Report 04

Yiming Bian

Jun 3rd 2022

Major Professor

Arun Somani(arun@iastate.edu)

Committee Members

Henry Duwe (duwe@iastate.edu)
Aditya Ramamoorthy (adityar@iastate.edu)
Alexander Stoytchev (alexs@iastate.edu)
Cindy Yu (cindyyu@iastate.edu)

1 Introduction

In this report, we will focus on the salt and pepper(SNP) noise. The aim is to improve the prediction accuracy on images with SNP noise for given models. The competition is between our retrained model and its mighty rival, median filter.

2 Introduction

2.1 Retrained models

As stressed in the previous reports, we have five retrained models on SNP noise images, namely **RtN_0.1**, **RtN_0.2**, **RtN_0.3**, **RtN_0.4**, and **RtN_0.5**. All of which were based on a pretrained model, then they were retrained using SNP noise images at the corresponding noise level as shown in Table 1. To have a more straightforward sense of SNP noise, we apply each level of SNP noise on an original sample image in Fig. 1. As the noise density increases, the image is harder to be recognized, to both human and neural networks.

Table 1: Information of retrained models: the number in the model name is the density of SNP noise. e.g. **RtN_0.1** stands for the model retrained on images with SNP noise of density 0.1. The pretrained model is available to public in libraries such as Tensorflow and Pytorch. In this project, we choose the implementation in Pytorch. There are several numbers in the training set configuration column, 1040, 130 and 5 stand for the number of images for training, validation and the number of objects. Moreover, five objects are hamster, goldfinch, upright, pitcher and vizsla. All of the original images are from ImageNet and you may notice in the first two reports, one of the objects was frying pan instead of vizsla. The reason of the substitution is that the number of images of frying pan in ImageNet is 1222, while the rest, including vizsla, is 1400 and we would like to keep the consistency of the number of each object for more convenient noise data set construction.

Model Name	Based on	Training set configuration
RtN_0.1	Pretrained	(1040 + 130)×5 images with SNP_0.1 noise
RtN_0.2	Pretrained	(1040 + 130)×5 images with SNP_0.2 noise
RtN_0.3	Pretrained	(1040 + 130)×5 images with SNP_0.3 noise
RtN_0.4	Pretrained	(1040 + 130)×5 images with SNP_0.4 noise
RtN_0.5	Pretrained	(1040 + 130)×5 images with SNP_0.5 noise

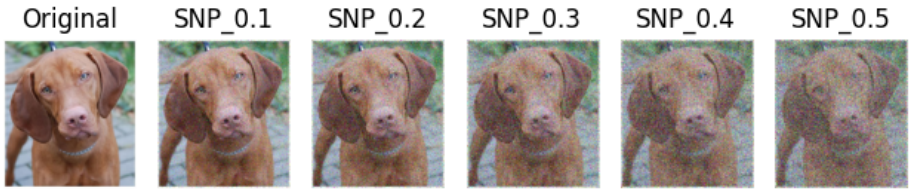

Figure 1: The original image and the corresponding noise images with multiple SNP noise densities: the original image is "n02100583_516.JPEG" under object "vizsla" from ImageNet

2.2 Median filter

To find a mighty rival of our retrained models, we challenge the mighty median filter. It is widely-acknowledged as one of the best solutions for SNP noise removal as it is simple and removes the noise while keeps the edge at the same time. To explain median filter, we can simplify an image as a 2D array of integer numbers, each is between 0 and 255. SNP noise is noise pixels that has a value of either 0 or 255 as 0 stands for black pixel and 255 stands for white pixel, it resembles salt(while) and pepper(black). A detailed explanation is provided in Fig. 2.

123	125	126	130	140	145
121	122	116	134	111	140
120	108	122	136	145	122
119	115	125	124	146	117
118	165	130	122	140	154
111	125	106	114	175	163

123	125	126	130	140	145
121	122	**255**	134	111	140
120	108	122	136	145	**0**
119	115	125	**255**	146	117
118	**0**	130	122	**255**	154
111	125	106	114	175	163

123	125	126	130	140	145
121	122	255	134	111	140
120	108	122	136	145	0
119	115	125	255	146	117
118	0	130	122	255	154
111	125	106	114	175	163

Figure 2: Sample image, image with SNP noise and noise image applying a 3×3 median filter kernel: **Left** is a sample image with 6×6 pixels, each pixel has a value between 0 and 255. **Middle** is the original image with several SNP noise pixels, which are squared out with black boxes. Each noise pixel has a value of either 0 or 255. **Right** is applying a 3×3 median filter kernel and this kernel is at the initial position. It returns the median among the numbers in the kernel. For example, in the current position, the median of 123, 125, 126, 121, 122, 255, 120, 108 and 122 is 122 and it is the output of the current kernel. Then it will slide right and when this row is done, it will go the next row, which is sliding one row down and starts from the left-most position. While applying this kernel, it sacrifices details for the noise pixel removal. In the current kernel, it loses all pixel values including the noise pixel except the median value 122. Therefore, the output of a kernel will never be 0 nor 255 unless all the numbers in the kernel is 0 or 255. However, white/black noise pixels in the while/black background are invisible hence trivial.

To have a clear sense of how median filter fixes the SNP noise images, we provide three examples of applying median filter with different kernel sizes on three levels of SNP noise images. They are shown in Fig.3, 4 and 5. All images after applying median filter tend to be more vague as we mentioned in the caption of Fig.2, median filter sacrifices details for noise removal. Moreover, larger kernel size has a better noise removal ability while it adds more blur to the result image. This is the second trade-off.

Figure 3: Applying median filter with kernel size 3, 5, 7 and 9 on an SNP_0.1 noise image of a vizsla

Figure 4: Applying median filter with kernel size 3, 5, 7 and 9 on an SNP_0.3 noise image of a vizsla

Figure 5: Applying median filter with kernel size 3, 5, 7 and 9 on an SNP_0.5 noise image of a vizsla

3 Experiment results and analysis

We did the following experiment to compare the prediction accuracy between model **RtO** on restored five levels of SNP noise images using median filter of four kernel sizes and five retrained **RtN** models. The results show the superiority of each method in different scenarios. The details are as follows.

We took six test sets: original, SNP_0.1, SNP_0.2, SNP_0.3, SNP_0.4 and SNP_0.5. The configuration of each is shown in Table.2.

Table 2:

Test set	Contents	Configuration
Original	Only original images	130×5 = 650
SNP_0.1	Only images with SNP_0.1 noise	130×5 = 650
SNP_0.2	Only images with SNP_0.2 noise	130×5 = 650
SNP_0.3	Only images with SNP_0.3 noise	130×5 = 650
SNP_0.4	Only images with SNP_0.4 noise	130×5 = 650
SNP_0.5	Only images with SNP_0.5 noise	130×5 = 650

To test the restoration effect of median filters, we take an noise image then apply median filter and test the prediction accuracy of RtO, to see if the model that has never seen any noise image(RtO) can recognize the restored noise image. To deal with multiple noise densities, we test median filters with multiple kernel sizes, thus 3, 5, 7 and 9. When the kernel size is larger than 9, the image becomes very blurry and hard to recognize because the input image size is 224×224×3. The performance of RtO models on original image test set and five levels of SNP noise image test set together with the noise image sets after four levels of median filter restorations are listed in Table.3. Very obvious improvement can be observed and median filter justifies its popularity when handling SNP noise images.

To test the prediction accuracy of our retrained models, we simply test them on the original test set and five SNP noise sets, without any restoration. The results are shown in Table 4 and 5.

We evaluate two methods by comparing their best and worst prediction accuracy on noise sets, together with the initial RtO model. The results are shown in Fig.6. Combining the results above, when the noise level is low, median filter is the preferred method to improve the prediction accuracy. However, when the noise level is high, retrained models is a better solution.

4 Plan for next two weeks

Since Jun 2nd, I started working on a paper that finalized my recent work on object recognition in SNP noise images and prepared to submit it to IDSTA2022. It will be a 6-to-8-page paper and the submission deadline is Jun 15th. I will finish the initial version by the next Monday(Jun 6th) and work with Professor Somani on the revision and polish work.

Table 3: Prediction accuracy of RtO models on multiple test sets and restored noise sets: figure 3,5,7,9 stands for kernel size of the applied median filter. e.g. "SNP_0.1 | 3 | 99.077 | 98.769" means "after applying median filter of kernel size of 3 on SNP noise images of 0.1 noise density, the prediction accuracy of RtO ResNet-18 model is 99.077% and that of RtO VGG16 model is 98.769%". Moreover, red and blue mark the best and worst accuracy by applying median filters with different kernel sizes

Test set		**ResNet-18**	**VGG16**
Original		98.615	99.538
SNP_0.1	**-**	96	96
	3	99.077	98.769
	5	98.462	98.615
	7	97.538	97.846
	9	96.308	97.077
SNP_0.2	**-**	91.077	91.231
	3	95.385	96.769
	5	95.231	96.308
	7	95.077	95.077
	9	94.615	93.538
SNP_0.3	**-**	80.923	76.923
	3	88	90
	5	91.077	91.385
	7	91.692	91.385
	9	91.077	91.385
SNP_0.4	**-**	67.846	63.538
	3	80	75.692
	5	80.615	80.154
	7	82.462	80.154
	9	83.538	80.154
SNP_0.5	**-**	56.769	52.769
	3	61.538	62.769
	5	69.385	66.923
	7	74.154	64.769
	9	78.154	62

Table 4: Test accuracy of RtO and retrained RtN ResNet-18 models: red and blue mark the best and worst accuracy of retrained models

test set \ model	**RtO**	**RtN_0.1**	**RtN_0.2**	**RtN_0.3**	**RtN_0.4**	**RtN_0.5**
I Original	98.615%	75.538%	52%	45.538%	37.231%	27.846%
III SNP_0.1	96%	97.385%	83.692%	75.846%	55.692%	36.462%
IV SNP_0.2	91.077%	95.077%	93.231%	88.308%	73.077%	51.692%
V SNP_0.3	80.923%	89.077%	91.846%	90%	82.769%	62%
VI SNP_0.4	67.846%	83.231%	90%	88.769%	85.077%	76%
VII SNP_0.5	56.769%	72.923%	84.615%	83.231%	85.385%	80.615%

Table 5: Test accuracy of RtO and retrained RtN VGG16 models: red and blue mark the best and worst accuracy of retrained models

test set \ model	**RtO**	**RtN_0.1**	**RtN_0.2**	**RtN_0.3**	**RtN_0.4**	**RtN_0.5**
I Original	99.538%	99.385%	98.923%	99.077%	97.385%	80%
III SNP_0.1	96%	98.462%	98.154%	97.538%	95.077%	87.692%
IV SNP_0.2	91.231%	96%	95.692%	95.231%	91.692%	87.385%
V SNP_0.3	76.923%	88%	90.769%	91.077%	89.846%	86.615%
VI SNP_0.4	63.538%	75.846%	81.385%	83.077%	82.462%	79.385%
VII SNP_0.5	52.769%	62.462%	70.308%	74.615%	78.153%	75.538%

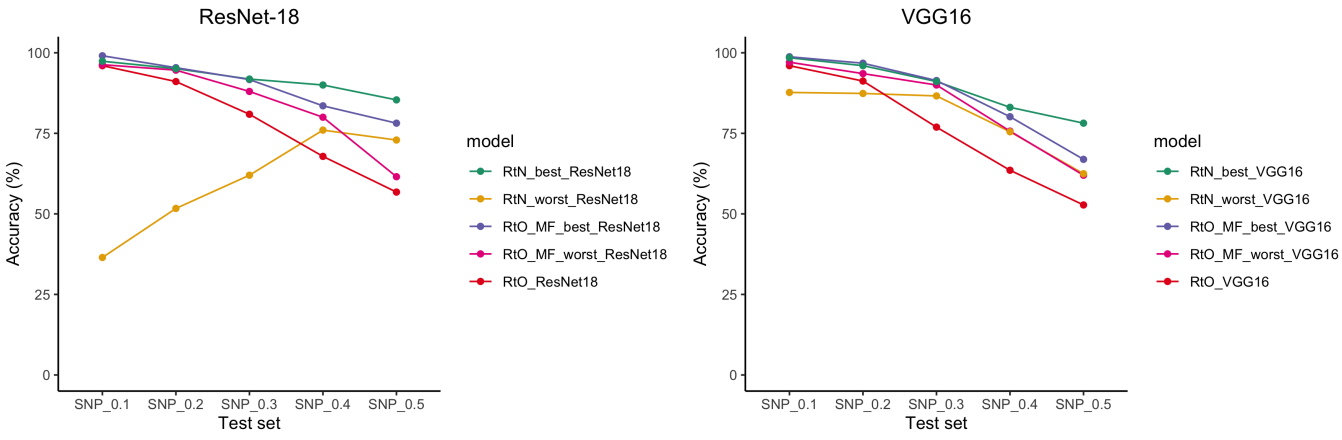

Figure 6: caption

After this paper, I will find a strong rival in the speckle noise removal area and compare it with our retrained models. In a big picture, I would like to prove that retrained models have advantage in some scenarios compared with traditional preprocessing methods(noise removal, enhancement etc).

5 Resource and Rule

Github Repo: Project Repo
Report Frequency: Every two weeks
Next Report: June 17, 2022