Supplementary Material M-CAM: Visual Explanation of Challenging Conditioned Dataset with Bias-reducing Memory

Seongyeop Kim
seongyeop@kaist.ac.kr
Yong Man Ro
ymro@kaist.ac.kr

Korea Advanced Institute of Science and Technology (KAIST)
Daejeon, Korea

1 Settings

1.1 Dataset

As described in the paper, we use five different datasets. We conduct multi-label classification task for NIH Chest X-ray 14 (NIH) [12] , VinDR-CXR (Vin) [9], and MS COCO (COCO) [7], and single-label classification for Retinal optical coherence tomography (OCT) [5] and EndoTect Challenge dataset (Endo) [3]. The number of images included in training and test set of each dataset is described in Table. 1. In case of NIH, we use validation set of the distributed version as the test set of our experiment because of the absence of test set label in the distributed version of NIH dataset. Vin dataset only provides 15,000 training set images with label, hence we randomly sample 3,000 images to use it as the test set of our experiment. As the same manner, we randomly sample 1,080 images as the test set for EndoTect dataset.

Dataset	Training set	Test set
NIH	78,469	11,219
Vin	12,000	3,000
COCO	82,783	40,504
OCT	83,484	968
Endo	9,582	1,080

Table 1: The number of images included in the training and test set of each dataset.

1.2 Implementation of Deep Network

In this section, we describe the setting of the network and its implementation for the experiments. The backbone network used for the feature encoder F is DenseNet 121 [4] for all

datasets. We pre-train the feature encoder F until it converges to the best performance for each dataset, then we train the Bias-reducing memory while F being fixed. We set the number of slots of Bias-reducing memory as 5 multiple to the number of classes of each dataset (*e.g.* NIH includes 14 classes, hence the number of memory slots is 70). To pre-train the feature encoder F and train the Bias-reducing memory, we use Adam optimizer [6] with a learning rate of 0.0001 and a learning rate decay of 0.7 each epoch. The implementation of the work is done with Pytorch 1.3.1 [10].

The performance evaluation of the pre-trained network and the one with the Bias-reducing memory is described in Table. 2 and Table. 3. We measure area under the ROC curve (AUC) for the multi-label classification tasks, and classification accuracy for the single-label classification tasks.

Dataset	Pre-trained Network **without** Memory	Pre-trained network **with** Memory
NIH	0.8487	0.8511
Vin	0.9462	0.9574
COCO	0.9119	0.9201

Table 2: Performance measured with AUC for multi-label classification tasks. The corresponding datasets are NIH Chest X-ray 14 , VinDR-CXR, and MS COCO.

Dataset	Pre-trained Network **without** Memory	Pre-trained network **with** Memory
OCT	99.37	99.89
Endo	89.08	89.45

Table 3: Performance measured with classification accuracy (%) for single-label classification tasks. The corresponding datasets are OCT and EndoTect Challenge dataset.

2 Quantitative Results

2.1 Average Drop Percentage

In this section, we want to verify if the generated visual explanation map highlights critical regions well that influence the deep network decision. Average Drop Percentage [1] measures the decrease of the target class prediction score once only the highlighted region of the visual explanation of the original image is provided as input. Let Y_i^c be the prediction score of the model for class c on the i th original image, and let O_i^c be the prediction score of the model for class c on the i th image, where the 40% of the top highlighted region of the image is only provided to the network. The metric is as follows,

$$AverageDrop\% = \sum_{i=1}^{N} \frac{\max(0, Y_i^c - O_i^c)}{Y_i^c} \cdot 100. \tag{1}$$

2.2 Percentage Increase in Confidence

Percentage Increase in Confidence [1] measures the frequency of appearance where the prediction score for the target class c increases when the unnecessary region is removed from the original image. As a complementary metric to Average Drop Percentage, Percentage Increase in Confidence is as follows,

$$\%Increase = \sum_{i=1}^{N} \frac{\mathbb{1}_{Y_i^c < O_i^c}}{N} \cdot 100. \tag{2}$$

Y_i^c and O_i^c are as the same as in Eq. 1, and $\mathbb{1}_x$ is an indicator function that returns 1 if the condition x is met.

2.3 Infidelity

Infidelity metric [13] measures the expected mean squared error between the generated explanation perturbed by significant noise and the difference between the original prediction and the one outputted by the perturbed input. The intuition is, faithful explanation method should be negatively influenced by significant noise added to an input. If the generated explanation remains similar to the original one even if the input is completely altered, the explanation method is not trustworthy. The equation used for the experiment is as follows,

$$Infidelity = \mathbb{E}_{\mathbf{I}\sim\mu_\mathrm{I}}\left[\left(\mathbf{I}^T\Phi(\mathbf{f},\mathbf{x}) - (\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}-\mathbf{I}))\right)^2\right], \tag{3}$$

where $\mathbf{I}$ is the significant perturbation noise, which is Gaussian random vector for the experiment, Φ is the explanation method, $\mathbf{f}$ is the target model, and x is the input image.

2.4 Sensitivity

Sensitivity metric [13] measures the change of explanation when small noise is added to an input. The intuition is, the generated explanation should remain similar to the original one even if the input is perturbed by insignificant noise. If not, the explanation method is considered to be fragile. The equation used for the experiment is as follows,

$$Sensitivity = \max_{\|\mathbf{y}-\mathbf{x}\|\leq r} \|\Phi(\mathbf{f},\mathbf{y}) - \Phi(\mathbf{f},\mathbf{x})\| \tag{4}$$

where Φ is the explanation method, $\mathbf{f}$ is the target model, x is the input image, y is the input image perturbed by small noise, and r is a hyperparameter, input neighborhood radius, which is set to 0.02 in the experiment.

3 Discussion on Memory Slot and Address Vector

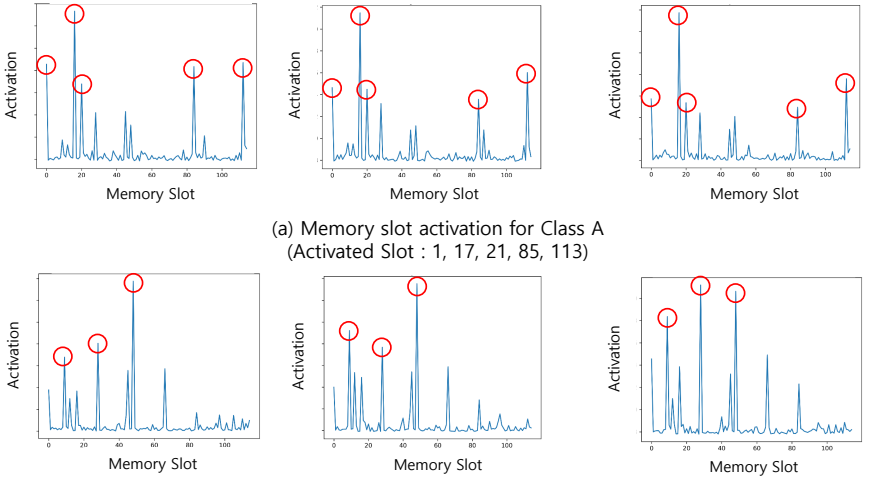
3.1 Number of Memory Slots

In repeated experiments varying the number of memory slots, we paid attention on two major factors. The first is, each memory slot is not to be shared by a single class. That is, we wanted different semantic (class) information to be stored in different memory slots being distinguishable from each other. The second is, for the sake of efficient optimization and

reduction of parameters, we wanted to minimize inactive memory slots. From the preliminary experiments, we have determined the number of memory slots N as $5C$, where C is the number of classes of a dataset (e.g. 70 memory slots for Chest X-ray 14 dataset with 14 classes). Regarding its influence on the performance, reduction of memory slots leads to faster optimization but it hinders the visualization quality due to sharing of slot by different classes. Increase of memory slots leads to slow and difficult optimization of the memory module due to inactive memory slots.

3.2 Memory Slot Activation Observed by Address Vector

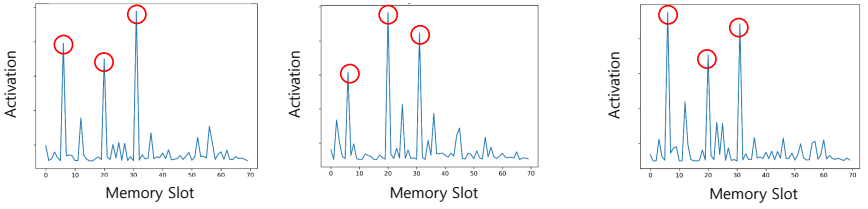
The address vector represents change on the memory slot activation depending on the input image. Figure.1, 2, and 3 are the samples of address vectors p. Fig.1 shows address vectors obtained from six different images of EndoTect dataset. The ground truth class is the same for the top three images, and so it is for the bottom three images with another class. Note that the red circles are the top-n activated slots for different images. We want to demonstrate that different images that are labeled with the same class show very similar patterns of activating memory. In addition, the location of the slots activated by different classes is not shared. We mostly observe 1 to 5 activation peaks for each class regardless of the dataset.



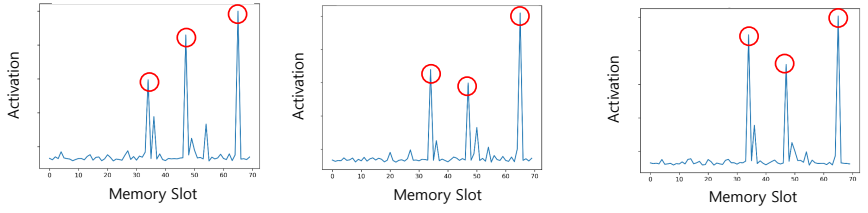
(a) Memory slot activation for Class A
(Activated Slot : 1, 17, 21, 85, 113)

(b) Memory slot activation for Class B
(Activated Slot : 10, 29, 49)

Figure 1: Address vectors obtained from six different images of EndoTect dataset. The top three examples are obtained from three different images that are labeled with the same class. The bottom three examples are obtained from the other three images being labeled with another class. The red circles denote the top-n activated slots for each sample.

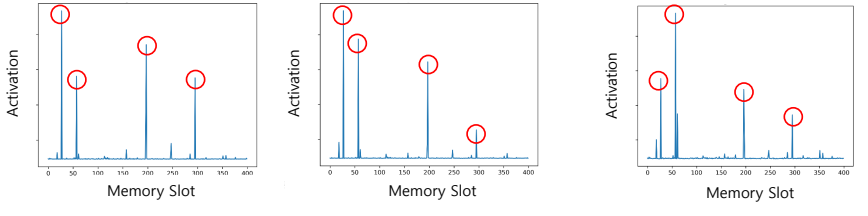


(a) Memory slot activation for Class A
(Activated Slot : 7, 21, 32)

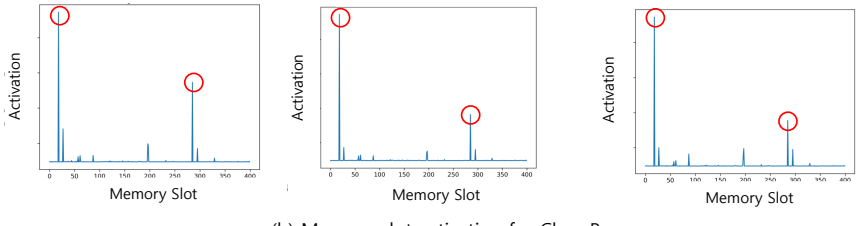


(b) Memory slot activation for Class B
(Activated Slot : 35, 48, 66)

Figure 2: Address vectors obtained from six different images of Vin chest X-ray dataset. The top three examples are obtained from three different images that are labeled with the same class. The bottom three examples are obtained from the other three images being labeled with another class. The red circles denote the top-n activated slots for each sample.



(a) Memory slot activation for Class A
(Activated Slot : 28, 58, 198, 296)

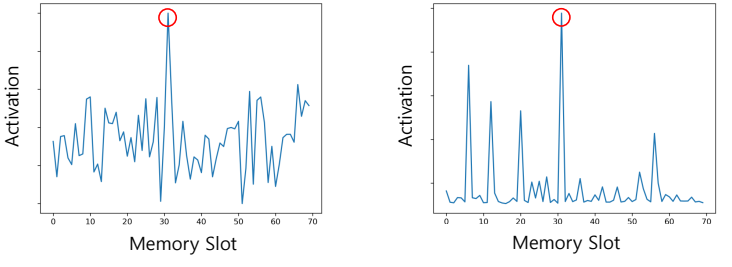


(b) Memory slot activation for Class B
(Activated Slot : 19, 286)

Figure 3: Address vectors obtained from six different images of MS COCO dataset. The top three examples are obtained from three different images that are labeled with the same class. The bottom three examples are obtained from the other three images being labeled with another class. The red circles denote the top-n activated slots for each sample.

3.3 Spatial Feature Representation Dictionary

We design $\mathcal{L}_{address}$ in order to store the same semantic information in the same location of Spatial Feature Representation Dictionary slot and Key Memory slot. A capacity of a single slot of Spatial Feature Representation Dictionary S is $(7 \times 7 \times 1024)$ and the one of Key Memory K is $(1 \times 1 \times 1024)$. Hence it is harder to train Spatial Feature Representation Dictionary than Key Memory, and p and p_s are not exactly identical. However, we included a sample figure Fig 4 to show a case where the most highly activated slots of the both modules being the same. Since we utilize a single slot for generating visual explanation, as long as Spatial Feature Representation Dictionary and Key Memory store meaningful information at the same location of memory slot, the visual explanation still can benefit from Spatial Feature Representation Dictionary reference. We also included Table 4 to show the loss of $\mathcal{L}_{address}$. The value in Table 4 is obtained by averaging the loss in the last training epoch of each model trained for each dataset. We have observed successful convergence of $\mathcal{L}_{address}$ during training, and we assume the noisy activation of irrelevant slots, like as (a) of Figure 4, is the cause of the difference between $KL(p' \,||\, p_s)$ and $KL(p' \,||\, p)$.



(a) p_s of Image A (b) p of Image A

Figure 4: Address vector of Spatial Feature Representation Dictionary (a) and the one of Key Memory (b) obtained from an image of Vin chest X-ray dataset. The location of the most highly activated slot is the same for both of the modules.

Dataset	$KL(p' \,\|\| p_s)$	$KL(p' \,\|\| p)$
NIH	5.426e-3	2.618e-3
Vin	6.199e-3	1.713e-3
COCO	6.338e-4	2.568e-4
OCT	3.081e-3	1.374e-3
Endo	1.542e-3	4.536e-4

Table 4: Loss value of $KL(p' \,||\, p_s)$ and $KL(p' \,||\, p)$ obtained from the last training epoch of the proposed model for each dataset.

4 Qualitative Results

We provide more qualitative results in the following four figures. We provide more results on MS COCO dataset on Figure 5, NIH & Vin dateset on Figure 6, OCT dataset on Figure 7, and EndoTect dataset on Figure 8.

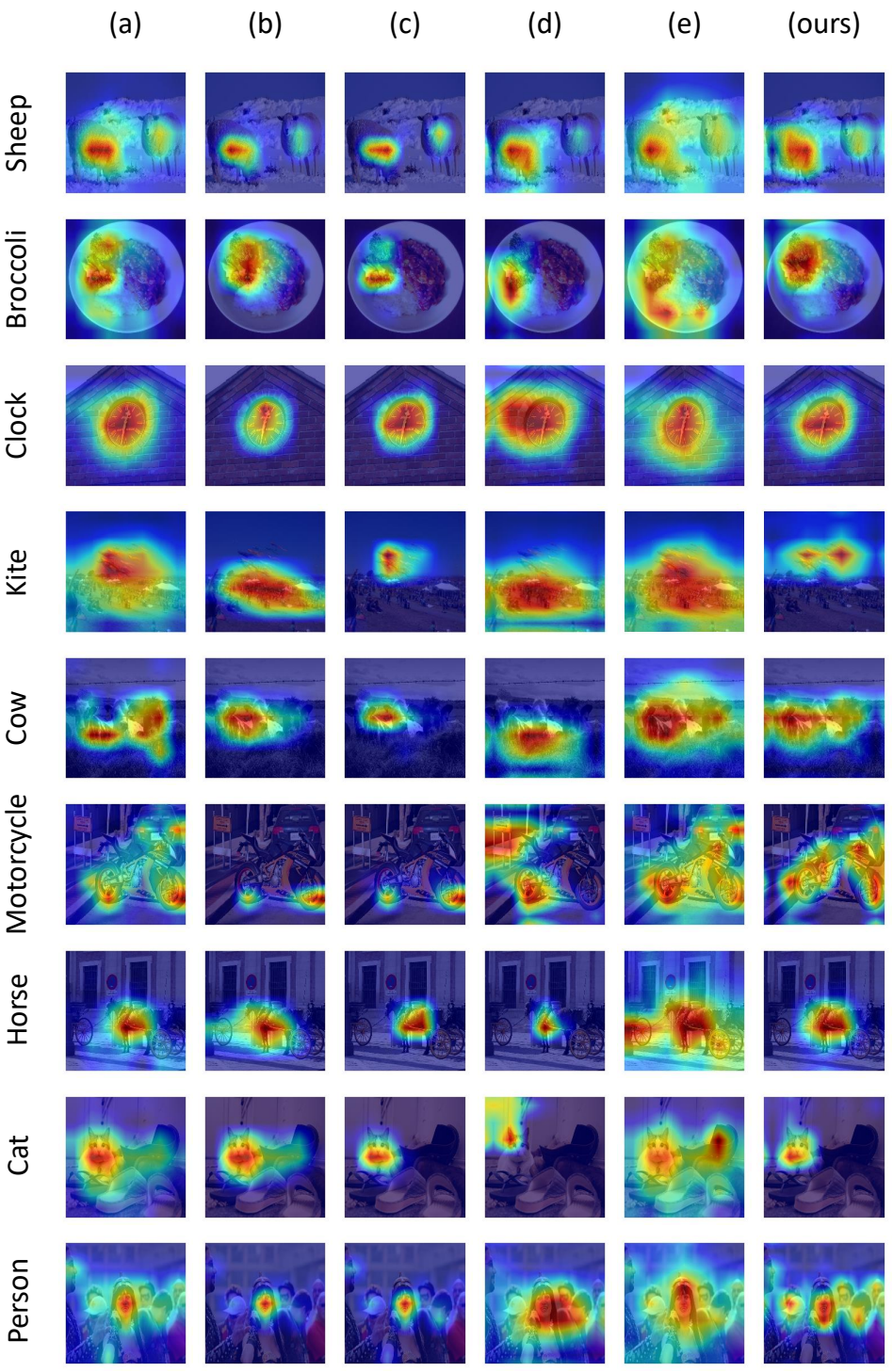


Figure 5: Qualitative results on MS COCO dataset. (a): Ablation CAM [2], (b): Eigen-CAM [3], (c): EigenGradCAM [3], (d): GradCAM [11], (e): GradCAM++ [1], (ours): M-CAM.

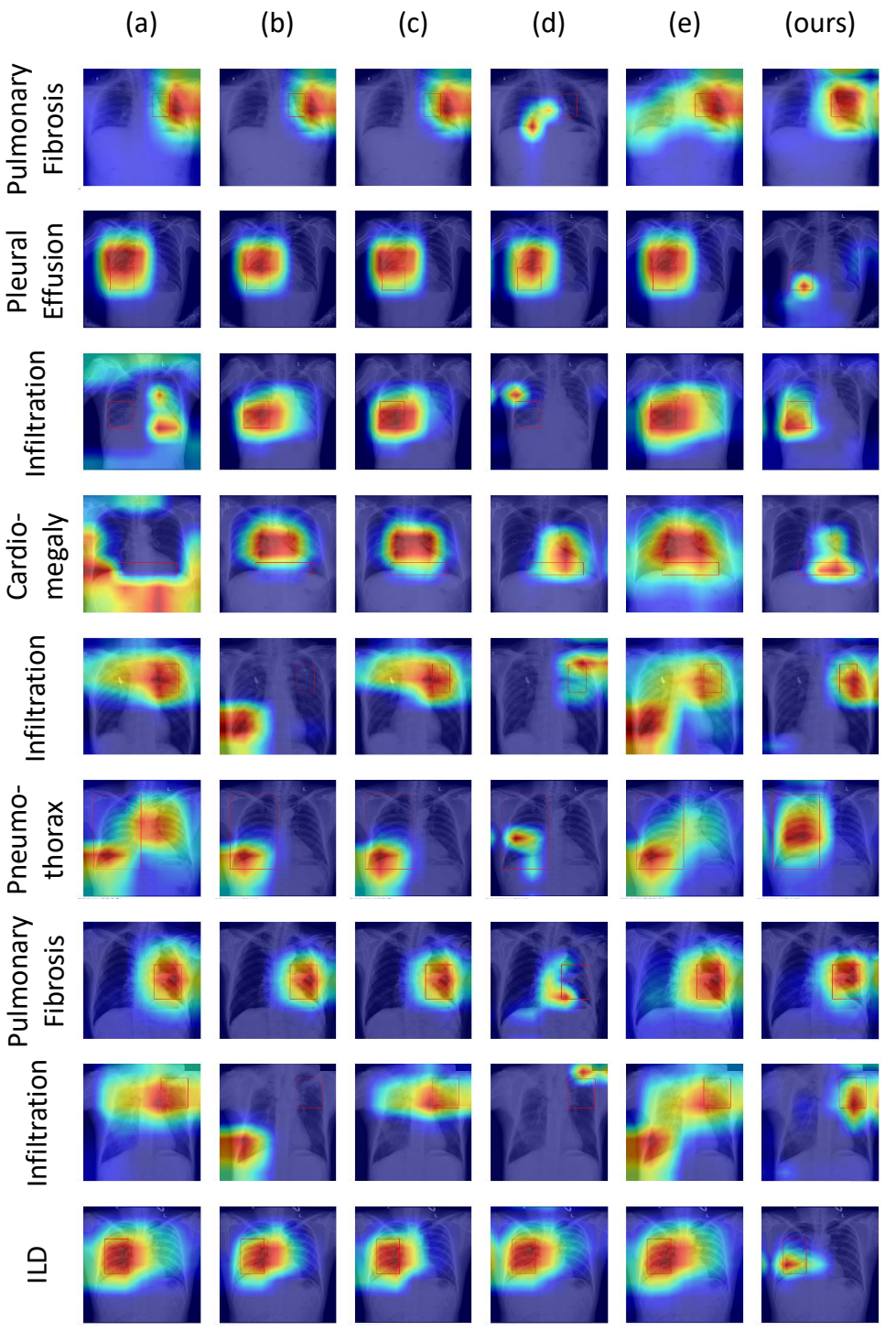


Figure 6: Qualitative results on NIH & Vin dataset. (a): Ablation CAM [2], (b): Eigen-CAM [3], (c): EigenGradCAM [3], (d): GradCAM [11], (e): GradCAM++ [1], (ours): M-CAM.

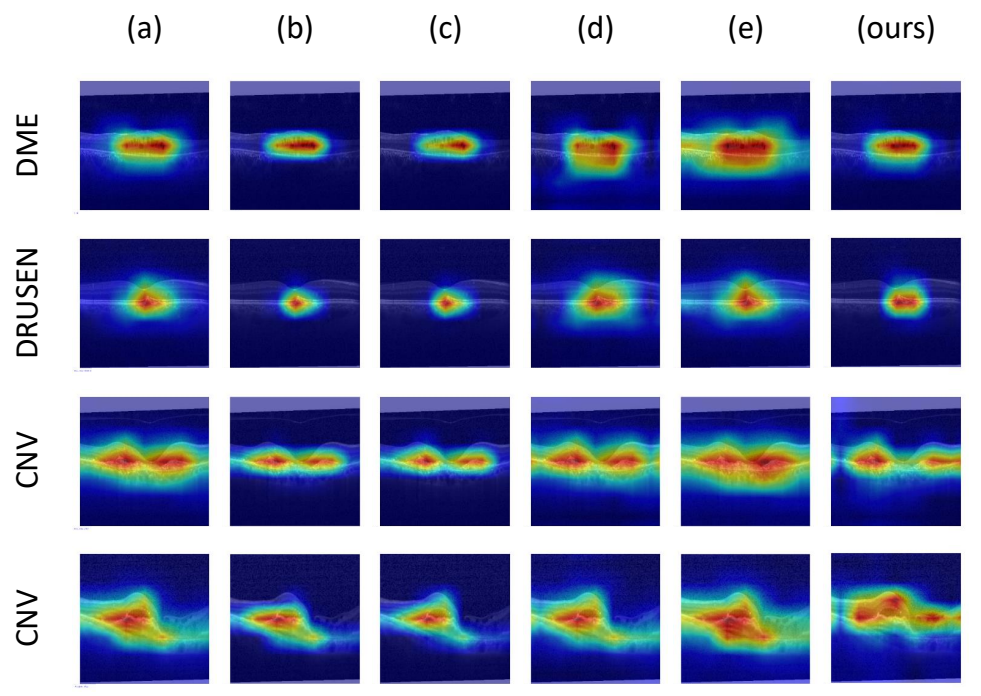


Figure 7: Qualitative results on OCT dataset. (a): Ablation CAM [7], (b): EigenCAM [8], (c): EigenGradCAM [8], (d): GradCAM [11], (e): GradCAM++ [1], (ours): M-CAM.

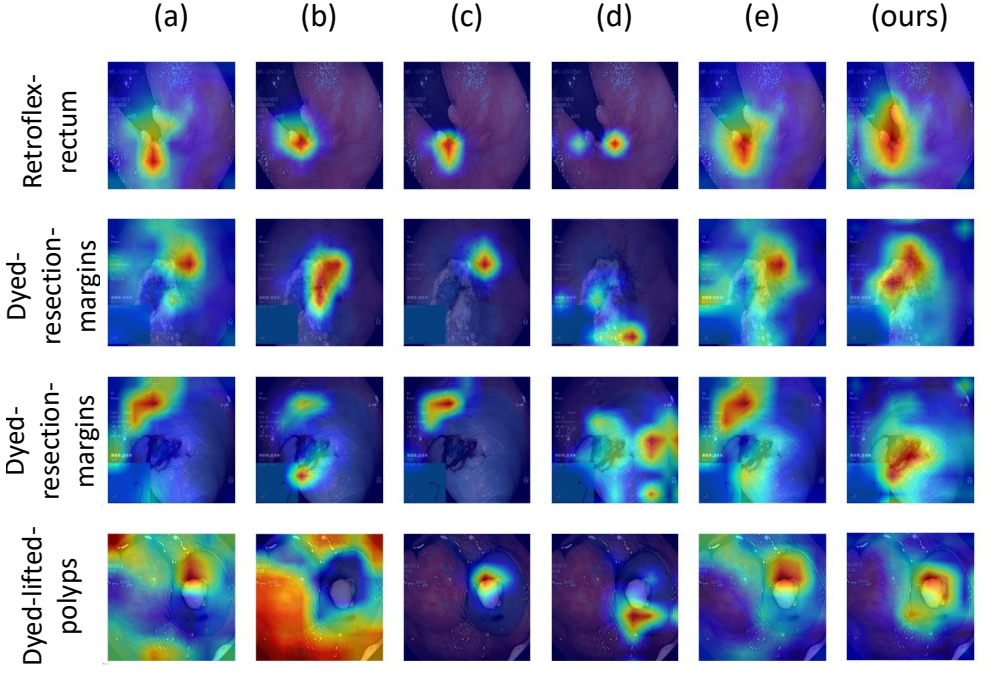


Figure 8: Qualitative results on EndoTect dataset. (a): Ablation CAM [7], (b): Eigen-CAM [8], (c): EigenGradCAM [8], (d): GradCAM [11], (e): GradCAM++ [1], (ours): M-CAM.

References

[1] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847. IEEE, 2018.

[2] Saurabh Desai and Harish G. Ramaswamy. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 972–980, 2020. doi: 10.1109/WACV45572.2020.9093360.

[3] Steven A. Hicks, Debesh Jha, Vajira Thambawita, Pål Halvorsen, Hugo L. Hammer, and Michael A. Riegler. The endotect 2020 challenge: Evaluation and comparison of classification, segmentation and inference time for endoscopy. In Alberto Del Bimbo, Rita Cucchiara, Stan Sclaroff, Giovanni Maria Farinella, Tao Mei, Marco Bertini, Hugo Jair Escalante, and Roberto Vezzani, editors, *ICPR International Workshops and Challenges*, pages 263–274, 2021. ISBN 978-3-030-68793-9.

[4] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[5] Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131, 2018.

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

[8] Mohammed Bany Muhammad and Mohammed Yeasin. Eigen-cam: Class activation map using principal components. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2020.

[9] Ha Q Nguyen, Khanh Lam, Linh T Le, Hieu H Pham, Dat Q Tran, Dung B Nguyen, Dung D Le, Chi M Pham, Hang TT Tong, Diep H Dinh, et al. Vindr-cxr: An open dataset of chest x-rays with radiologist's annotations. *arXiv preprint arXiv:2012.15029*, 2020.

[10] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[11] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[12] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2097–2106, 2017.

[13] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity and sensitivity of explanations. *Advances in Neural Information Processing Systems*, 32:10967–10978, 2019.