Learning with Domain Knowledge to Develop Justifiable Convolutional Networks

Rimmon Bhosale RIMMON281996@GMAIL.COM and **Mrinal Das** MRINAL@IITPKD.AC.IN
Indian Institute of Technology, Palakkad, Kerala, India

Editors: Emtiyaz Khan and Mehmet Gönen

1. Detailed Experiment Setup

Below we provide a detailed description of the model architectures and the datasets used in our experiments in order to help reproducibility of the results.

1.1. Experimental Setup

All our experiments were run on a GPU system with 16 GB RAM and a single GeForce RTX 2080 GPU. The codes are implemented in Python 3.7 with Tensorflow v2.2. We have used the 'matplotlib' library in Python to generate all the plots and used the 'polyfit' function in 'numpy' library to regress a curve in the plots wherever necessary. As a preprocessing step, we normalize the input images in the range [0,1].

1.2. Model Architecture Detail

In all our experiments, we have used custom CNN architectures, the specifications for which are provided in Table 1 of the paper. Below we provide the additional details that can help in properly reproducing the model architectures from the description provided in Table 1 of the paper. For each dataset, all the experiments were run by maintaining an identical experimental setup for all the models. Input images from all the models except the IDRiD dataset were resized to the shape 96x96 while those belonging to the IDRiD dataset were resized to the shape 250x175. The number of filters used in each convolutional layers of these architectures is given in Table 1 of the paper. Every convolutional layer is followed by a batch normalization layer and later by a dropout layer (except for the Brain MRI dataset). We use a dropout(DO) value of 0.3. The last dropout layer is followed by a convolutional layer with 1×1 filter size and 16 filters in the case of CFCN-F models and 'C' filters in the case of CFCN-C, where 'C' is the number of classes in the given dataset. As IDRiD is a multi-label classification dataset, the 1x1 convolutional layer in this case has 3 filters, for both CFCN-F and CFCN-C. The 1×1 convolutional layer is then followed by a flatten layer in the case of CFCN-F, while for the CFCN-C models, we use global average pooling. In the case of CFCN-F the flatten layer is followed by the output layer which is a fully connected layer with 'C' neurons in it. In CFCN-C, global average pooling layer acts as the output layer. We have used 'ReLU' activation in all the intermediate layers. For output layer, we use 'softmax' activation in the case of multi-class classification and

'sigmoid' in the case of multi-label classification (i.e for IDRiD dataset). Details about the batch size (BS), learning rate (LR), number of epochs and the value for trade-off weight parameter α introduced in Eq. 1 and 2, for each dataset are given in Table 1 of the paper. Additionally, in order to calculate the justification loss as proposed in Eq. 1 and Eq. 2, we expect the activation masks to be of the same size as that of the feature map outputs from the last convolutional layer of the CNN. Accordingly, if the feature maps are of smaller size as compared to the input images, we can either down-scale the input activation masks to their size or up-scale the feature map outputs to the size of the input activation masks before computing the justification loss. That said, our approach can be easily adopted with modern CNN architectures like ResNet, VGG-16, VGG-19, etc. with slight modification only in the loss computation method.

1.3. Dataset Details

Table 1: Brief summary of the datasets used in the experiments.

Dataset	Classes	Size	Input Size	Significance
Oxford IIIT Pets	2	7349	96x96	Large annotated dataset.
Aeroplane-Cow	2	718	96x96	Small and biased dataset created from 'aeroplane' and 'cow' classes in Pascal VOC 2012 dataset.
Brain MRI	3	3064	96x96	Small medical dataset.
IDRiD	3	82	250×175	Very small medical dataset.

We have particularly selected the following four datasets - *Oxford IIIT Pets* , *Aeroplane-Cow*, *Brain MRI* and *IDRiD* - for performance comparison with the baseline. Below is a brief description of these datasets. Summary of these details is provided in Table 1.

Oxford IIIT Pets

We use this dataset for animal species classification into two classes: cat (2371 samples) and dog (4978 samples). It is an imbalanced dataset with 7349 images in total. We use the segmentation annotations available in this dataset, by converting them into binary activation masks.

Aeroplane-Cow

This is a biased dataset that we generated from the images belonging to the aeroplane and cow classes in the Pascal VOC 2012 dataset. We created the training set with 225 aeroplane images containing sky in the background and 143 cow images containing grassland in the background. The test set consists of the remaining 220 aeroplane and 130 cow images. We have manually generated the activation masks for all these images.

Brain MRI The Brain MRI dataset contains 3064 images belonging to three tumor classes: meningioma (708 samples), glioma (1426 samples), and pituitary tumor (930 samples). The ground-truth tumor locations for all the images are available which we have used as the input activation masks. The dataset comes with five subsets for cross validation, of which we have used the first four folds as our training data and the fifth fold as the testing data.

IDRiD

We use the segmentation data available in the publicly available Indian Diabetic Retinopathy image Dataset as a second example of a medical dataset. We use the following three disease classes available in this dataset: haemorrhages, hard exudates and soft exudates. This is a multi-label dataset with just 55 images in the training set and about 27 images in the testing set, which is an additional challenge brought up by this dataset.

2. Additional Qualitative Results

Below we present a few additional qualitative results for various experiments discussed in Section 4 of the paper.

Table 2: **(Paper Section 4.3): Effect of varying the trade-off parameter(α) in jCNN loss:** Below are the results reporting accuracy values for different models trained using different value of trade-off parameter α in the justification loss. One observation that we can make from it is that, we obtain similar performance in terms of classification accuracy for different values of α, but the qualitative results from Figure 5 of the paper help us in highlight the importance of our method.

α	0	0.2	0.5	1.0	2.0
jCNN-C	93%	88%	93%	91%	91%

Table 3: **(Paper Section 4.4): Experimenting with Different Mask Types:** Below are the quantitative results comparing accuracy values for different models trained using different mask types on the penguin dataset that we manually collected. In comparison to the below results, if we observe the qualitative results from Figure 6 of the paper, we can effectively assess the performance of the proposed method in learning more meaningful justifiable features.

Mask Type	Coarse	Fine-Grain	Important-Region	Non-Important Region
jCNN-C	80%	85%	85%	85%

Figure 1: **(Paper Section 4.6): Learn more with less data:** For better visualization purpose, we present the quantitative results for this experiment in the form of two plots rather than a matrix of metric values for different experimental settings. We can observe similar trends as mentioned in the observations for this experiment in the paper using the combined loss plot. The motivation for using the combined loss plot to present the results for this experiment in the paper was that, it captures both the performance for classification and justifiable feature learning simultaneously in a single plot. Also, please note that for the below plots, one unit on the x-axis represents 1000 training samples.

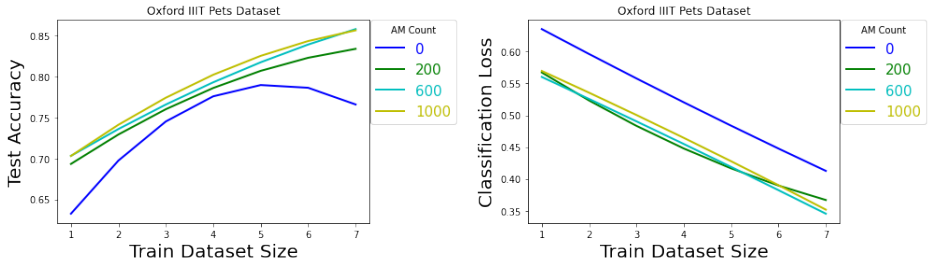

Table 4: **(Section 4.7 (a)): Varying background for the images:** Below table shows the quantitative results for the experiment in which we vary the background of the test images in the Oxford IIIT Pets dataset. We created 5 copies of the same test set by changing the background image in each set. We present the average accuracies for these five sets. If we compare these values with those presented in Table 2 of the paper, we observe that there is little to no variation in jCNN performance. But at the same time the performance for CNN and GAIN varies a bit. This highlights the influence of background features in the class predictions from these models.

Model	CNN	GAIN	jCNN-F	jCNN-C
Accuracy	88%	84%	87%	90%

Table 5: **(Section 4.7 (b)): Varying brightness of the images:** Below are the results for experiment in which we vary the brightness values for the input images. Even though we see performance variation in these values and those presented in Table 2 of the paper, we observe from Figure 11 of the paper that the jCNN has only slight variations in terms of features learnt. The results also suggests that we still have a scope of improvement in order to keep the quantitative performance stable in this case and we would like to work on it in our future work.

Model	CNN	GAIN	jCNN-F	jCNN-C
Accuracy	90%	84%	83%	87%