Image Distortion Detection using Convolutional Neural Network

Namhyuk Ahn
Ajou University
Suwon, Korea
aa0dfg@ajou.ac.kr

Byungkon Kang
Ajou University
Suwon, Korea
byungkon@ajou.ac.kr

Kyung-Ah Sohn
Ajou University
Suwon, Korea
kasohn@ajou.ac.kr

Abstract

Image distortion classification and detection is an important task in many applications. For example when compressing images, if we know the exact location of the distortion, then it is possible to re-compress images by adjusting the local compression level dynamically. In this paper, we address the problem of detecting the distortion region and classifying the distortion type of a given image. We show that our model significantly outperforms the state-of-the-art distortion classifier, and report accurate detection results for the first time. We expect that such results prove the usefulness of our approach in many potential applications such as image compression or distortion restoration.

1. Introduction

With the development of the Internet and mobile devices, demand for streaming media and cloud service have skyrocketed. These services need a lot of storage to store multimedia, and it is crucial to compress data using lossy compression techniques before storing. Higher compression level is better in terms of storage, but it could cause serious local distortion to images. However, if we can detect the region in which the distortions occurred, we can correct the problem by dynamic compression techniques. Such techniques reduce the compression level of detected distortion regions and re-compress using the reduced level.

Our motivation for conducting this research is to build a system that detects the distortion region and performs compression dynamically. Hence, automatic distortion detection is an essential part of this system. However, despite the importance of this task, recent image quality assessment (IQA) methods only focus on predicting perceptual quality scores, such as the mean opinion score (MOS) [2, 3, 11, 12].

One might question the validity of the assumption that multiple distortions exist in a single image. While it is true that distortions in an image are likely to occur globally rather than locally, we consider a situation where individual images are assembled to form a larger one. For example, when creating a panorama picture, the compression of each shot may be subjected to independent distortion. When the individual shots are combined to form the full photo, it might end up with localized distortions.

In recent years, many deep learning based IQA approaches have been proposed, especially for non-reference IQA (NR-IQA). NR-IQA methods perform image quality assessment without any direct comparison between the reference and the distorted image. The IQA-CNN [11] model is the first such model that applies deep learning in IQA task. This model uses a convolutional neural network (CNN) composed of five layers, that achieves the result comparable to the full-reference IQA (FR-IQA) methods, such as FSIM [24].

Deeper networks have been used in [3], whose structure is inspired by the VGGNet [22], and yields results that surpass FR-IQA approaches. DeepBIQ [2] is the first to use pre-trained CNN and show state-of-the-art result in the IQA task. However, although there exist many outstanding results, the distortion classification task remains mostly unchallenged. Two notable exceptions to this are the IQA-CNN+ and the IQA-CNN++ [12], that predict both the MOS and the distortion type with the similar network used in IQA-CNN. One shortcoming of these models is that they are shallow architectures, and thus might have limited capacity to successfully solve our task.

In addition, to best to our knowledge, distortion detection task with deep learning method has not been applied yet. The reason is twofold: First, there is no sufficiently large distortion detection dataset suitable for deep learning, and second, detection task is a much more challenging problem than a classification or predicting MOS are. The difficulty of distortion detection is based on the fact that images can have heterogeneous and multiple distortion types. In general, most IQA datasets contain only homogeneous distortion types that make prediction relatively easy.

To tackle these issues, we created a new dataset for both distortion classification and detection. Then we apply pre-trained CNNs such as VGGNet [22] and ResNet [10] for distortion classification. Finally, we use deep learning based

2327-0985/17 $31.00 © 2017 IEEE
DOI 10.1109/ACPR.2017.95

(a) Reference (b) *Classification* (c) *Detection-basic* (d) *Detection-difficult*

Figure 1: Example of *Flickr-Distortion* dataset. (a) is the reference image, (b) is the distorted image with salt and pepper noise for classification task. (c) and (d) are both for detection task, with different levels of detection difficulty described in Section 3.2.

detection methods such as single shot multi-box detector (SSD) [16] to locate the distortion regions.

Our main contributions are as follows: 1) We create a dataset for distortion classification and detection task. There are no publicly available such datasets. 2) We fine-tune a pre-trained CNN to this dataset to get high performance. 3) We propose a distortion detection system that uses an existing CNN model trained to achieve good performance in object detection task. To the best of our knowledge, our method is the first attempt to use deep learning based detection method in distortion detection task.

Section 3 presents our dataset, followed by the description for our system in Section 2. Experiments and results are explained in Section 4, and finally conclusion in Section 5.

2. CNN for Distortion Recognition

The main model we choose for our distortion recognition system is a convolutional neural network (CNN) [14]. CNNs have been widely used and verified over a variety of image understanding tasks [5, 13, 15]. The overall structure we use is a 'Y'-shaped CNN that performs distortion classification and detection simultaneously. Both the classifier and the detector share the same feature-extraction portion, after which the structure splits into two sets of layers to perform classification and detection, respectively.

2.1. Distortion Classification

In this paper, we experiment with VGG-16 [22] and ResNet-101 [10] models. Both networks are variants of CNN which consist of several convolution, pooling and fully-connected (FC) layers to recognize images. VGG-16 has 13 convolution layers and 3 FC layers. Because of the simplicity of this network, many researchers use the VGG-16 as a base network. The Atrous VGGNet is introduced by DeepLab [6] with an architecture similar to the VGGNet, but with a difference in the number of parameters in the final fully-connected layers, and its use of Atrous convolution that allows for the processing of arbitrary-sized field-of-views.

ResNet uses *residual connections* to avoid the degradation problem. Without residual connections, deep networks are known to not only overfit but also show increasing training error. Unlike the VGGNet, ResNet-101 uses 101 layers with only the last layer being fully-connected. Additionally, a global average pooling technique is used to reduce the number of parameters.

In practice, training the entire CNN from scratch is a difficult and time-consuming job. Also, if the dataset does not have sufficient training data, training does not converge well. Therefore, it is common to use pre-trained networks which have been trained on large external datasets such as ImageNet [8]. This transfer learning strategy works well if the distribution of the source dataset (used for pre-trained) and target dataset are similar. As ImageNet and our dataset have similar distribution, we use CNNs pre-trained on ImageNet for all our experiments.

2.2. Distortion Detection

For the distortion detection task, we use the single shot multibox detector (SSD) [16]. With the development of CNN, many detection methods have been proposed such as R-CNN [9], Faster R-CNN [19], YOLO [18], OverFeat [20], and SSD [16]. R-CNN and its variants perform state-of-the-art detection, while inference time is very slow due to the limitation of their architecture. On the other hand, YOLO and SSD are real-time detection algorithms, with SSD outperforming YOLO.

SSD computes multi-scale feature maps for detection by adding extra convolution layers at the end of the base network. Then six output feature maps from different convolution layers are concatenated to form the final layer. With this idea, SSD effectively detects from objects of various

sizes using a single, simple architecture, since the output maps from lower layer tend to capture fine-grained details of object. Predictors of SSD rely on convolution layers instead of the conventional fully-connected layers, to reduce inference time.

In our experiments, we use the best setting for SSD: 1) Use Atrous VGG-16 as a baseline network since it shows similar result with faster running time. 2) We use 300x300 as the input dimension. If we increase input dimension to 500x500, the inference becomes much slower while the performance gain is relatively small. This shows that using 300x300 input can capture small-sized objects reasonably well in short time via multi-scale feature maps. 3) On the contrary to the SSD used in object detection task, the only data augmentation we use is the horizontal flip. This is because affine transformations might corrupt the details of the distortion, such as in the case of scaling and shearing.

3. Flickr-Distortion Dataset

We creat a new dataset named *Flickr-Distortion* dataset to evaluate image distortion classification and detection task. To make this dataset, we first collect 804 reference images from Flickr with similar way to NUS-WIDE [7] dataset, and make distorted images using the collected reference images. We use eight distortion types: 1) Gaussian white noise (GWN), 2) Gaussian blur (GB), 3) salt and pepper noise (S&P), 4) quantization noise, 5) JPEG compression noise, 6) low-pass noise, 7) denoising and 8) fnoise.

The reason we do not use the LIVE dataset directly is that it contains *global* distortions, whereas we deal with *local* distortions. Furthermore, prevalent distortion dataset such as LIVE [21] or TID2013 [17] have insufficient amount of reference images which are not suitable for training deep learning models.

3.1. Flickr-Distortion-Classification Dataset

In the dataset for distortion classification task, each reference image is distorted using eight distortion types with three levels. Thus, a single reference image results in 24 distorted images. The distortion procedure follows the LIVE dataset [21], and includes such distortion types as homogeneous distortion. The distortion is applied to the entire image (see Fig. 1b). We create 19,296 distorted image in total, and randomly split data into 60% training, 20% validation and 20% test set.

We implemented the generation of most noises except fnoise, for which we used the scikit-image [23] python library. Detailed values we use are as follows: Gaussian noise is generated with three values of variances: $\{0.0125, 0.025, 0.05\}$, and the amount of salt and pepper noise are same as Gaussian noise. For the Gaussian blur, we use the three sigma levels $\{1.5, 3, 6\}$ and in JPEG compression we use $\{20, 10, 5\}$ quality levels. To implement the low-pass noise, we simply scaled images with ratios $\{0.3, 0.1, 0.03\}$, and re-sized to the original image size. We implemented denoising with non-local means algorithm [4] using factors $\{0.04, 0.06, 0.08\}$, but with fixed batch size and patch distance of 7 and 11, respectively. Finally, fnoise is implemented using noise level $1/f$ with factor $f \in \{2.5, 5, 10\}$.

3.2. Flickr-Distortion-Detection Dataset

Unlike the *Classification* dataset, each image in the detection dataset can have heterogeneous and multiple distortions, as shown in Fig. 1c. Since SSD network used in the detection task accepts images of dimension 300x300 as input, we crop the center of correct size before applying distortion.

Distortion levels are chosen uniformly at random with range of minimum and maximum values used in the *Classification* dataset. When choosing the distortion regions, we sample the number of regions in a single image from a uniform distribution over the interval $[1, 4]$. The ratio of region size to image size is also picked uniformly at random from $[0.3, 0.7]$ (average is 0.43). We generate 20 images per reference image, with a total of 16,080 images created. For evaluation, we split data into 80% training and 20% for test.

The assumption on the number of regions and sizes in the *Detection* dataset is quite reasonable. However in practice, there may be many small regions of distortions. Therefore, as in Fig. 1d, we created another dataset named *Detection-difficult* sets (Above dataset is named with *Detection-basic*). In this dataset, the minimum and maximum number of regions per image are 5 and 9, respectively. The ratio of region and image size is changed to 0.1 and 0.3, with an average of 0.18.

4. Experiment

In this section, we describe our experimental results. For ease of exposition, we separate the report on classification and detection into subsections. With the exception of pre-training, our dataset is given in Section 3.

4.1. Distortion Classification

We first evaluate the distortion classification task using pre-trained networks. To do this, we remove the last fully-connected layer in the pre-trained network and freeze all layers in network. Then, we add a new fully-connected layer suited for the number of classes of our dataset. Only this new layer is trained from scratch. Since our classification dataset homogeneously distorted as in LIVE [21], we center-crop the images so the size fit the input layer of the network without concern for equal distribution of the distortion. We also evaluate a fine-tuned network. The training procedure of the fine-tuned network is the same as that of the non-fine-tune version, but in this case we let the gradient propagate through all layers.

Method	w/o finetuning	w/ finetuning	FPS
IQA-CNN+	-	0.820	166
IQA-CNN++	-	0.782	**250**
VGG-16	0.858	0.984	83
Atrous VGG-16	0.855	0.984	90
ResNet-101	0.926	**0.988**	20

Table 1: Quantitative results of classification accuracy and inference speed in distortion classification task. VGG- and ResNet-based models outperform IQA-CNN variants.

(a) quantization→fnoise (b) S&P→GWN

Figure 2: Example of mis-classified images. (a) has quantization distortion but is predicted as fnoise. (b) Salt and pepper noise classified as Gaussian white noise.

The result of the classification task is given in Table 1. To verify what effect pre-training has, we use IQA-CNN+ and IQA-CNN++[1] [12] as baseline. As can be seen in the results, the pre-trained networks outperform baseline networks. Since all pre-trained networks have deeper architectures compared to the baseline, they are suitable for complex data due to the high network capacity. Moreover, the fine-tuning procedure makes the network better-adapt to the new data.

Among non fine-tuned pre-trained networks, ResNet outperforms the VGGNet family. This is due to the output of VGGNet being 4096-dimensional, which is twice as large as that of ResNet.

However, all three networks show similar performance after being fine-tuned. We conjecture that this is probably because all networks have relatively large enough capacity to handle this task. Unlike the accuracy, inference time shows a large gap among different architectures. ResNet is much slower than the VGGNet family, and Atrous VGGNet is faster than the vanilla VGGNet, since Atrous VGGNet subsamples parameters in its final two fully-connected layers.

In most cases our model can classify distortion types very well. However, as in Fig. 2, some images are commonly mis-classified. For example, salt and pepper noise is often mistaken as Gaussian white noise as seen in Fig. 2b and vice versa. Fig. 2a shows a case where the image does not show abrupt change in color, in which case the model also confuses quantization noise with fnoise. Such problems can be alleviated if we directly compare the given image with a reference image, but in practice, there are restrictions on using reference images. Hence our approach well-balances between the accuracy and practicality.

[1]We re-implemented these networks using TensorFlow [1]. Note that we remove linear regression layer for predicting MOS.

4.2. Distortion Detection

In this section, we present the results on distortion detection experiment with SSD. Here, we only use the Atrous VGG-16 and ResNet-101 since VGG-16 and Atrous VGG-16 have similar performance but Atrous VGG-16 is faster. The IQA-CNN+ achieves reasonable result with very fast inference time, however, since this model only has single convolution layer, it is not appropriate to use the SSD that needs multiple convolution layers.

When training the network, we use the pre-trained CNN, which is fine-tuned over the *Classification* dataset, as a base network and stack SSD layer on top of the base network. In the evaluation step, we use the mean average precision (mAP) metric which measures the average precision of each class when the intersection over union (IoU) of the bounding box is one of $\{0.5, 0.75, 0.9\}$. In typical object detection tasks, performance evaluation is usually done with IoU @0.5. However, in our assumed scenario of finding the distortion regions for the purpose of applying local dynamic compression, finding accurate boxes is vital. This is why we use a variety of IoU thresholds to assess the degree of our algorithm's region detection accuracy. Result of experiments are in Table 2.

Method	FPS	mAP. IoU:		
		@ 0.5	@ 0.75	@ 0.9
ResNet-101	16	0.910	0.900	0.842
Atrous VGG-16	**63**	**0.919**	**0.906**	**0.873**

Table 2: Results of distortion detection (mAP) and inference speed over a variety of IoU threshold values. Atrous VGG performs slightly better than ResNet with higher FPS.

Surprisingly, there does not seem to be any advantages of using ResNet over VGGNet in this experiment. We believe that this is because VGGNet's capacity is large enough to fit to the given data. In addition, Atrous VGG-16 excels ResNet-101 in terms of inference time, and it can be done in real-time on state-of-the-art GPUs such as Maxwell TITAN X.

As described in Section 3.2, distortion may occur in small local regions in real world. Therefore, evaluating us-

Train data → Test data	mAP. IoU:		
	@0.5	@0.75	@0.9
basic → *basic*	0.919	0.906	0.873
difficult → *basic*	0.915	0.864	0.728
basic → *difficult*	0.717	0.467	0.109
difficult → *difficult*	0.908	0.895	0.785

Table 3: Transfer experiment results on detection task with Atrous VGG-16 network.

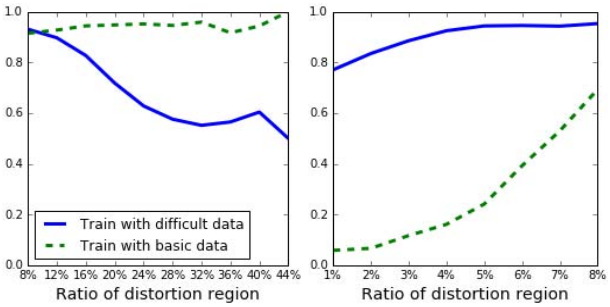
ing only the *basic* dataset might not be a desirable strategy. To further investigate, we conducted a set of transfer learning experiments that evaluates the four combinations of training-testing scenarios, where the training and testing datasets can be either *basic* or *difficult*. Table 3 shows that the models trained and tested on the same type of dataset yield the best performance. This is natural since the model trained using only the *basic* set cannot catch distortions with small region, while training only on the *difficult* set tends to drive the detector towards finding small regions. Note that in *basic* → *difficult* case, the trained model performs poorly when the IoU threshold is large, since it misses most small-size regions. The graph in Fig. 3 shows how the accuracy changes as the size of the ground-truth regions changes. This also illustrates that training with *basic* data is good when the area of the region is large, but performs worse for small sizes when trained on *difficult* data.

To sum up, it is crucial to match the settings between train and test data. But since we do not know much about the test set, it is better to train on the *difficult* set to better-cope with possible worst-case scenario based on the assumption that distortion may occur in small local region.

5. Conclusion

We investigated the novel problem of classifying and detecting distortion in an image without reference image using CNN architectures. To do that, we created a new *Flickr-Distortion* dataset to train on. In distortion classification, we used fine-tuned models that have been pre-trained on the ImageNet data, in order to reach convergence quickly. By doing so, we discovered that fine-tuned CNNs outperform other baseline models such as IQA-CNN+. Furthermore, we experimented the distortion detection task with the SSD [16] model, which has not been addressed previously. We found that our architecture is able to efficiently classify and detect various distortions, despite using a single set of weights for all distortion types.

We expect that our approach proves the usefulness of distortion detection in many applications such as dynamic compression technique or image reconstruction. One of our main discoveries is that the difference in the quality of training images and the testing images significantly affects the



(a) Test acc. with *basic* data (b) Test acc. with *difficult* data

Figure 3: Relationship between distortion size and detection accuracy with IOU @0.9. (a) shows test accuracy with *detection-basic* data and (b) is for *detection-difficult* scenario.

overall performance. This is not necessarily a surprising fact from a machine learning point-of-view, but it does have important practical implications. Since we do not know in advance the quality of the image we process, it might be difficult to guarantee the performance of our final system. We propose that we deploy our system after training on image sets consisting mostly of *difficult* images, in order to cope with the worst-case-scenario.

As a future work, we are planning to further develop our system to handle multiple distortions in a specialized manner. Our current system tries to classify and detect multiple distortions using a single structure. To account for multiple distortions, one must have a high-capacity system that could potentially lead to overfitting. If we could devise an ensemble-like system that specializes for each distortion type, the system might be able to focus on quality-neutral generalization within each distortion.

Acknowledgement

N.Ahn and K.-A. Sohn were supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education [NRF-2016R1D1A1B03933875], and B.Kang by [NRF-2016R1A6A3A11932796].

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016. 4

[2] S. Bianco, L. Celona, P. Napoletano, and R. Schettini. On the use of deep learning for blind image quality

assessment. *arXiv preprint arXiv:1602.05531*, 2016. 1

[3] S. Bosse, D. Maniry, T. Wiegand, and W. Samek. A deep neural network for image quality assessment. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3773–3777. IEEE, 2016. 1

[4] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005. 3

[5] L. Chen, S. Wang, W. Fan, J. Sun, and S. Naoi. Beyond human recognition: A cnn-based framework for handwritten character recognition. In *Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on*, pages 695–699. IEEE, 2015. 2

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016. 2

[7] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM international conference on image and video retrieval*, page 48. ACM, 2009. 3

[8] 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. 2

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014. 2

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, 2016. 1, 2

[11] L. Kang, P. Ye, Y. Li, and D. Doermann. Convolutional neural networks for no-reference image quality assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1733–1740, 2014. 1

[12] L. Kang, P. Ye, Y. Li, and D. Doermann. Simultaneous estimation of image quality and distortion via multi-task convolutional neural networks. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 2791–2795. IEEE, 2015. 1, 4

[13] 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. 2

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. 2

[15] F. Liu, G. Lin, and C. Shen. Crf learning with cnn features for image segmentation. *Pattern Recognition*, 48(10):2983–2992, 2015. 2

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. Ssd: Single shot multibox detector. *arXiv preprint arXiv:1512.02325*, 2015. 2, 5

[17] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti, et al. Image database tid2013: Peculiarities, results and perspectives. *Signal Processing: Image Communication*, 30:57–77, 2015. 3

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[19] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2

[20] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 2

[21] H. R. Sheikh, Z. Wang, L. Cormack, and A. C. Bovik. Live image quality assessment database release 2, 2005. 3

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 2

[23] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014. 3

[24] L. Zhang, L. Zhang, X. Mou, and D. Zhang. Fsim: a feature similarity index for image quality assessment. *IEEE transactions on Image Processing*, 20(8):2378–2386, 2011. 1