

Control the Information of the Image with Anisotropic Diffusion and Isotropic Diffusion for the Image Classification

Hyun-Tae CHOI^a, Nahyun LEE^a, Jewon NO^a, Sangil HAN^a, Jaeho TAK^a,
Hwijin KIM^a, Haegang LEE^a, Seonghoon HAM^a and Byung-Woo HONG^{a,1}

^aComputer Science Department, Chung-Ang University, Seoul, Korea

Abstract. Humans can recognize objects well even if they only show the shape of objects or an object is composed of several components. But, most of the classifiers in the deep learning framework are trained through original images without removing complex elements inside the object. And also, they do not remove things other than the object to be classified. So the classifiers are not as effective as the human classification of objects because they are trained with the original image which has many objects that the classifier does not want to classify. In this respect, we found out which pre-processing can improve the performance of the classifier the most by comparing the results of using data through other pre-processing. In this paper, we try to limit the amount of information in the object to a minimum. To restrict the information, we use anisotropic diffusion and isotropic diffusion, which are used for removing the noise in the images. By using the anisotropic diffusion and the isotropic diffusion for the pre-processing, only shapes of objects were passed to the classifier. With these diffusion processes, we can get similar classification accuracy compared to when using the original image, and we found out that although the original images are diffused too much, the classifier can classify the objects centered on discriminative parts of the objects.

Keywords. isotropic diffusion, anisotropic diffusion, classification, deep learning

1. Introduction

Even if the object is composed with many components or only the shape of the object is given, human can recognize what the object is. For the deep learning classification network, it is not easy to classify the object when only the object's shape is given, since most of the classifiers are trained with the original image. When we visualize the classification deep learning network with convolutional neural network, the network's layer which closer to an input layer, the more fined information is used and the last layer use more grained information [1]. Therefore, in order to accurately classify the object, the deep learning network uses both the shape and details of the object. In this respect, we want to find out how much object's information the network needed to classify the object. The simplest way to limit the amount of information is the diffusion. To diffuse the input

¹Corresponding Author: Byung-Woo HONG, Computer Science Department, Chung-Ang University, Seoul, Korea; E-mail: hong@cau.ac.kr.

images, we use anisotropic diffusion [2] and isotropic diffusion. With this two diffusion process, we can limit the information of the objects by removing the artifacts inside of the objects. Also, with the diffused images, we can get similar classification accuracy compare to using the original images. In the remainder of this paper, we briefly present the two diffusing processes in Section 2 and then explain the our method for diffusion in Section 3. Next, we show our classification results in Section 4. Finally, we summarize our paper in Section 5.

2. Isotropic Diffusion and Anisotropic Diffusion

Isotropic diffusion and anisotropic diffusion algorithm reduce the noise in the image with a partial differential equation. The isotropic diffusion is the solution of the heat equation. When we set $I(x, y, t)$ is the image at time t and the input image as I , the definition of the isotropic diffusion is Eq.(1).

$$\frac{\delta I(x, y, t)}{\delta t} = \Delta I \quad (1)$$

In Eq.(1), Δ is the laplacian operator which is the inner product of ∇ ; $\Delta = \nabla \cdot \nabla$ and ∇ is the gradient operator. But this isotropic diffusion algorithm diffuses the image uniformly in all directions, so when we apply the isotropic diffusion to the input image, it blurs too much that we cannot recognize the image. But unlike isotropic diffusion, anisotropic diffusion algorithm diffuses the image with preserving the edge of the objects. The definition of the anisotropic diffusion is Eq.(2).

$$\frac{\delta I(x, y, t)}{\delta t} = \text{div}(c(x, y, t)\nabla I) \quad (2)$$

Figure 1 shows the several diffused images. The kernel size of (a), (b) and (c) means the size of the gaussian kernel for the isotropic diffusion. When we apply large-sized kernel and big standard deviation, the gaussian kernel diffuses the image much more than the small-sized kernel. And the diffusion coefficient of (d), (e) and (f) is the α of the Eq. 3. (a), (b) and (c) are the isotropic diffused image and (d), (e) and (f) are the anisotropic diffused image. (c) is the much more diffused than (a) and (f) is much more diffused than (d). With (a), (b) and (c), we can know that when we diffuse the image with higher coefficient, we cannot recognize the image. However, with (d), (e) and (f), when we diffused the image with more higher coefficient, we can still recognize the object, because the anisotropic diffusion algorithm diffuses the image with preserving the edge of the objects.

3. Method

For the isotropic diffusion, we apply the gaussian kernel. When we use the small-sized and small standard deviation, the kernel diffuses the image little. On the contrary, with the big size kernel and big standard deviation, the kernel diffuses the image a lot. We use 8 kernel size; 3, 9, 15, 21, 27, 33, 39, 45. And set the standard deviation as twice of

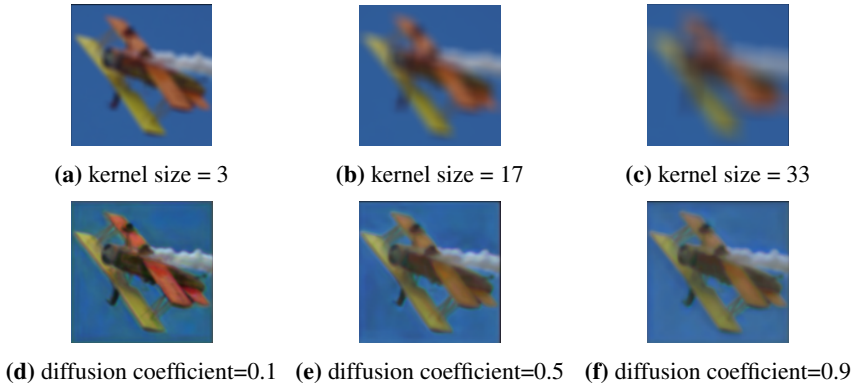


Figure 1. (a), (b) and (c) show the isotropic diffused image and (d), (e) and (f) show the anisotropic diffused image.

each kernel size. We apply the isotropic diffusion before the classification network and the classification network is trained with the results of the isotropic diffusion.

Also, for the anisotropic diffusion, we make an auto-encoder with VGG-16 [3]. The encoder part is same as the VGG-16 except the fully connected layers and the decoder part is composed of the transposed convolution layers. And Eq.3 is used for the loss of the auto-encoder.

$$\begin{aligned}
 Loss &= \|U - I\|^2 + \alpha \|\nabla U\| \\
 \|\nabla U\| &= \|\nabla U_x\| + \|\nabla U_y\|
 \end{aligned}
 \tag{3}$$

In Eq.3, I is the input and the U is the output of the anisotropic diffusion auto-encoder. α is the diffusion coefficient and ∇ is the gradient. So, ∇U_x is a differential U in the x-axis direction and ∇U_y is a differential U in the y-axis direction. We apply the anisotropic diffusion as an auto-encoder network, before the classification network. So, the classifier is trained with U instead of the original input image I . For the diffusion coefficient α , we use 7 diffusion coefficient; 0.001, 0.01, 0.1, 0.3, 0.5, 0.5, 0.9. In Eq.3, the first term of the $Loss$ is the data fidelity that control the difference between the output and the input image. And the second term of the $Loss$ is the total variation that diffuse the output image.

We use STL-10 [4] for the dataset and use VGG-16 [3], VGG-16 with batch normalization [5], VGG-19, VGG-19 with batch normalization, RESNET-18 [6] and RESNET-34. Also, we observe the class activation map [7] for each diffusion process and for each diffusion coefficient with the RESNET-34.

4. Results

To observe the two diffusion algorithms' influence to the information of the images, we check the classification score and class activation map. For the isotropic diffusion, we change the kernel size, and for the anisotropic diffusion, we change the α in Eq.(3). The classification score is shown with Table 1. For the classification, we use 6 structures with

softmax; VGG-16, VGG-16_bn, VGG-19, VGG-19_bn, RESNET-18, RESNET-34. The VGG-16_bn means the VGG-16 with batch normalization and the VGG-19_bn means VGG-19 with batch normalization. We apply two diffusion algorithms; isotropic diffusion and anisotropic diffusion. For anisotropic diffusion, we use 8 kernel for each experiment. We represent the kernel size in the Table 1 as isotropic_kernel size (e.g. isotropic_3 means isotropic diffusion when the kernel size is 3.). And we represent the diffusion coefficient (α in Eq.3) as aniso_diffusion coefficient (e.g. aniso_0.001 means anisotropic diffusion when the diffusion coefficient is 0.001.).

Table 1. classification accuracy(%) per each classifier, isotropic diffusion with 8 kernels size, and anisotropic diffusion with 7 diffusion coefficients. The **bn** means the batch normalization. We represent the kernel size of isotropic diffusion as **isotropic_size** and the diffusion coefficient as **aniso_coefficient**.

	VGG-16	VGG-16_bn	VGG-19	VGG-19_bn	RESNET-18	RESNET-34
original	95.06	93.96	91.53	94.36	95.82	97.16
isotropic_3	94.41	93.83	90.06	94.22	94.15	96.93
isotropic_9	89.25	86.98	83.35	88.07	89.51	93.4
isotropic_15	83.58	79.50	76.25	81.50	86.58	87.52
isotropic_21	79.21	74.08	73.07	74.56	81.06	82.86
isotropic_27	75.37	70.58	68.30	70.96	76.01	76.11
isotropic_33	71.15	67.12	68.53	67.93	68.80	70.86
isotropic_39	67.00	64.58	64.98	66.30	67.16	67.17
isotropic_45	61.56	63.05	62.23	63.56	61.42	64.82
aniso_0.001	92.33	91.51	89.08	92.57	93.57	95.05
aniso_0.01	92.50	91.85	88.57	92.53	93.53	94.88
aniso_0.1	91.42	91.30	88.07	92.15	93.22	94.58
aniso_0.3	90.28	89.06	86.96	90.53	92.28	93.57
aniso_0.5	89.31	88.56	85.00	89.55	92.05	92.93
aniso_0.5	89.05	87.56	83.45	88.21	90.85	92.18
aniso_0.9	87.57	86.53	82.13	87.22	90.30	91.15

As we can see with Table 1, the isotropic diffusion algorithm has a negative effect on classification as diffusion degree becomes larger(i.e. the bigger kernel size). But, even if we apply the anisotropic diffusion with the properly large diffusion coefficient, we can get still high classification score. It means that the isotropic diffusion cannot preserve the information of the objects, but the anisotropic diffusion can preserve the information of the objects. This can also be seen through Figure 1. Also, Figure 2 show the class activation map of each diffusion algorithm and diffusion coefficient with RESNET-34.

In the Figure 2, each of the image sets, the top 4 images is the diffused images and the bottom 4 images is the class activation maps. **aniso_***k* represent the anisotropic diffusion with diffusion coefficient *k*(e.g. aniso_0.001 means anisotropic diffusion when the diffusion coefficient is 0.001). **iso_***k* represent the isotropic diffusion with gaussian kernel size *k*(e.g. iso_3 means isotropic diffusion when the gaussian kernel size is 3). When we compare the each results with the original image results (*a*), when we apply the anisotropic diffusion with various diffusion coefficient (*b*), (*c*), (*d*), (*e*), (*f*), (*g*), (*h*) , the class activation maps do not change much. But, with the isotropic diffusion results (*i*), (*j*), (*k*), (*l*), (*m*), (*n*), (*o*), (*p*), we can see that the class activation maps get bigger as the kernel size gets bigger (i.e. much more diffusion). With, Table 1 and Figure 2, we

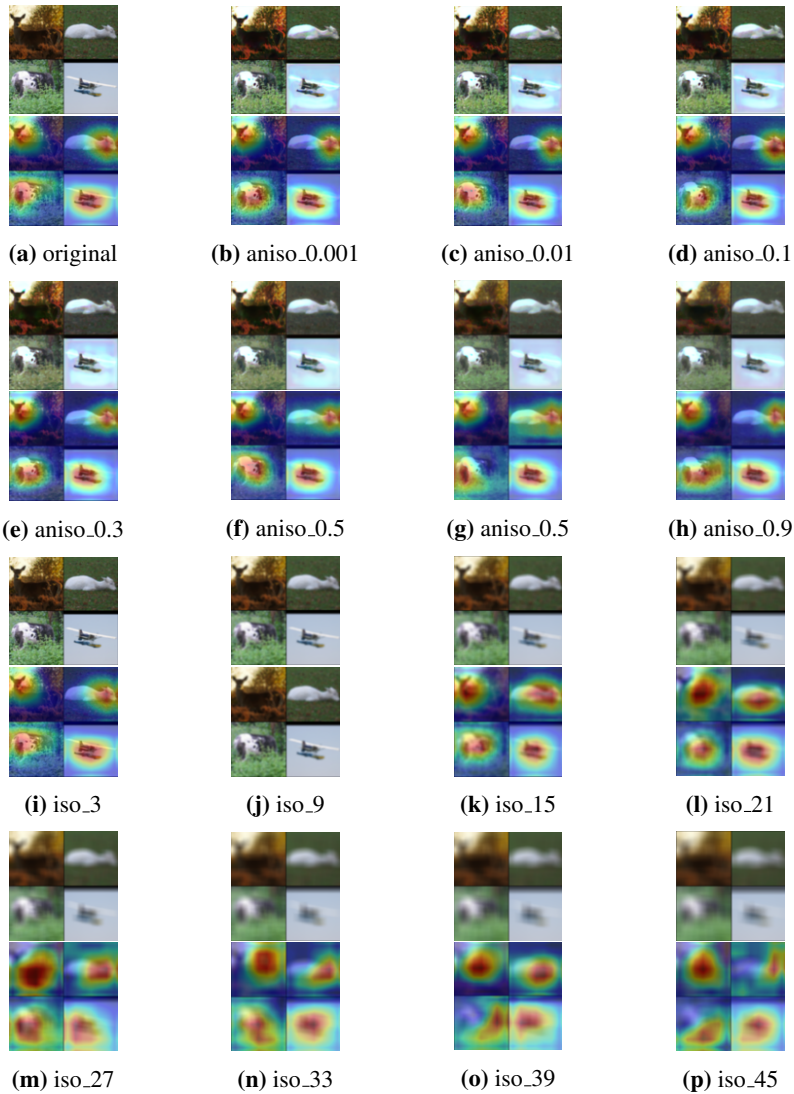


Figure 2. Diffused image and class activation map of each diffusion algorithm and diffusion coefficient. **aniso** $_k$ represent the anisotropic diffusion with diffusion coefficient k (e.g. aniso_0.001 means anisotropic diffusion when the diffusion coefficient is 0.001). **iso** $_k$ represent the isotropic diffusion with gaussian kernel size k (e.g. iso_3 means isotropic diffusion when the gaussian kernel size is 3).

can know that when we apply the properly large diffusion coefficient with anisotropic diffusion to the input image, the classifier can get similar classification accuracy even though the images are diffused. Also, with anisotropic diffusion we can decrease the noise of the image. Figure 3 shows the effect of anisotropic diffusion on the action of reducing noise of the image. In the Figure 3, the patterns on the wallpaper disappear.



(a) original image



(b) anisotropic diffused image

Figure 3. (a) is the original image and (b) is the anisotropic diffused image.

5. Conclusion

When the input image is diffused, there is a loss of information about the object. But if the object's shape is remained, the classification network still can classify the object. Two diffusion algorithm isotropic diffusion and anisotropic diffusion diffuse the image, but the isotropic diffusion cannot preserve the shape of the object well. On the other hand, anisotropic diffusion algorithm can preserve the edge that can preserve the shape of the object. With our experiment, the anisotropic diffusion algorithm is applied effectively to preserve the shape of the object. When we compare the anisotropic diffused images' classification score to original images, we can get similar classification score also, can get much more higher accuracy than using isotropic diffused images.

Acknowledgment

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the National Program for Excellence in SW(20170001000051001) supervised by the IITP(Institute of Information & communications Technology PlanningEvaluation) in 2021

References

- [1] Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. European conference on computer vision. D. Fleet et al. (Eds.): ECCV 2014, Part I, LNCS 8689, 2014; pp. 818–833.
- [2] Perona P, Malik J. Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence, July 1990; 12(7): 629-639, doi: 10.1109/34.56205.
- [3] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv, 2014; abs/1409.1556.
- [4] Coates A, Ng A, Lee H. An analysis of single-layer networks in unsupervised feature learning. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, in Proceedings of Machine Learning Research, 2011; 15:215-223 Available from <https://proceedings.mlr.press/v15/coates11a.html>.
- [5] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on Machine Learning, in Proceedings of Machine Learning Research, 2015; 37:448-456 Available from <https://proceedings.mlr.press/v37/ioffe15.html>.
- [6] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016; 770-778, doi: 10.1109/CVPR.2016.90.

- [7] Zhou BL, Khosla A, Lapedriza A, Oliva A and Torralba A. Learning deep features for discriminative localization. Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition. 2016; 2921-2929.