https://doi.org/10.2352/EI.2024.36.10.IPAS-249

Coverage Optimization for Training Data Enhancement of Ill-Performing Classes

R. Park and C. Lee
Yonsei University, Seoul, Korea

Abstract

Although CNN-based classifiers have been successfully applied to object recognition, their performance is not consistent. In particular, when a CNN-based classifier is applied to a new dataset, the performance can substantially deteriorate. Furthermore, classification accuracy for certain classes can be very low. In many cases, the poor performance of ill-performing classes is due to biased training samples, which fail to represent the general coverage of the ill-performing classes. In this paper, we explore how to enhance the training samples of such ill-performing classes based on coverage optimization measures. Experimental results show some promising results.

1. Introduction

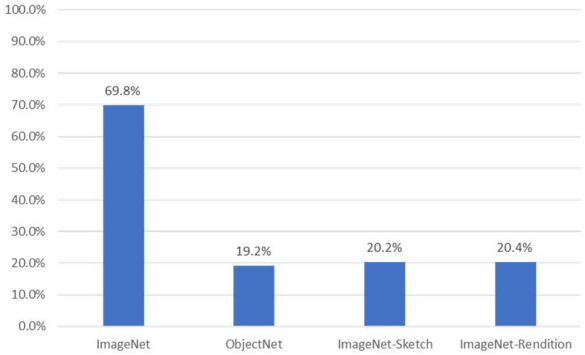
CNN-based classifiers have been successfully applied to objection recognition, outperforming conventional classification methods [1-9]. However, there are noticeable performance differences between training and validation data (Fig. 1) and the performance can deteriorate when applied to a new dataset (e.g., ObjectNet).

When a CNN-based classifier trained using the IMAGENET dataset is applied to the ObjectNet dataset, the classification accuracy can be very low for some classes (Fig. 2). When the CNN-based classifier was applied to other datasets (ImageNet-Sketch, ImageNet-Rendition), the performance also deteriorated noticeably as shown in Fig. 2.

If class-wise performance is compared, the performance differences raise more concerns. Fig. 3 shows class-wise performance differences between ImageNet and the other three datasets (ObjectNet, ImageNet-Sketch, ImageNet-Rendition). For some classes, the differences are very large. For example, for the pitcher class (ObjectNet), the classification accuracy is zero. The reason for this poor performance is that the pitcher types of the ImageNet dataset are different from those of the ObjectNet dataset (Fig. 4). Most of the ImageNet pitcher classes are glasses or porcelain, whereas most of the ObjectNet pitcher classes are plastic. Due to this kind of biased training data, the performance of CNN-based classifiers can deteriorate for new test samples.

It would be very difficult to collect training samples that cover all types of certain classes. If performance deterioration occurs due to biased training samples, the training samples of such ill-performing classes need to be enhanced to obtain more consistent performance. Adding more training samples to ill-performing classes may not work adequately. Also, increasing the number of training samples of certain classes may produce undesirable effects since the classes may be over-represented during the training procedure.

In this paper, we investigate how to enhance the training samples of such ill-performing classes based on coverage optimization measures.



- Fig. 2. Performance on new datasets (ResNet18).

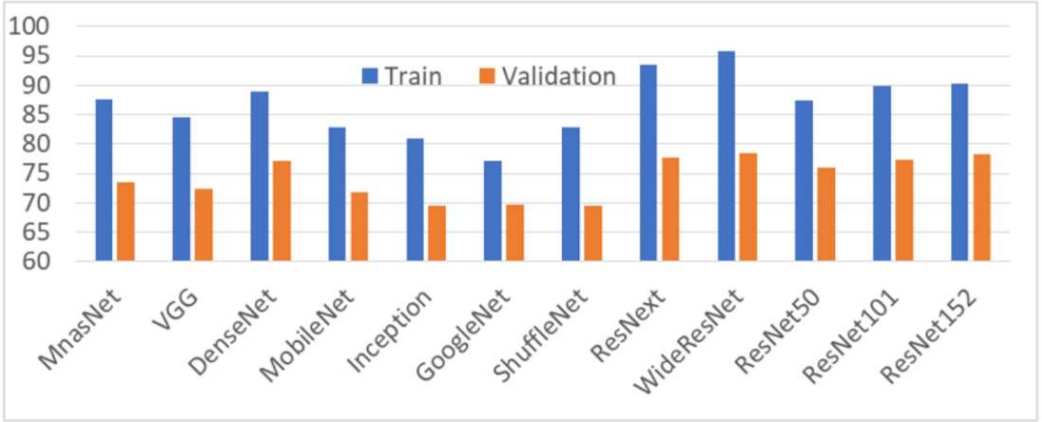


Fig. 1. Performance differences between training and validation data.

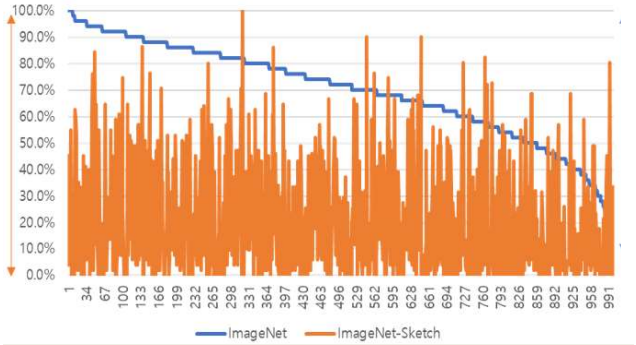


Fig. 3. Class-wise performance differences.

2. Method

2.1 Feature Space

Researchers have proposed various algorithms to add training samples systemically, which include uncertainty sampling [10], query-by-committee [11], ensemble approach [12], Bayesian approach [13], and learning loss [14].

In this paper, we defined an uncertainty metric to add new training samples without redundant samples. For high-dimensional data with a large number of classes, it is difficult to measure the usefulness of new training samples for performance enhancement. To solve this problem, we first find an effective feature space where the usefulness of new training samples can be effectively measured. Then, for ill-performing classes, we first examined their distributions in the feature space.

For example, a training sample of the ImageNet dataset is an image (224x224x3). We want to add a new image if the new image is substantially different from the existing training images. Defining such a metric in the input space (224x224x3) is rather difficult. The number of nodes of the final layer of most CNN-based classifiers is the same as the number of classes. We can use the values of these nodes as a feature vector. However, since the number of classes of the ImageNet dataset is 1000, it is difficult to define a metric that measures the usefulness of new training samples.

In this paper, we selected the top three nodes of the final layer and used the three values to define a feature space. Although the top three nodes may be different for each input image, we found that the variations are very small. For example, Table 1 shows the top three node percentiles for the 'mitten' class. For 88.8%, the top three nodes are identical. We used these dominant top three nodes as a feature space.

Table 1. The top three node percentiles for the 'mitten' class.

ImageNet index	**658**	**806**	**474**	911	496
Output count	1099	35	21	19	14
Proportion	84.5%	2.7%	1.6%	1.5%	1.1%

Fig. 5 shows the distributions of the existing training samples of the 'sock' classes and candidate training samples in the feature space. It can be seen that most of the candidate training samples show similar characteristics and may not be helpful in enhancing the performance of ill-performing classes.

Fig. 4. (a) the pitchers of the ImageNet dataset are glass, metal or porcelain, (b) the ObjectNet dataset has many plastic pitchers.

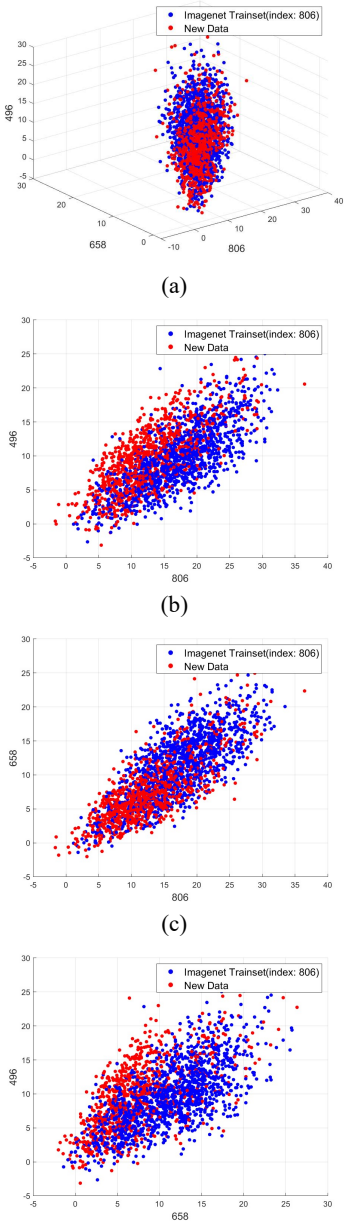


Fig. 5. Training sample distributions in the feature space (Sock). (a) 3D distribution, (b-d) 2D distributions.

2.2 Uncertainty Metric for Usefulness

We want to add a new training image that is substantially different from the existing training images. We defined an uncertainty metric as follows (Fig. 6):

$$m_{uncertainty} = min_{t \in trainset}(\|new\ data - t\|)$$

We used the Euclidean distance in the dominant 3-dimensional space. A new image is added to the train set if

$$m_{uncertainty} > t_{uncertainty}.$$

For a small value of $t_{uncertainty}$, more samples will be selected (Table 2).

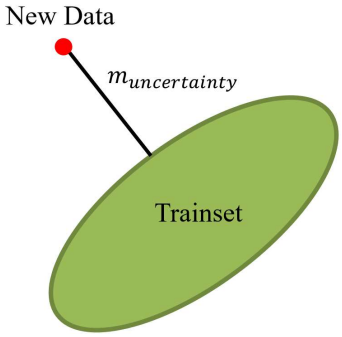


Fig. 6. Measuring the uncertainty metric.

Table 2. Number of selected images for different threshold values.

Index	t_uncertainty			
	0.25	0.3	0.35	0.4
487	218	209	200	187
588	745	718	678	644
658	676	665	652	629
737	332	318	284	256
806	854	835	819	800
879	295	280	260	253
930	773	752	718	671
Average	556	540	516	491

3. Experimental results

We selected seven ill-performing classes for the experiments. Table 3 shows the seven classes and the number of new images, which were manually collected using a search engine.

Table 3. Selected seven classes.

ImageNet index	class name	number of new images
487	cellular telephone	227
588	hamper	783
658	mitten	691
737	pop bottle	359
806	sock	873
879	umbrella	317
930	french loaf	813

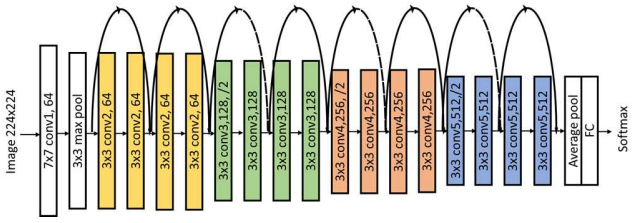


Fig. 7. ResNet architecture.

We used the ResNet architecture (Fig. 7) for the experiments. We chose it since we needed to train the network many times. We trained the network using the original ImageNet dataset and the selected new images with different threshold values (t_uncertainty: 0.25, 0.3, 0.35, 0.4). We used the pretrained weights as initial

weights. The batch size was set to 650 and the max epoch was set to 90. Fig. 8 shows the selected new training images for the 'Mitten' class.

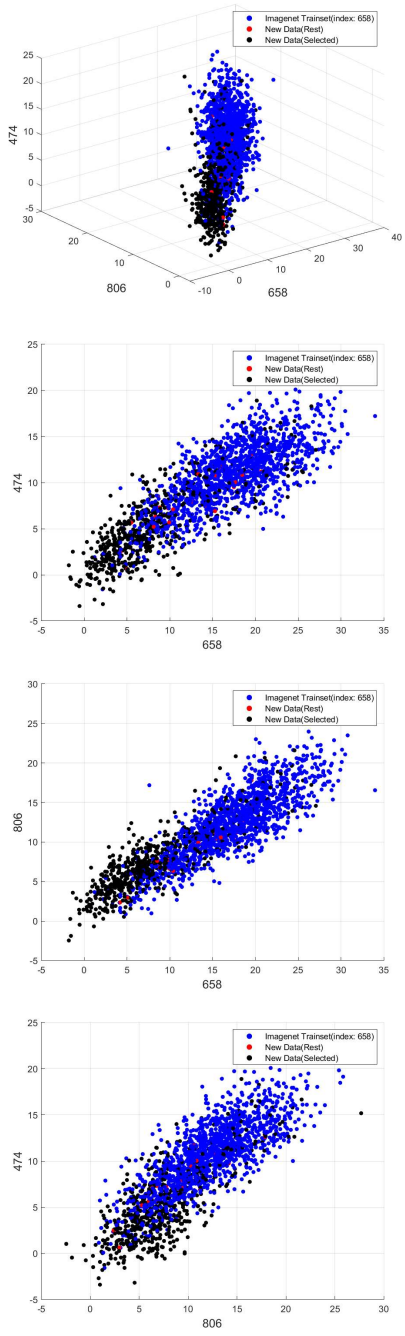


Fig. 8. Selected training images for $t\_uncertainty = 0.3$ (Mitten).

Tables 4-7 show the performance improvement for the four datasets (ImageNet, ObjectNet, ImageNet-Sketch, ImageNet-Rendition) when the training samples of the seven ill-performing classes were enhanced using the proposed algorithm. We also used all the new training images for comparison ('passive'). For some classes, the improvement is noticeable. For example, for the 'French loaf' class of ObjectNet, the classification accuracy was changed from 1.8% to 15.4%. For the 'umbrella' class of ImageNet-Sketch, the classification accuracy was changed from 32% to 70%.

4. Conclusions

In this paper, we explored how to improve the coverage of training data by adding new training samples for ill-performance classes without introducing redundant training samples. To address this problem, we first find an effective feature space where the usefulness of new training samples can be measured. Experimental results show performance improvement for some ill-performance classes.

References

[1] Karen Simonyan, Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," https://doi.org/10.48550/arXiv.1409.1556, 2014.

[2] He, K., Zhang, X., Ren, S., & Sun, J., "Deep Residual Learning for Image Recognition," Proc. the IEEE conference on CVPR, 2016.

[3] S. Zagoruyko and N. Komodakis, "Wide residual networks," in Proc. Brit. Mach. Vis. Conf., pp. 87.1–87.12, 2016.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. Learn. Representations, 2015.

[5] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 2261–2269, 2017

[6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Comput. Vision Pattern Recognit., pp. 4510–4520, 2018

[7] S. Xie et al., "Aggregated residual transformations for deep neural networks," in Proc. CVPR, pp. 5987–5995, 2017.

[8] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in Proc. Eur. Conf. Comput. Vis., Sep., pp. 122–138, 2018.

[9] M. Tan et al., "MnasNet: Platform-aware neural architecture search for mobile," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 2820–2828, 2019.

[10] Settles, Burr. "Active learning literature survey." (2009).

[11] Dagan, Ido, and Sean P. Engelson. "Committee-based sampling for training probabilistic classifiers." Machine Learning Proceedings 1995. Morgan Kaufmann, 1995. 150-157.

[12] W. Beluch, et al. "The power of ensembles for active learning in image classification." Proc. the IEEE conf. CVPR. 2018.

[13] Gal, Yarin, Riashat Islam, and Zoubin Ghahramani. "Deep Bayesian active learning with image data." International Conference on Machine Learning. PMLR, 2017.

[14] Yoo, Donggeun, and In So Kweon. "Learning loss for active learning." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.

Table 4. Performance improvement for the ImageNet dataset

ImageNet	Pretrained	Passive	Proposed (t_uncertainity)			
			0.25	0.3	0.35	0.4
cellular telephone	72.0%	70.0%	68.0%	74.0%	68.0%	74.0%
hamper	66.0%	70.0%	64.0%	64.0%	68.0%	60.0%
mitten	64.0%	74.0%	70.0%	72.0%	76.0%	70.0%
pop bottle	58.0%	70.0%	72.0%	68.0%	74.0%	66.0%
sock	60.0%	60.0%	68.0%	62.0%	64.0%	60.0%
umbrella	50.0%	50.0%	52.0%	56.0%	54.0%	52.0%
french loaf	56.0%	58.0%	58.0%	58.0%	56.0%	58.0%
Average	60.9%	64.6%	64.6%	64.9%	65.7%	62.9%

Table 5. Performance improvement for the ObjectNet dataset

ObjectNet	Pretrained	Passive	Proposed (t_uncertainity)			
			0.25	0.3	0.35	0.4
cellular telephone	4.8%	11.3%	12.6%	12.1%	10.4%	10.8%
hamper	3.1%	9.8%	9.8%	10.4%	9.8%	11.0%
mitten	2.1%	11.1%	12.5%	11.1%	10.4%	14.6%
pop bottle	4.3%	17.7%	16.3%	17.7%	17.7%	19.1%
sock	2.2%	9.3%	11.0%	11.0%	11.0%	14.8%
umbrella	3.1%	6.3%	5.6%	7.5%	3.8%	6.3%
french loaf	1.8%	11.8%	14.5%	14.9%	15.4%	14.9%
Average	3.0%	11.0%	11.8%	12.1%	11.2%	13.1%

Table 6. Performance improvement for the ImageNet-Sketch dataset

ImageNet-Sketch	Pretrained	Passive	Proposed (t_uncertainity)			
			0.25	0.3	0.35	0.4
cellular telephone	11.80%	27.50%	23.5%	35.3%	21.6%	37.3%
hamper	11.80%	17.60%	17.6%	27.5%	23.5%	21.6%
mitten	3.90%	5.90%	5.9%	9.8%	11.8%	5.9%
pop bottle	5.90%	11.80%	17.6%	9.8%	9.8%	9.8%
sock	0.00%	34.00%	32.0%	38.0%	36.0%	40.0%
umbrella	32.00%	68.00%	64.0%	70.0%	62.0%	62.0%
french loaf	0.00%	2.00%	2.0%	3.9%	2.0%	3.9%
Average	9.30%	23.80%	23.2%	27.8%	23.8%	25.8%

Table 7. Performance improvement for the ImageNet-Rendition dataset

ImageNet-Rendition	Pretrained	Passive	Proposed (t_uncertainity)			
			0.25	0.3	0.35	0.4
cellular telephone	9.70%	12.90%	11.30%	17.70%	12.90%	19.40%
hamper	N/A	N/A	N/A	N/A	N/A	N/A
mitten	0.00%	5.20%	5.20%	5.20%	6.50%	2.60%
pop bottle	N/A	N/A	N/A	N/A	N/A	N/A
sock	N/A	N/A	N/A	N/A	N/A	N/A
umbrella	N/A	N/A	N/A	N/A	N/A	N/A
french loaf	N/A	N/A	N/A	N/A	N/A	N/A
Average	4.80%	9.00%	8.20%	11.50%	9.70%	11.00%