ranking of a severity score in percentage, which measures the defect severity of a given image compared to all the other photos in a testing set. Higher numbers indicate more severe defects. Our prediction rankings (blue) are consistent with the human judgment (green).

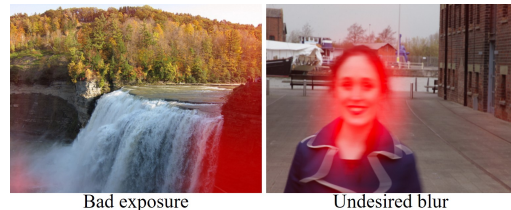


Figure 6. Examples of defect localization, where the amount of red color indicates the severity of defects in a local region. In the left image, our heat map highlights indicates that the rock in shadow suffers from the *bad exposure* defect. In the right image, our heat map indicates that the girl’s head suffers from motion blur.

6. Conclusion

In this paper, we introduce the problem of simultaneously detecting multiple photographic defects, and make a first attempt of addressing this problem by collecting a large-scale dataset with human annotation, and training a multi-column CNN for prediction. In the experiments, we validated that the proposed model achieves much higher consistency with human judgments than previous single-defect estimation methods as well as baseline CNN models, and also outperforms an average user.

7. Acknowledgement

We thank our anonymous reviewers for beneficial feedback. Thanks to the photographers for licensing photos under Creative Commons or public domain. This project was funded by Adobe Research Funding.

References

- [1] A. Adams, N. Gelfand, J. Dolson, and M. Levoy. Gaussian kd-trees for fast high-dimensional filtering. In *ACM Trans. Graphics*, volume 28, page 21. ACM, 2009. 2
- [2] Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188, 2001. 4
- [3] A. Chakrabarti, T. Zickler, and W. T. Freeman. Analyzing spatially-varying blur. In *CVPR*, pages 2512–2519. IEEE, 2010. 1, 2, 7
- [4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014. 2
- [5] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. *arXiv preprint arXiv:1511.03339*, 2015. 2
- [6] A. Chetouani, A. Beghdadi, S. Chen, and G. Mostafaoui. A novel free reference image quality metric using neural network approach. In *Proc. Int. Workshop Video Process. Qual. Metrics Cons. Electr.*, pages 1–4, 2010. 2
- [7] D. Cireřan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, pages 3642–3649. IEEE, 2012. 2
- [8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. 1, 2
- [9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 6
- [10] M. Dixit, S. Chen, D. Gao, N. Rasiwasia, and N. Vasconcelos. Scene classification with semantic fisher vectors. In *CVPR*, pages 2974–2983, 2015. 2
- [11] C. Fang, Z. Lin, R. Mech, and X. Shen. Automatic image cropping using visual composition, boundary simplicity and content preservation models. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1105–1108. ACM, 2014. 1, 2
- [12] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. In *ACM Trans. Graphics*, volume 25, pages 787–794. ACM, 2006. 2
- [13] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE Trans. PAMI*, 33(12):2341–2353, 2011. 1, 2, 7
- [14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 2
- [15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5
- [16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 2, 5
- [17] 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. 6
- [18] L. Kang, P. Ye, Y. Li, and D. Doermann. Convolutional neural networks for no-reference image quality assessment. In *CVPR*, pages 1733–1740, 2014. 2
- [19] M. G. Kendall and B. B. Smith. The problem of m rankings. *The annals of mathematical statistics*, 10(3):275–287, 1939. 4
- [20] 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. 4
- [21] C. Li, A. C. Bovik, and X. Wu. Blind image quality assessment using a general regression neural network. *IEEE Transactions on Neural Networks*, 22(5):793–799, 2011. 2
- [22] L. Liu, R. Chen, L. Wolf, and D. Cohen-Or. Optimizing photo composition. In *Computer Graphics Forum*, volume 29, pages 469–478. Wiley Online Library, 2010. 2
- [23] X. Liu, M. Tanaka, and M. Okutomi. Single-image noise level estimation for blind denoising. *IEEE transactions on image processing*, 22(12):5226–5237, 2013. 1, 2, 7
- [24] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 8
- [25] X. Lu, Z. Lin, H. Jin, J. Yang, and J. Z. Wang. Rapid: rating pictorial aesthetics using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 457–466. ACM, 2014. 2
- [26] 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 *ICCV*, pages 990–998, 2015. 2
- [27] L. Mai, H. Jin, and F. Liu. Composition-preserving deep photo aesthetics assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 497–506, 2016. 3
- [28] J. L. Myers, A. Well, and R. F. Lorch. *Research design and statistical analysis*. Routledge, 2010. 4
- [29] M. Oquab, L. Bottou, I. Laptev, J. Sivic, et al. Weakly supervised object recognition with convolutional neural networks. In *Proc. of NIPS*. Citeseer, 2014. 2
- [30] I. Ovsiannikov. Backlit subject detection in an image, Oct. 12 2010. US Patent 7,813,545. 1
- [31] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. In *ECCV*, pages 568–580. Springer, 2006. 2
- [32] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. Single image dehazing via multi-scale convolutional neural networks. In *ECCV*, pages 154–169. Springer, 2016. 2

- [33] J. C. Russ and R. P. Woods. The image processing handbook. *Journal of Computer Assisted Tomography*, 19(6):979–981, 1995. [2](#)
- [34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [5](#)
- [35] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *CVPR*, pages 769–777. IEEE, 2015. [2](#)
- [36] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *ICCP*, pages 1–8. IEEE, 2013. [1](#)
- [37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. [2](#), [5](#), [6](#)
- [38] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. The new data and new challenges in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015. [3](#)
- [39] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015. [5](#)
- [40] J. Yan, S. Lin, S. Bing Kang, and X. Tang. Learning the change for automatic image cropping. In *CVPR*, pages 971–978, 2013. [2](#)
- [41] L. Yuan and J. Sun. Automatic exposure correction of consumer photographs. In *ECCV*, pages 771–785. Springer, 2012. [2](#)
- [42] Q. Zhu, J. Mai, and L. Shao. A fast single image haze removal algorithm using color attenuation prior. *IEEE Transactions on Image Processing*, 24(11):3522–3533, 2015. [2](#)