Informative Dropout: Learning More Robust Features for Few-shot Classification

Baifeng Shi [*] [1] **Dejia Xu** [*] [1]

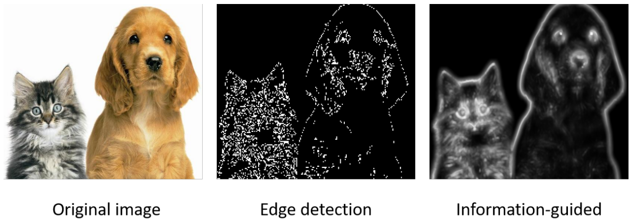
Abstract

Standard CNNs learn more texture-biased features, which limits their ability on transferring to novel classes or domains with different texture distribution, and thus degrades few-shot classification performance. This paper introduces Informative Dropout, which improves the robustness of learned features by paying more attention to informative image regions (*e.g.* shape information), and repressing activations at regions with less information (*e.g.* mundane repeated textures). Experimental results show that Informative Dropout can be integrated to improve the performance of many few-shot classification methods under various settings.

1. Introduction

With the rapid development of Deep Learning, Convolutional Neural Networks (CNNs) have achieved great success in various computer vision tasks. However, there remains one crucial issue about what the CNNs learn after the long training period. Recently, a number of works (Geirhos et al., 2018; Brendel & Bethge, 2019) demonstrate that CNNs learn more about local texture-biased features rather than shape-biased features which align better to human vision (Landau et al., 1988). Consequently, the bias towards texture limits the performance of CNNs from generalizing to images from different classes or domains, which have distinct texture style, and also degrades the performance for few-shot classification (Ringer et al., 2019).

Previous works (Geirhos et al., 2018; Zhang & Zhu, 2019) show that doing augmentation to the dataset or training in an adversarial manner benefits the CNNs to learn more robust features. This leaves an intriguing question to be solved. Are there other methods to improve the robustness of CNNs?



Original image

Edge detection

Information-guided

Figure 1. Saliency map generated from raw image by Info-Dropout and comparison with other shape-biased methods, *e.g.*, edge detector. One can observe that Info-Dropout can highlight more informative regions including global shapes and other characteristical features, *e.g.*, eyes and stripes. As comparision, other shape-biased methods like edge detecting are not robust to local textures, and the global appearance can be severely disrupted (see the cat for an example).

As shown in (Kornblith et al., 2019), a number of widely used regularizers improve standard ImageNet performance, but do not produce better learned representations for transferring. In this work, we introduce a novel Informative Dropout(Info-Dropout), which pays more attention to informative image regions (*e.g.* shape information), and represses activations at regions with less information (*e.g.* mundane repeated textures). An example is illustrated in Fig. 1. In this way, Info-Dropout can effectively alleviate texture-bias of CNN, and largely improves the robustness of learned features, as compared against regular Dropout (Srivastava et al., 2014). Extensive experiments demonstrate that our approach can boost the performance of many few-shot classification networks (Snell et al., 2017; Sung et al., 2018; Vinyals et al., 2016; Finn et al., 2017) under various test settings.

2. Related Works

Few-shot Classification

Significant progress has been made in the area of few-shot classification, which aims to learn a classifier to recognize unseen classes during training with limited labeled examples. One major kind of approach is the distance metric learning based method. They learn an embedding function

[*]Equal contribution [1]School of EECS, Peking University, Beijing, China. Correspondence to: Baifeng Shi <bfshi@pku.edu.cn >, Dejia Xu <dejia@pku.edu.cn>.

Preliminary work of ICML submission.

as well as an associated metric space, and thus no fine-tuning is needed when given the support set. Specifically, MatchingNet (Vinyals et al., 2016) uses cosine similarity, RelationNet (Sung et al., 2018) uses CNN based relation model, and ProtoNet (Snell et al., 2017) uses Euclidean distance to class-mean representation.

Another kind of approach, also known as parametric methods, involve two stages. They first pretrain a given model and then fine-tune the model on the given support set. One parametric method is model-agnostic meta-learning (MAML) (Finn et al., 2017), which adopts second-order gradients to learn an ideal initialization before further fine-tuning.

Robustness for CNNs

CNNs are widely believed to recognise objects by learning increasingly complex spatial features. Until recently, (Geirhos et al., 2018) suggest that ImageNet-trained standard CNNs are more biased towards texture. However, as shown in (Landau et al., 1988), shape-biased representations are more important for human vision.

(Geirhos et al., 2018) show that CNNs trained using a combination of Stylized-ImageNet and ImageNet can help standard CNNs learn more shape-biased representations. (Zhang & Zhu, 2019) prove that adversarially trained networks are also capable of learning more shape-biased features. Experiments (Geirhos et al., 2018; Zhang & Zhu, 2019) demonstrate the robustness of shape-biased features over various distortions and transformations.

Although texture-bias is acceptable for standard classification tasks, it severely degrades the few-shot classification performance (Ringer et al., 2019). In this paper, we aim to learn more robust features for few-shot classification with the help of Info-Dropout. Similar approach such as Information Maximization is previously used in (Bruce & Tsotsos, 2006) for saliency tasks.

The following sections are organised as follow. In Sec. 3 we introduce the proposed Informative Dropout(Info-Dropout). Experimental results are presented in Sec. 4. And finally in Sec. 5, discussions and future works are given.

3. Proposed Method

3.1. Notations

Assume we have a dataset of image-label pairs $\{(\mathbf{x}, y)\}$, where $\mathbf{x} \in \mathbb{R}^d$ is an image with d pixels and labeled as y. For a convolutional neural network, we denote input of lth layer as $\mathbf{z}^l \in \mathbb{R}^{d_l}$ and output as $\mathbf{z}^{l+1} \in \mathbb{R}^{d_{l+1}}$. Note that $\mathbf{z}^0$ equals to the input image $\mathbf{x}$. Assume the lth layer has a convolutional kernel $\mathbf{k}^l \in \mathbb{R}^{k\times k}$ and bias b^l, where k is kernel size. Then for the jth element z_j^{l+1} in output $\mathbf{z}^{l+1}$, we have $z_j^{l+1} = f(\mathbf{k}^l * \mathbf{p}_j^l + b^l)$, where $\mathbf{p}_j^l$ is the jth patch in $\mathbf{z}^l$ with size $k \times k$, $*$ indicates inner product and f is activation function (*e.g.* ReLU).

3.2. Informative Dropout

With an input patch $\mathbf{p}_j^l$ and corresponding output neuron $z_j^{l+1} = f(\mathbf{k} * \mathbf{p}_j^l + b^l)$, the main intuition of Info-Dropout is to drop the neuron with higher probability when input patch is less informative. First, we need to calculate information of patch $\mathbf{p}_j^l$. Assume that $\mathbf{p}_j^l$ and other patches in its neighbourhood $\mathcal{N}_j^l$ come from the same distribution $\mathbf{p} \sim q_j^l(\mathbf{p})$. Here the neighbourhood means a local region centered at $\mathbf{p}_j^l$, with radius R, *i.e.*, the neighbourhood contains $(2R+1)^2$ patches.

In our method, we approximate $q_j^l(\cdot)$ by kernel density estimation, *i.e.*,

$$q_j^l(\mathbf{p}) = \frac{1}{(2R+1)^2} \sum_{\mathbf{p}' \in \mathcal{N}_j^l} K(\mathbf{p} - \mathbf{p}'), \tag{1}$$

where $K(\mathbf{p}-\mathbf{p}')$ is a kernel function. Here we use Gaussian kernel, *i.e.*, $K(\mathbf{p}-\mathbf{p}') = \frac{1}{\sqrt{2\pi}h} \exp(||\mathbf{p}-\mathbf{p}'||^2/2h^2)$, where h is the bandwidth. Then information of $\mathbf{p}_j^l$ is given by

$$I(\mathbf{p}_j^l) = -\log q_j^l(\mathbf{p}_j^l). \tag{2}$$

We can observe that, the more different $\mathbf{p}_j^l$ is from neighbouring patches, the more information it contains.

For patches with higher information $I(\mathbf{p}_j^l)$, corresponding neurons z_j^{l+1} should be dropped out with lower probability. Therefore, we can define drop probability of neuron in a single sampling as

$$r(z_j^{l+1}) = \frac{1}{Z} e^{-I(\mathbf{p}_j^l)/T}, \tag{3}$$

where T is temperature and Z is normalizing parameter. Apparently, when T goes to infinity, each neuron has equal drop probability, and the whole algorithm becomes regular Dropout. When T lowers down, the algorithm becomes more conservative, *i.e.*, only patches with least information (*e.g.* a patch in a solid-colored region) will be dropped.

During training, for each output $\mathbf{z}^{l+1} \in \mathbb{R}^{d_{l+1}}$, we follow the regular Dropout to sample $r_0 \cdot d_{l+1}$ elements with probability $r(z_j^{l+1})$ and set them to 0, where $r_0 \in [0,1]$ is the drop rate. Normally, this involves weighted sampling without replacement from $\mathbf{z}^{l+1}$. However, this process is serialized and has low efficiency. Therefore, we instead use sampling *with replacement*, which can be paralleled during implementation. As a consequence, drop rate r_0 can be any positive real number due to collision of samples, and the actual drop rate (ratio of sampled neurons) will be lower than r_0. In

Algorithm 1 Informative Dropout

```
Input: input activation map $\mathbf{z}^l$
Parameters: convolutional kernel $\mathbf{k}^l$, bias $b^l$, radius $R$, temperature $T$, bandwidth $h$, drop rate $r_0$, output dimension $d_{l+1}$
Output: output activation map $\mathbf{z}^{l+1}$

for each patch $\mathbf{p}_j^l$ in $\mathbf{z}^l$ do
  $\mathbf{z}_j^{l+1} \leftarrow \mathbf{k}^l * \mathbf{p}_j^l + b^l$
end for
if is training then
  for $i = 1$ to $r_0 \cdot d_{l+1}$ do
    sample $j$ from $[1, d_{l+1}]$ with probability $r(z_j^{l+1})$ given by Eq. 3
    $\mathbf{z}_j^{l+1} \leftarrow 0$
  end for
else
  for $j = 1$ to $d_{l+1}$ do
    $\mathbf{z}_j^{l+1} \leftarrow \mathbf{z}_j^{l+1} \cdot e^{-r(z_j^{l+1}) \cdot r_0 \cdot d_{l+1}}$
  end for
end if
```

the following discussion, we use the notation r_0 for both Dropout (sampling *without replacement*) and Info-Dropout (sampling with replacement).

During inference, each neuron should be multiplied by a factor, so that its value can meet its average value during dropping process. In regular Dropout, each neuron will be preserved with probability $1 - r_0$, thus each neuron is multiplied by $1 - r_0$ during inference. For Info-Dropout, each neuron z_j^{l+1} is preserved with probability

$$\begin{aligned}(1 - r(z_j^{l+1}))^{r_0 \cdot d_{l+1}} &= (1 - r(z_j^{l+1}))^{\frac{1}{r(z_j^{l+1})} \cdot r(z_j^{l+1}) \cdot r_0 \cdot d_{l+1}} \\ &\approx e^{-r(z_j^{l+1}) \cdot r_0 \cdot d_{l+1}},\end{aligned} \tag{4}$$

where we use the limit $\lim_{x \to 0} (1 - x)^{\frac{1}{x}} = e^{-1}$ observing that normally number of neurons is really large and drop probability of a single neuron is close to 0. Thus for Info-Dropout, each neuron is multiplied by a factor $e^{-r(z_j^{l+1}) \cdot r_0 \cdot d_{l+1}}$ during inference. The whole algorithm is demonstrated in Alg. 1.

4. Experiments

4.1. Experimental Setup

4.1.1. Datasets

We demonstrate effectiveness of our algorithm under three scenarios with three datasets, respectively:

- **Generic object recognition**. In this scenario, we use *mini*-ImageNet for evaluation, which is commonly used in few-shot learning community. The dataset contains a subset of 100 classes from the whole ImageNet dataset and contains 600 images for each class. Following settings in previous works, we randomly divide the whole 100 classes into 64 training classes, 16 validation classes and 20 novel classes.
- **Fine-grained object recognition**. For this task, we use CUB-200-2011 dataset for evaluation. This dataset contains 200 classes with 11788 images. We divide it into 100 base classes, 50 validation classes and 50 novel classes.
- **Cross-domain object recognition**. Both scenarios mentioned above use images from the same domain during training and testing, which induces less gap between base and novel classes. Therefore, we also test our model on novel classes from a *different* domain which is not seen during training. In specific, we train our model on CUB dataset and test it on *mini*-ImageNet (this setting is denoted as CUB→*mini*-ImageNet). We use 100 base and 50 validation classes from CUB during training, and test on 20 novel classes from *mini*-ImageNet.

4.1.2. Implementation Details

For each dataset, we test both 5-way 5-shot setting and 5-way 1-shot setting. We also conduct experiments on both raw and augmented data. For data augmentation, we choose three common methods, namely random crop, random horizontal flip and image jitter.

For each experiment, we train the model for 80000 episodes with batch size of 16. For each episode, we sample N classes to form N-way classification, and k support images and 16 query images from each class to form k-shot setting. All models are trained from scratch with Adam optimizer and learning rate of 1e-3 During testing, we sample 600 episodes and use the average accuracy as the final result.

Note that our algorithm is plug-and-play style and can be integrated into any few-shot algorithms using a neural network. Thus, we can prove the effectiveness of our method by incremental results on existing methods when integrated with Info-Dropout. In our experiments we test our method on three common methods, namely ProtoNet (Snell et al., 2017), MatchingNet (Vinyals et al., 2016) and RelationNet (Sung et al., 2018). We use a 4-layer convolution network as backbone for all methods.

4.2. Quantitative Results

Before evaluating Info-Dropout, we start with validating our re-implementation of the previous methods on *mini*-

Table 1. Comparisons of reported and our re-implemented results on 5-shot *mini*-ImageNet. For all previous methods our results is higher than reported ones, which verifies our re-implementation.

	mini-ImageNet (5-shot)	
Method	Reported	Ours
ProtoNet	-	66.85 +- 0.65
MatchingNet	55.31 +- 0.73	65.73 +- 0.69
RelationNet	65.32 +- 0.70	68.51 +- 0.64

Table 2. Comparisons of original ProtoNet and one integrated with Info-Dropout on *mini*-ImageNet. Info-Dropout can improve the result under different settings, regardless of number of shots and usage of augmentation.

Shot	Aug	ProtoNet	+Info-Dropout
5	w/o	63.84 +- 0.67	**66.98 +- 0.68**
5	w/	66.85 +- 0.65	**69.20 +- 0.64**
1	w/o	47.96 +- 0.79	**49.92 +- 0.77**
1	w/	46.93 +- 0.80	**47.87 +- 0.81**

ImageNet. Results are listed in Table 1. On all tested methods, our implementation is better than original ones. Note that in original implementation, random sized crop is used for data augmentation, which we find undermining for both baseline and our method, and replace with random crop, where the cropped size is fixed.

4.2.1. Generic Object Recognition

Now we evaluate our algorithm on three scenarios. For generic object recognition, we conduct our experiments on *mini*-ImageNet. First we evaluate Info-Dropout with ProtoNet as baseline in settings of 5(1)-shot and with(out) augmentation. Results are illustrated in Table 2. One can observe that under all settings, Info-Dropout can improve the results with a nontrivial margin. It's worth noticing that under 1-shot setting, data augmentation can downgrade the accuracy, which is possibly because data augmentation tends to make the model overfit to base classes.

We also test Info-Dropout on different baseline methods to further validate the generalizing capability of our method. We test Info-Dropout on different baselines in 5-shot classification of *mini*-ImageNet with augmentation. Results are demonstrated in Table 3. On all three tested algorithms Info-Dropout can improve the baseline results, which verifies soundness and generalization of our method.

4.2.2. Fine-grained Object Recognition

For fine-grained recognition, we compare different methods on CUB dataset. We test effect of Info-Dropout on ProtoNet baseline, under settings of different number of shots and augmentation. Results can be found in Table 4. We can observe that Info-Dropout can improve the result under different settings, regardless of number of shots and usage of augmentation. Especially, under 5-shot setting without augmentation, Info-Dropout can improve the result by 11 percent accuracy.

Table 3. Effect of Info-Dropout on different baselines. For all the three common baseline methods, Info-Dropout can improve the result.

Method	Baseline	+Info-Dropout
ProtoNet	66.85 +- 0.65	**69.20 +- 0.64**
MatchingNet	65.73 +- 0.69	**66.37 +- 0.70**
RelationNet	68.51 +- 0.64	**68.78 +- 0.66**

Table 4. Comparisons of original ProtoNet and one integrated with Info-Dropout on CUB dataset. Info-Dropout can improve the result under different settings, regardless of number of shots and usage of augmentation. Especially, under 5-shot setting without augmentation, Info-Dropout can improve the result by 11 percent accuracy.

Shot	Aug	ProtoNet	+Info-Dropout
5	w/o	61.54 +- 0.77	**72.96 +- 0.66**
5	w/	77.88 +- 0.62	**78.90 +- 0.63**
1	w/o	47.76 +- 0.86	**52.66 +- 0.93**
1	w/	57.34 +- 0.93	**59.14 +- 0.92**

4.2.3. Cross-domain Object Recognition

For cross-domain object recognition, we illustrate results on CUB→*mini*-Imagenet dataset. As other two scenarios, we test effect of Info-Dropout on ProtoNet method. Results are listed in Table 5. We can observe that Info-Dropout can improve the result under different settings, regardless of number of shots and usage of augmentation. Furthermore, improvements are larger under 1-shot setting, which indicates that Info-Dropout can learn more robust features.

4.3. Ablation Study

First we conduct ablation study to validate our motivation, viz., Info-Dropout can help network learn more robust features. Despite improvements illustrated in Section 4.2, there are two possible explanation for the results:

1. Info-Dropout does make network attend more to global information and learn more robust features.
2. The network manifest more robustness only because less robust features are multiplied by a smaller factor

Table 5. Comparisons of original ProtoNet and one integrated with Info-Dropout on CUB→*mini*-Imagenet dataset. Info-Dropout can improve the result under different settings, regardless of number of shots and usage of augmentation. We can observe that improvements are larger under 1-shot setting, which indicates that Info-Dropout can learn more robust features.

Shot	Aug	ProtoNet	+Info-Dropout
5	w/o	39.04 +- 0.62	**40.30 +- 0.61**
5	w/	44.01 +- 0.67	**45.82 +- 0.66**
1	w/o	28.05 +- 0.54	**29.52 +- 0.58**
1	w/	29.32 +- 0.57	**32.20 +- 0.62**

Table 6. Ablation Study on effect of Info-Dropout during training and inference, respectively. One can observe that Info-Dropout in test time only has no effect on model's performance, which indicates that simply masking out less robust features during inference barely makes any difference. In the other hand, there is already a magnificent improvement when Info-Dropout is used only in training. This indicates that Info-Dropout do helps in learning more robust features during training stage. Last but not least, Info-Dropout during inference does not make much difference, which means the model has already learn robust features and is not depending on the inference masking.

Training	Testing	Accuracy
w/o	w/o	61.54 +- 0.77
w/o	w/	60.28 +- 0.69
w/	w/o	69.78 +- 0.69
w/	w/	69.97 +- 0.70

(given by Eq. 4) during inference.

The first explanation is what we intend for in the beginning, but the second one is also possible. To this concern, we conduct ablation studies by disabling Info-Dropout during training and testing, respectively, to find out whether the new method make the model learn more robust features during training, or it just mask out less robust features during inference. Results are listed in Table 6. One can observe that Info-Dropout in test time only has no effect on model's performance, which indicates that simply masking out less robust features during inference barely makes any difference. In the other hand, there is already a magnificent improvement when Info-Dropout is used only in training. This indicates that Info-Dropout do helps in learning more robust features during training stage. Last but not least, Info-Dropout during inference does not make much difference, which means the model has already learn robust features and is not depending on the inference masking.

Table 7. Ablation study on temperature T under 5-shot *mini*-Imagenet recognition with data augmentation. When $T = 0.7$, the model gets the best result. If temperature is too low, Info-Dropout gets too conservative and only mask out the least informative positions, which can be suboptimal. In opposite, when temperature goes to infinity, the whole algorithm degrades to regular Dropout, which is not discriminitive of informative positions and thus is worse at learning more robust features.

T	0.1	0.5	0.7	1	5	inf
Acc	68.84	69.20	**69.29**	68.91	68.54	67.88

Table 8. Ablation study on drop rate r_0 under 5-shot *mini*-Imagenet recognition with data augmentation. Results stay similar under different drop rate, which indicates robustness of our model.

r_0	0.1	0.3	0.5	0.7
Acc	68.84	68.35	67.29	67.99

Now we explore how different hyperparameters can effect our algorithm. Specifically, we test various drop rate r_0 and temperature T in our experiments. Results on different temperature are shown in Table 7. We can observe that when $T = 0.7$, the model gets the best result. If temperature is too low, Info-Dropout gets too conservative and only mask out the least informative positions, which can be suboptimal. In opposite, when temperature goes to infinity, the whole algorithm degrades to regular Dropout, which is not discriminitive of informative positions and thus is worse at learning more robust features.

In the end we test Info-Dropout under different drop rate r_0. Results of 5-shot *mini*-Imagenet recognition with data augmentation are listed in Table 8. Results stay similar under different drop rate, which indicates robustness of our model.

5. Discussion and Future Works

In this work, we start from the relationship between CNN's texture-shape bias and its robustness to domain gap in few-shot settings. Then we design a novel, plug-and-play method, namely Info-Dropout. It forces the model depend more on informative regions (*e.g.* shape information) in an image, and less on mundane features (*e.g.* repeated textures), and thus improves its robustness against domain gap between different classes. We integrate Info-Dropout into different baseline methods, and evaluate on different few-shot settings and datasets. The new method demonstrate clear improvements on various methods, which verifies its effectiveness and generality.

However, there are still possible improvements on the current algorithm. For example, when approximating patch distribution in Eq. 1, we adopt a hard selection and only consider the contribution of patches in a fix-sized neighbourhood. For future exploration, we can softly consider all the patches in the image and give the closer ones with higher weights. Besides, we can also try different kernels rather than Gaussian kernel in density estimation. Experiments on other tasks, *e.g.*, domain generalization and adversarial defending, can further verify our method's robustness.

Acknowledgements

Thanks to the teachers who prepared these courses this semester. We learned a lot from the fantastic lectures.

References

Brendel, W. and Bethge, M. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*, 2019. 1

Bruce, N. and Tsotsos, J. Saliency based on information maximization. In *Advances in neural information processing systems*, 2006. 2

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017. 1, 2

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 1, 2

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2661–2671, 2019. 1

Landau, B., Smith, L. B., and Jones, S. S. The importance of shape in early lexical learning. *Cognitive development*, 3(3):299–321, 1988. 1, 2

Ringer, S., Williams, W., Ash, T., Francis, R., and MacLeod, D. Texture bias of cnns limits few-shot classification performance. *arXiv preprint arXiv:1910.08519*, 2019. 1, 2

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017. 1, 2, 3

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 1

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 3

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, 2016. 1, 2, 3

Zhang, T. and Zhu, Z. Interpreting adversarially trained convolutional neural networks. *arXiv preprint arXiv:1905.09797*, 2019. 1, 2