Image Classification of Blurred Images Without Deblurring

Jeehong Kim
Seoul National University
williamkim10@snu.ac.kr

Yeongin Kim
kyi8871@snu.ac.kr

Woohyun Han
hyhan1114@snu.ac.kr

Abstract

When a blur occurs in an image, the most basic and obvious way to solve the problem is to deblur the image. By deblurring the image, the blur is removed and the image is restored into a sharp state. Despite the good results they have produced in various image related tasks, there exists critical drawbacks including computational costs and the necessity for a given blur kernel size. Considering these limitations, we thought of a method that can utilize a raw blurred image as an input. The idea of using a relatively low-quality input despite the existence of a solution that can enhance the quality may seem unreasonable. Although the performance of using such input for various image-related tasks may not suffice compared to the usage of a deblurred image as an input, the trivial loss of accuracy in return of a huge decrease in computational cost will be a meaningful tradeoff. In this paper, we conduct various experiments to come up with a model that is robust to the blurs. Gathering up the results, we propose a framework that uses a proportion of the input image depending on the blur status and feed it to a existing ViT model. The proposed methods are evaluated on a single dataset.

1. Introduction

Blur exists in many digital images and is one of the typical factors that damages the quality of an image. It occurs due to many reasons including object motion, camera lens out of focus, and camera shake.

(a) (b)

Figure 1. (a) motion blurred image (b) out of focused image

For most of the time, such blurs are not desired and there have been many efforts to eradicate these blurs. The basic principle of a deblurring method is to deconvolute an image based on the blur kernel of the image [12]. Over many years, methods based on DNN have been adopted for deblurring and have created significantly good results [15]. However, there still exists an unsolved problem related to deblurring, which is the 'ill-posed problem'. Answer to deblurring is not unique, and this leads to many problems including heavy computational costs. Deblurring methods require many parameters in training such as blur kernel size. Motivated by this, we came up with an idea of using a raw blurred image as a direct input for a model. Of course, using a blurred image as an input will result in a relatively low accuracy compared to using a sharp image as an input, but having a huge computational decrease in return will be a meaningful tradeoff.

The model will be consisted of a dilated convolutional layer that studies particularly on the blurred features. As receptive size decreases as downsampling is proceeded throughout the CNN models, we decided that using a larger receptive size and extract meaningful features from the blurred images is important. With the features learned from the early layers of the model, we will aggregate the spatial features learned from the latter part of the model by tuning a existing CNN model. The experiment is done using a ImageNet Data(2012). We have set up our baseline with 2 methods : 1. Pretrained CNN Models on sharp ImageNet Data / 2. CNN Models trained on blurred ImageNet Data

It is expected that this approach will suggest a new approach for dealing with blurred images. Furthermore, it will make a huge contribution in terms that the model has brought successful results using low-quality input. Not having to render a blurred image into a high-resolution image reduces computational costs significantly and is applicable to domains that require real time image classification such as surveillance cameras.

2. Related Works

2.1. Image Deblurring

Image deblurring is a classic problem in low-level computer vision with the aim to recover a sharp image from a blurred input image. Before deep-learning based deblurring methods appeared, the classical approach was to formulate the task as an inverse filtering problem, where a blurred image is modeled as the result of the convolution with blur kernels, either spatially invariant or spatially varying [14]. Some early approaches assume that the blur kernel is known and adopt classical image deconvolution algorithms such as Lucy-Richardson or Weiner deconvolution.

As such classical approaches relied on the existence of the blur kernel, there were several problems: for example, if the camera rotated, more than one blur could occur in the image and such blurs cannot be explained with a single convolution kernel. As most of the images in the reality are consisted of such spatially varying blur, deep learning based deblurring methods have started to be put into use. It would use a framework where a blurred image is taken as an input and produces a deblurred image using a deblurring network. Recent advances of deep learning techniques have revolutionized the field of computer vision in many areas including image classification, object detection, video deblurring etc.

2.2. Blurred Region Detection

Blurred region detection is important in blurred image classification because it helps to identify the areas of an image that are blurred and distinguish them from the areas that are in focus.

Common approaches to blurred region detection were based on the estimation of local blur measures. I can be categorized into frequency-based, depth-based. Frequency-based studies such as [8] presented a method called singular value decomposition (SVD) based on single thresholding on image features to detect the blurred and nonblurred regions. In depth-based studies such as [5] presented different local features association like congruence, gradient histogram, and power bands to specify the type of blur from the images. These methods use measures such as gradient magnitude, Fourier spectrum, and Laplacian of Gaussian to estimate the local blur levels. However, these methods are prone to noise and may not be accurate in complex scenes.

In recent years, deep learning-based methods have shown promising results in blurred region detection. These methods utilize convolutional neural networks (CNNs) to learn effective representations of image features and classify pixels as either blurred or sharp. For example, in [3] introduced a deep learning method based on a CNN for the detection of sharp and blur regions of the image. And in [9] proposed a Deep Neural Network based technique Diffusion Network that fused the refined features extracted by the networks to obtain the segmented blur and sharp regions.

Recent deep learning-based methods approach this problem by learning an end-to-end mapping between the blurred input and a binary mask representing the localization of its blurred areas. Nevertheless, the effectiveness of such deep models is limited due to the scarcity of datasets annotated in terms of blur segmentation, as blur annotation is labor intensive.

2.3. Blur Type Classification

Blur type classification is an important task in computer vision that has received significant attention from researchers in recent years. One of the early works in blur type classification was proposed by Su and Grauman [2], who used hand-crafted features such as color, texture, and edge information to classify blur into motion blur, out-of-focus blur, and camera shake. They achieved an accuracy of 85

With the advent of deep learning, several researchers have proposed methods that use convolutional neural networks (CNNs) to automatically learn features for blur type classification. For instance, Kupyn et al. [4] used a CNN with a Siamese architecture to classify blur into motion blur, out-of-focus blur, and no blur. They achieved an accuracy of 98% on a dataset of 5000 images. More recently, Zhang et al. [9] proposed a multi-stream network that learns different features from different blur types. They also introduced a new dataset with four blur types: motion blur, out-of-focus blur, Gaussian blur, and unknown blur. Their method achieved an accuracy of 94.2% on this dataset.

In addition to these works, several researchers have also proposed methods that combine blur type classification with other tasks such as deblurring [11], image restoration [13], and object recognition [10]. These works highlight the potential applications of blur type classification in various computer vision tasks. Overall, the works discussed in this section demonstrate the importance and potential of blur type classification and provide a foundation for future research in this area.

3. Proposed Approach

In this section, we introduce a proposed framework that estimates the amount of blur in an image and eradicates the blurred part to reduce the amount of input used for a ViT model.

3.1. Approach Overview

Our method mainly adopts an existing Vision Transformer model. We basically train the model on blurred image dataset using existing CNN models. During the model inference, we first divide the image into a fixed-size patch.

After diving the image into small patches, we calculate the amount of blur occurred in each patch in order to decide whether a certain patch is viable as a meaningful input or not. We calculate the amount of blur in a float number, and if the number exceeds a certain threshold, the patch is excluded. The remaining patches will be used as the final input.

3.2. Blur Detection

The amount of blur is calculated using the variation of the Laplacian. The idea is simple: we take a single channel of an image and convolve it with the following 3 x 3 kernel.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Figure 2. Most commonly used discrete Laplacian matrix

The reason this method works is due to the definition of the Laplacian operator [7] itself, which is used to measure the 2nd derivative of an image. The Laplacian highlights regions of an image containing rapid intensity changes, much like the Sobel and Scharr operators [6]. And, just like these operators, the Laplacian is often used for edge detection. The assumption here is that if an image contains high variance, then there is a wide spread of responses, both edge-like and non-edge like, representative of a normal, in-focus image. But if there is very low variance, then there is a tiny spread of responses, indicating there are very little edges in the image.

As we know, the more an image is blurred, the less edges there are, meaning that it will have a lower variance of the Laplacian image, compared to sharp images. Obviously, the trick here is setting the correct threshold which can be quite domain dependent. Too low of a threshold can lead to incorrect marks of images as blurry when they are not. Too high of a threshold can produce errors where images that are actually blurry will not be marked as blurry. This method tends to work best in environments where you can compute an acceptable focus measure range and then detect outliers.

After conducting multiple experiments, we decided to set the threshold as 100. Patches that have a blur higher than 100 will no longer be put into use, and the remaining patches will go through the Transformer encoder. This process allows to reduce the computational cost by eradicating the parts in images where it's too blurred.

3.3. Vision Transformer

Images are first separated into small patches and are tokenized [1]. Then, these tokens are flattened and mapped to D dimensions with a trainable linear projection. The output of the projections is referred as the patch embeddings. Along with the patch embeddings, positional embeddings are added in order to give the positional information of each tokens, just like the original Transformer model. An additional token known as 'classification token' is added. This token does its role of 'classifying', and to successfully do its role, no biases of the image is included in this particular token. All of these are fed in a sequence as an input to a standard transformer encoder. After the model is pretrained on a huge dataset, it is finetuned on the downstream dataset for image classification.

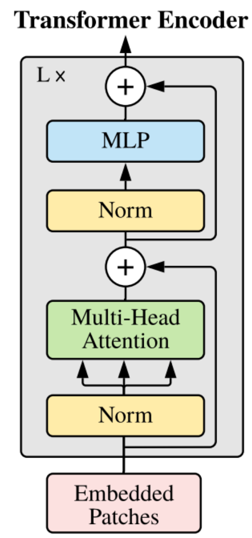

Figure 3. Vision Transformer Encoder

4. Experiment

4.1. Experimental Setting

Dataset: We conduct our experiment on the ImageNet dataset(2012). Due to limitations of computational resources, we decided to use only 20% of the ImageNet dataset(2012). We used both of the sharp and blurred versions of the images. For the blurred version, we adopted *GaussianBlur* method. The kernel size is *19 by 19* and the standard deviation to be used for creating kernel to perform blurring is chosen uniformly at random between *(1.0, 2.0)* In all experiments, we use the official train and validations splits for evaluation

Baselines: For the baseline, we adopted pretrained models on sharp image of ImageNet dataset. The result for the baseline is in Table 1 and Table 2 which shows the test accuracy for the sharp image set and the test accuracy for the blurred image set of the three models. The only difference is that Table 1 shows top-1 accuracy, and Table 2 shows top-5 accuracy.

Method	Test With Sharp Image	Test With Blurred Image
Resnet50	76.13%	63.16%
VGG19	72.38%	54.12%
GoogleNet	69.78%	55.03%

Table 1. Top-1 Accuracy

Method	Test With Sharp Image	Test With Blurred Image
Resnet50	92.86%	85.08%
VGG19	90.88%	78.19%
GoogleNet	89.53%	79.14%

Table 2. Top-5 Accuracy

4.2. Blurred Image Training

We trained each CNN model using the blurred images from the ImageNet dataset. The objective was to determine whether training on blurred images would result in improved classification performance for blurred images.

Data Preprocessing: Prior to training, we applied the same blurring process mentioned earlier (using the *GaussianBlur* method with a kernel size of 19 by 19 and a uniformly random chosen standard deviation between 1.0 and 2.0) to create the blurred version of the ImageNet dataset. Additionally, we create a mixture of blurred and sharp images(50% each) for the training dataset. We then used this datasets for training the models.

Training Procedure: For each model (Resnet50, VGG19, and GoogleNet), we initialized the network weights randomly and trained it using the blurred image dataset and mixture of blurred image and sharp image dataset. We employed the same training hyperparameters as in the baseline experiment to ensure consistency and fair comparison.

Evaluation: After training the models on the blurred image dataset, we evaluated their performance on the same test sets used in the baseline experiment. This allowed us to directly compare the performance of the models when trained on blurred images or mixture of blurred images and sharp images versus when trained on sharp images.

Results: Table 3 and Table 4 present the test accuracy of the models trained on blurred images and trained on mixture of blurred images and sharp images for both top-1 and top-5 accuracies, respectively.

Method	Train Blurred Image	Train Blur+Sharp Image
Resnet50	58.69%	61.24%
VGG19	46.33%	48.19%
GoogleNet	48.97%	51.62%

Table 3. Top-1 Accuracy - Test With Blurred Image

Method	Train Blurred Image	Train Blur+Sharp Image
Resnet50	80.12%	83.72%
VGG19	71.86%	73.55%
GoogleNet	73.42%	76.98%

Table 4. Top-5 Accuracy - Test With Blurred Image

Discussion: From the results of the second experiment, it can be observed that training the CNN models on the blurred images led to a decrease in classification performance for blurred test image sets. The top-1 and top-5 accuracies obtained were consistently lower compared to the models trained on sharp images, as shown in Tables 3 and 4.

This decrease in performance can be attributed to the loss of fine details and reduced clarity in the blurred images, which makes it more challenging for the models to learn discriminative features. Additionally, the blurring process introduces noise and distortions that negatively affect the model's ability to generalize to unseen images.

Additionally, it is evident that training the CNN models on a mixed dataset of 50% blurred images and 50% sharp images yields better classification performance for blurred test images compared to training solely on blurred images. The top-1 and top-5 accuracies obtained are consistently higher than the models trained on only blurred images, as shown in Tables 3 and 4. This improvement can be attributed to the inclusion of sharp images in the training dataset. By incorporating sharp images, the models have access to rich and detailed information, enabling them to learn more discriminative features.

Overall, these findings suggest that training CNN models directly on blurred images is not effective in improving the classification performance for blurred test images. The sharp image features appear to be essential for learning robust representations that generalize well to both sharp and blurred images.

4.3. Blurred Image Classification using Vision Transformer (ViT)

After observing that training CNN models directly on blurred images did not improve classification performance for blurred test images, we turned our attention to Vision Transformers (ViT) as an alternative approach.

In this experiment, our goal was to leverage ViT models to classify blurred images by detecting and removing blurry patches. The intuition behind this approach was that if an image is blurry, the corresponding patches are likely to contain less useful information for classification. By selectively removing these blurry patches, we aimed to improve the overall classification performance for blurred images.

Blur Detection and Patch Removal: For each image in the test dataset, we applied a Gaussian blur treatment to introduce blur in various regions of the image. Subsequently, we used a blur detection algorithm to identify the patches within the image that exhibited high levels of blur. Specifically, we calculated blur scores for each patch and considered patches with scores above a certain threshold as blurry patches. We calculated the blur scores for each patch using the variance of Laplacian metric, a widely-used blur detection technique

To remove the identified blurry patches, we masked out these regions by setting their values to zero. This process ensured that the ViT model would not utilize information from these patches during classification.

Testing Procedure: We evaluated the performance of the ViT model on the blurred images by comparing two scenarios: (1) Classifying the original blurred images without removing any patches, and (2) Classifying the blurred images with the identified blurry patches removed.

For both scenarios, we computed the classification accuracy on the test dataset and compared the results to assess the impact of patch removal on the model's ability to classify blurred images.

Results: Table 5 presents the classification accuracies of the ViT model for both scenarios: (1) Classifying the original blurred images, and (2) Classifying the blurred images with the identified blurry patches removed.

Method	Original Blurred Image	Patches Removed
ViT	58.04%	51.75%

Table 5. Blurred Image Classification Results - ViT Model

Discussion: The results of Table 5 indicate that using Vision Transformers (ViT) for blurred image classification by detecting and removing blurry patches did not improve the classification performance compared to classifying the original blurred images without patch removal.

This indicates that simply removing blurry patches from an image did not significantly enhance the ViT model's ability to classify blurred images accurately. While patch deletion eliminates some of the blurriness, it may also remove useful information and context from the image, hindering the model's ability to make accurate predictions.

These findings suggest that for blurred image classification, alternative approaches beyond patch deletion are required. The limitations of solely relying on patch-level blur detection and removal highlight the need for developing more sophisticated techniques that can effectively handle blurriness in images during training.

5. Conclusion

In this paper, we introduced a new approach of dealing with blurred images. Rather than deblurring an image, we came up with an idea of adopting ViT model for image classification. In order to reduce the computational cost, we tried to eradicate the blurred part of the image by first dividing the image into small patches and abandoned the blurred part depending on the threshold. The results showed that removing the blurred patches did not significantly enhance the performance of the preexisting ViT model. There is a definite need for further investigation on how to remove the blurred part of the image while minimizing the loss of useful information of an image.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[2] Kristen Grauman and Bastian Leibe. Local features: detection and description. In *Visual Object Recognition*, pages 11–25. Springer, 2011. 2

[3] Beomseok Kim, Hyeongseok Son, Seong-Jin Park, Sunghyun Cho, and Seungyong Lee. Defocus and motion blur detection with deep contextual features. In *Computer Graphics Forum*, volume 37, pages 277–288. Wiley Online Library, 2018. 2

[4] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8183–8192, 2018. 2

[5] Renting Liu, Zhaorong Li, and Jiaya Jia. Image partial blur detection and classification. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008. 2

[6] EePing Ong, Weisi Lin, Zhongkang Lu, Xiaokang Yang, Susu Yao, Feng Pan, Lijun Jiang, and Fulvio Moschetti. A no-reference quality metric for measuring image blur. In *Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings.*, volume 1, pages 469–472. Ieee, 2003. 3

[7] George E Sotak Jr and Kim L Boyer. The laplacian-of-gaussian kernel: a formal analysis and design procedure for fast, accurate convolution and full-frame output. *Computer Vision, Graphics, and Image Processing*, 48(2):147–189, 1989. 3

[8] Bolan Su, Shijian Lu, and Chew Lim Tan. Blurred image region detection and classification. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1397–1400, 2011. 2

[9] Chang Tang, Xinzhong Zhu, Xinwang Liu, Lizhe Wang, and Albert Zomaya. Defusionnet: Defocus blur detection via recurrently fusing and refining multi-scale deep features. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2700–2709, 2019. 2

[10] Zhonghe Tang, Shengzhe Wang, Jianliang Huo, Hang Guo, Haibo Zhao, and Yuan Mei. Bayesian framework with non-local and low-rank constraint for image reconstruction. In *Journal of Physics: Conference Series*, volume 787, page 012008. IOP Publishing, 2017. 2

[11] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8174–8182, 2018. 2

[12] Fagun Vankawala, Amit Ganatra, and Amit Patel. A survey on different image deblurring techniques. *International Journal of Computer Applications*, 116(13):15–18, 2015. 1

[13] Qingxiong Yang, Kar-Han Tan, and Narendra Ahuja. Real-time o (1) bilateral filtering. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 557–564. IEEE, 2009. 2

[14] Kaihao Zhang, Wenqi Ren, Wenhan Luo, Wei-Sheng Lai, Björn Stenger, Ming-Hsuan Yang, and Hongdong Li. Deep image deblurring: A survey. *International Journal of Computer Vision*, 130(9):2103–2130, 2022. 2

[15] Shangchen Zhou, Jiawei Zhang, Jinshan Pan, Haozhe Xie, Wangmeng Zuo, and Jimmy Ren. Spatio-temporal filter adaptive network for video deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2482–2491, 2019. 1