Classification of Biomedical Journal Images using Retargeting-Based Data Augmentation and Visually Explainable Attention Priors

Vinit Veerendraveer Singh
vinitvs@udel.edu
Chandra Kambhamettu
chandrak@udel.edu

VIMS Lab
Department of Computer and Information Science
University of Delaware
Delaware, USA

Abstract

Identifying regions of interest in images is vital for solving various computer vision tasks. Convolutional Neural Networks (CNNs) implicitly detect these regions. Per contra, CNN-compatible retargeting-based data augmentation approaches explicitly detect task-critical regions and enhance their spatial coverage. However, these retargeting approaches require modifying the original network architecture and have high space and time complexity. In addition, the task-critical regions learned by these methods can be inaccurate. Techniques that produce visual explanations for decisions from CNNs can faithfully identify task-critical regions, yet, they are primarily used for interpretability purposes. This paper proposes a data augmentation approach that utilizes outputs from visual explanation techniques as attention priors to retargeting-based data augmentations. We evaluated our approach to categorize biomedical journal images in three ImageCLEF datasets. The proposed approach outperformed state-of-the-art data augmentation approaches on these datasets. In addition, our approach has a significantly lower space complexity compared to other retargeting-based data augmentations approaches.

1 Introduction

Regularization approaches are commonly employed in Deep Neural Networks (DNNs) to minimize the empirical risk and to introduce productive biases. Data augmentation is an implicit approach to regularizing DNNs. Usually, data is *only* augmented during a neural network's training phase. Training time augmentations make DNNs robust to noise and variations in data at test time. They decrease the model variance by increasing the variance and noise in the training data. These augmentations are decided by a human user based on prior knowledge about the test distribution. For example, Convolutional Neural Networks cannot handle significant geometric variations in the data and are thus trained with rotated and flipped images. These augmentation strategies are not usually learned during training, and their magnitude is set in an ad-hoc manner.

On the other hand, retargeting-based data augmentation approaches [22, 26] identify and reduce the noise and variations in the data while training *and* testing a CNN. These augmentation approaches learn to disambiguate noise from a signal during the forward pass

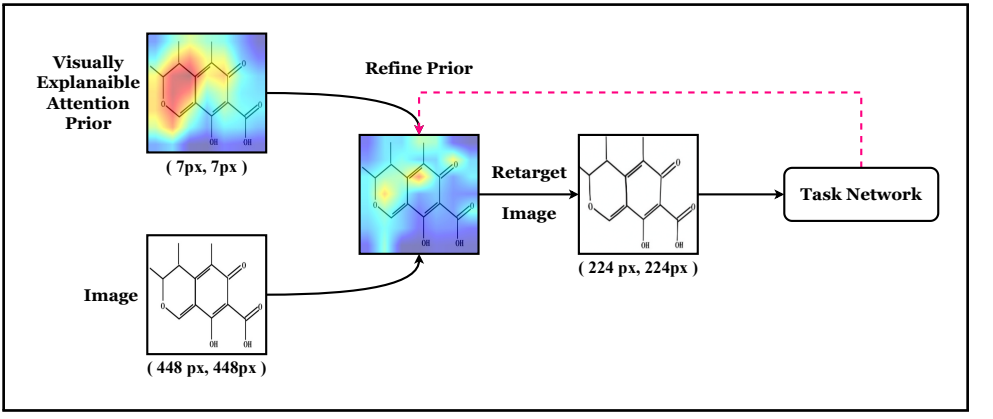


Figure 1: Our approach is to first obtain attention priors from techniques [2, 17, 23] that produce visual explanations for a Convolutional Neural Network's decisions. Then, we refine and propagate the attention to a CNN-compatible retargeting-based data augmentation approach [26] while retraining and testing the same Convolutional Neural Network. As shown, our data augmentation approach increased the spatial coverage of the task-critical regions in the image before sending it to the task (classification) network. The red color and blue color in the prior represent more and less critical regions, respectively. [Best viewed in color]

of the network by using an attention mechanism. They reduce the noise by preserving the task-critical regions prior to image sub-sampling. Consequently, they increase the spatial coverage of the regions essential to a task. However, unlike non-learnable data augmentations, learnable ones have some drawbacks. The obvious drawback is that they perform computationally intensive optimization at inference time and require additional learnable parameters. The additional computational cost leads to significant time delays during inference. Moreover, since they do not rely on prior knowledge from a human user about the test distribution, their learning needs to be more precise to generalize across different datasets. In this work, we address these drawbacks. The main contributions of this work are:

- We introduce a novel end-to-end CNN-compatible retargeting-based image augmentation strategy. The task-critical image regions for image retargeting are inferred by techniques [2, 17, 23] that produce visual explanations for CNN decisions. Thus, compared to all the existing retargeting-based data augmentation approaches [18, 22, 26, 28], our approach reasonably conserves the original CNN architecture. In addition, the search space to learn the attention priors for image retargeting is reduced.

- We utilize the Spatial Warper from AIM [26] to increase the spatial coverage of the task-critical image regions. By utilizing the aforementioned visual explanations techniques, we reduced the space complexity of the Spatial Warper from $\mathcal{O}(n)$ to $\mathcal{O}((\log n)^2)$. Here n is the number of pixels in the attention map.

- We propose a simple and lightweight refinement module to refine the priors. An illustration of the refinement process is shown in Fig. 1.

- We evaluated our approach to categorize images in the ImageCLEF2013 [12], ImageCLEF2015 [13], and ImageCLEF2016 [6] datasets. Our approach significantly outperformed **seven** state-of-the-art augmentation approaches on all three datasets.

2 Related Works

2.1 Retargeting-based Image Augmentations for CNNs

Uniform down-sampling of images proportionally reduces the size of important and unimportant regions in an image. In computer graphics [1, 24, 27], retargeting algorithms preserve aesthetically appeasing regions during image downsampling. Recasens *et al*. [22] were the first to utilize retargeting-based data augmentations while training and evaluating CNNs to perform gaze detection and image classification. Their Saliency Sampler detected task-critical regions with another Convolutional Neural Network [14] during the forward pass. Thus, the original architecture of the neural network was not conserved, and the search space to find the task-critical regions was substantially increased. Works derived from the Saliency Sampler [18, 25, 28] face similar issues. S3Ns [9] used the Saliency Sampler to increase the spatial coverage of the task-critical regions inferred by Class Peak Response Maps [31]. However, their approach's space and time complexity are high during the retargeting phase due to the high spatial resolution of the class response maps. STN [16] can crop the regions of interest. However, this transformation is not guaranteed as it is performed implicitly by the network. AIM [26] converts an image to a graph to perform retargeting. However, this approach is computationally expensive as it requires solving a sparse linear system with thousands of graph vertices.

2.2 Visual Explanations for CNNs

Visual explanation methods [2, 11, 17, 19, 21, 23, 30] for CNNs are designed to explain a CNN's decision to a human user. The output of these methods is a localization map highlighting the task-critical regions. CAM [30] identified class-specific discriminative regions in the input images to CNN while performing image classification. However, CAM is only applicable to CNNs without fully connected layers. GradCAM [23] utilizes gradients of a target concept flowing into a convolutional layer to produce a visual explanation for a decision by a CNN. Unlike CAM, GradCAM does not require altering the CNN architecture. GradCAM++ [2] improved upon GradCAM to give better object localization and better explain multiple occurrences of objects. Recently proposed, LayerCAM [17] can reliably provide class activation maps for different CNN layers. While these visual explanation techniques are popular for interpretability purposes and weakly-supervised object detection, their role in end-to-end CNN-compatible retargeting-based algorithms has not been explored. Chen *et al*. [3] used visual explanation techniques to perform task-based image cropping. Unlike them, we use these techniques for image retargeting.

3 Method

3.1 Overview

Our data augmentation approach consists of two stages. In Stage I, we train a Convolutional Neural Network to produce a coarse localization map that highlights the task-critical regions. We consider these localization maps induced by the network as prior knowledge for Stage II. In Stage II, we retrain the Convolutional Neural Network while performing retargeting-based image augmentations. A detailed visual illustration of our approach is shown in Figure 2.

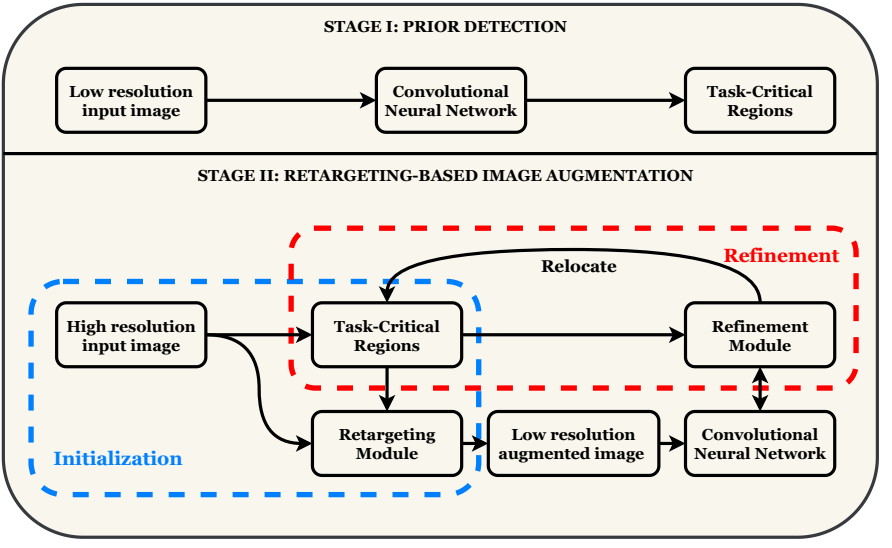


Figure 2: **Overview of our data augmentation approach**. In the first stage **(Stage I)**, a Convolutional Neural Network is trained on the training set containing low-resolution images. Then, any approach that faithfully outputs a visual explanation for the network's decisions is utilized to obtain task-critical regions on the training and testing images. These priors are employed in the second stage (**Stage II**) to perform retargeting-based image augmentations. The second stage is categorized into two sub-categories; **initialization mode** and **refinement mode**. In the initialization mode, higher resolution images (the dataset is the same) are sent to the retargeting module along with the prior. The retargeting module outputs images at a lower resolution, but the spatial coverage of task-critical regions is increased or at least conserved. In the refinement mode, a refinement module jointly optimizes with the network to relocate task-critical regions in the priors acquired from Stage I.

3.2 Stage I: Prior Detection

Ideally, learnable data augmentation approaches should not rely on prior knowledge about the data distribution from a human user. If a learnable augmentation approach can benefit from prior knowledge, it should infer it by itself and without any human-in-the-loop. We design our augmentation approach around this principle. CNN-compatible retargeting-based data augmentation approaches rely on accurate localization maps highlighting task-critical regions. In this paper, we refer to such localization maps as priors. Existing works [22, 26] employed attention mechanisms with a few convolutional layers to explicitly learn these priors. However, we hypothesize that attention mechanisms located at the start of a network cannot faithfully learn the priors. For example, in image classification, research has demonstrated that deeper layers in CNN contain class-discriminative semantic and spatial information. Therefore, expecting an attention mechanism to explicitly detect class-discriminative regions at the beginning of the network is counter-intuitive. We begin Stage I by training a CNN on the training split of a dataset containing low-resolution images. Then we employ existing approaches that can reliably output localization maps containing task-critical regions in the training and testing set. In our experiments, we utilized either GradCAM [23], GradCAM++ [2], or LayerCAM [17] to obtain the priors.

3.3 Stage II: Retargeting-Based Image Augmentation

CNN-compatible retargeting-based data augmentations downsample high-resolution input images so that the spatial coverage of task-critical regions is increased or preserved. While downsampling an image, these retargeting algorithm replicates regions (pixel values in a region) with higher attention values more than regions with lower attention values. However, these approaches are susceptible to image foldovers and extreme transformations. Thus, in this paper, we utilize the Spatial Warper from AIM [26] as our Retargeting Module as it can avoid extreme image transformations. We propose two modes for performing retargeting-based image augmentations. Details on each mode are presented below.

3.3.1 Initialization Mode

In this mode, high-resolution images and the attention priors (from Stage I) are sent to the Retargeting Module. The Retargeting Module increases the spatial coverage of the task-critical image regions while downsampling. It then propagates the low-resolution images to the same CNN used to infer the priors, and this process is repeated until the training loss is converged. The key contribution of our approach is to reduce the computational complexity of the Spatial Warper, as discussed below.

The Spatial Warper in AIM begins retargeting by converting an image's grid representation into a graph. Pixel locations are considered the graph's vertices ($\mathcal{V}$). Edges are defined between a vertex's horizontal and vertical vertex neighbors. Once the graph is defined, the Spatial Warper minimizes an energy function based on the learned attention to infer the new location of the graph vertices. Minimizing the energy function translates to solving a sparse linear system of the form $AX = B$. The vector X is unknown. The matrix $A \in \mathbb{R}^{\mathcal{V}\times\mathcal{V}}$ and the vector $B \in \mathbb{R}^{\mathcal{V}\times 1}$ given in Equation 1 are known. For ease of notation, we have followed the exact definition of A and B in the original paper [26]. In Equation 1, v_i is the location of an i^{th} vertex along an independent direction in space, $\mathcal{N}(v_i)$ is the vertex neighborhood of v_i, and v_j is the vertex neighbor of v_i. γ_i is the attention at the edge between v_i, and v_j. A_{IJ} is a row in A for a vertex v_i and its vertex neighbors.

$$A_{IJ} = \begin{cases} \sum\limits_{v_j \in \mathcal{N}(v_i)} \frac{2}{(v_i - v_j + \varepsilon^{-1})^2}, & \text{if } I = J \\ \frac{-2}{(v_i - v_j + \varepsilon^{-1})^2}, & \text{if } I \neq J \\ 0, & \text{otherwise} \end{cases} \qquad (1)$$

$$B_I = \sum_{\substack{v_j \in \\ \mathcal{N}(v_i)}} \frac{2\gamma_i(v_i - v_j)}{(v_i - v_j + \varepsilon^{-1})^2}$$

Despite being tolerant to extreme transformations, the Spatial Warper has a high space complexity. The reason for this high space complexity is the size of the matrix $A \in \mathbb{R}^{\mathcal{V}\times\mathcal{V}}$.The space complexity of the Spatial Warper is $\mathcal{O}(\mathcal{V}^2)$. Thus, as the number of $\mathcal{V}$ in a graph increases, the space complexity of AIM starts to increase significantly. Moreover, since the new locations of the vertices are learned iteratively by a sparse linear solver, the time complexity of AIM also increases significantly with an increase in the size of $\mathcal{V}$. In our approach, since attention is inferred by visual attention methods near the final convolutional layers of the CNN, the number of vertices in the attention map is significantly lower than in the original implementation of AIM. For example, in the original work, $|\mathcal{V}| = 1024$, and thus, the

space complexity of their approach was 100k vertices. However, in our approach $|\mathcal{V}| = 7$. Compared to AIM, the space complexity of our approach is $\mathcal{O}((\log \mathcal{V}^2)^2)$. Thus, our approach also decreases the time complexity of AIM.

3.3.2 Refinement

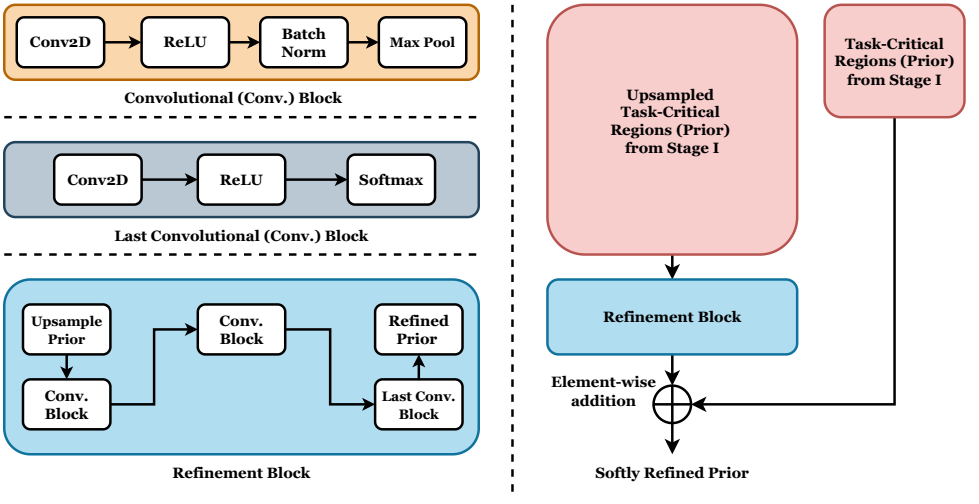


Figure 3: **Architecture and components of our Refinement Module.** We first upsample the attention prior (from Stage I) to have the same dimension as the low-resolution task images. We then process it through our lightweight Refinement Block. The Refinement Block consists of sequentially stacked Convolution (Conv.) Blocks that contain convolutional and pooling layers. These blocks perform channel rising and downsample the feature maps to refine the upsampled prior. The last Conv. Block converts the feature maps to a localization map having the same dimension as the prior (from Stage I). Finally, the attention prior (from Stage I) is softly weighted by adding the Refinement Block's output.

The training and evaluation strategy in the refinement mode is similar to the initialization mode besides one aspect. In the refinement mode, we refine the attention priors (from Stage I) before propagating them to the Retargeting Module. The key contribution here is our novel Refinement Module that softly refines the attention priors to better relocate the task-critical regions in the images. An illustration and details about the Refinement module's components and architecture are given in 3. Unlike other methods that used attention mechanisms to learn disassociation between pose and texture, our Refinement Module only refines the visually explainable decisions by a CNN. Thus, compared to other approaches such as AIM and Saliency Sampler, our approach only requires a few trainable parameters (<1000) to infer the task-critical regions.

3.4 Implementation Details

We implemented our approach using PyTorch [20] and PyTorch Geometric [10]. The number of channels in the first and second Convolutional Block in the Refinement Module is 32 and 64, respectively. The size of the low resolution images is 224px $\times$ 224px. The size of the high resolution images is 448px $\times$ 448px.

4 Evaluation

We evaluated our approach for categorizing images in the ImageCLEF2013 [12], ImageCLEF2015 [13], and the ImageCLEF2016 [6] datasets. These datasets contain diagnostic or general illustrations from biomedical documents. Most images contain noise in the form of writing. In addition, ImageCLEF datasets exhibit high intra-class variance and class imbalance. For example, in the ImageCLEF2016 dataset, some classes contain less than ten images, while 76% of the images belong to only five categories. The distribution of the training and the testing set for each data set is provided in Table 1.

We utilized ResNet [14] with 50 hidden layers (ResNet-50) and DenseNet [15] with 121 hidden layers (DenseNet-121) as our task networks. Both networks were pre-trained on ImageNet [7]. The height and width of images to the task network were 224 pixels. The images were normalized using the mean and standard deviation of images in ImageNet. The batch size was 128, and the networks were trained for 25 epochs. We used the Adam optimizer with average decays β_1 and β_2 set to 0.9 and 0.999, respectively. The learning rate was set to 0.0002. The networks were trained and evaluated on four NVIDIA RTX 2080 TI graphics cards. All experiments were performed using the same settings.

4.1 Baseline

We trained ResNet-50 and DenseNet-121 on the training set of ImageCLEF2013, ImageCLEF2015, and ImageCLEF2016. Then, we evaluated the trained networks on the testing set of these datasets. Test results are reported in Table 2. Classification accuracies in Table 2 serve as a baseline for comparison with all other experiments in the remaining sections.

In our next set of experiments, we trained and tested ResNet-50 and DenseNet-121 using different [4, 5, 8, 22, 26, 29] state-of-the-art augmentation approaches on all three datasets. Results are reported in Table 3. The classification accuracies using data augmentation were generally lower than the baseline results. In particular, no augmentation approach improved the baseline classification accuracy on the ImageCLEF2015 dataset. These experiments demonstrated that the task networks learned undesirable biases because of geometric and photometric perturbations to the original data. In the case of non-learnable augmentations, results suggest that randomly removing regions was not conducive to learning. However, as these approaches do not participate in the learning process, it is impossible to constrain them to remove unimportant regions. In the case of learnable augmentations, it is evident that they did not learn and focus on regions that were conducive to the task. It must be noted that these augmentation approaches have demonstrated excellent results on images of natural scenes but did not perform well for classifying biomedical document images.

# Samples	Image CLEF13	Image CLEF15	Image CLEF16
training	2879	4532	6776
testing	2570	2244	4166

Table 1: The number of (#) samples in the training and testing set of the ImageCLEF2013, ImageCLEF2015, and ImageCLEF2016 datasets.

Model	CLEF13 (Acc%)	CLEF15 (Acc%)	CLEF16 (Acc%)
ResNet50	86.96	75.00	85.33
DenseNet121	87.16	76.07	87.30

Table 2: Classification accuracies (Acc%) of ResNet50 and DenseNet121 on ImageCLEF2013(CLEF13), CLEF15, and CLEF2016 datasets.

Augmentations	Augmentation Type	Models (Acc%)	ImageCLEF13 (Acc%)	ImageCLEF15 (Acc%)	ImageCLEF16
Random Cropping	Train, Geo., Non-Learnable	ResNet	83.11 (↓ **3.85**)	73.57 (↓ **1.43**)	85.41 (↑ **0.08**)
		DenseNet	84.36 (↓ **2.80**)	75.58 (↓ **0.49**)	86.20 (↓ **1.10**)
Random Erasing [29]	Train, Geo., Non-Learnable	ResNet	86.38 (↓ **0.58**)	74.82 (↓ **0.18**)	85.36 (↑ **0.03**)
		DenseNet	85.91 (↓ **1.25**)	75.62 (↓ **0.45**)	87.40 (↑ **0.10**)
CutOut [8]	Train, Geo., Non-Learnable	ResNet	86.65 (↓ **0.31**)	73.89 (↓ **1.11**)	85.45 (↑ **0.12**)
		DenseNet	87.67 (↑ **0.51**)	75.85 (↓ **0.22**)	86.65 (↓ **0.65**)
RandAugment [6]	Train, Geo. + Photo., Non-Learnable	ResNet	86.19 ((↓ **0.77**)	74.55 (↓ **0.45**)	85.43 (↑ **0.10**)
		DenseNet	87.32 (↑ **0.16**)	75.49 (↓ **0.49**)	87.71 (↑ **0.41**)
AutoAugment [1]	Train, Geo. + Photo., Non-Learnable	ResNet	85.29 (↓ **1.67**)	74.69 (↓ **0.31**)	85.67 (↑ **0.34**)
		DenseNet	86.73 (↓ **0.43**)	74.91 (↓ **0.09**)	86.53 (↓ **0.77**)
Saliency Sampler [23]	Train + Test, Geo., Learnable	ResNet	86.34 (↓ **0.62**)	74.82 (↓ **0.18**)	85.74 (↑ **0.41**)
		DenseNet	87.47 (↑ **0.31**)	75.62 (↓ **0.45**)	86.87(↓ **0.43**)
AIM [28]	Train + Test, Geo., Learnable	ResNet	86.77 (↓ **0.19**)	73.75 (↓ **1.25**)	85.29 (↓ **0.04**)
		DenseNet	87.59 (↑ **0.43**)	75.09 (↓ **0.98**)	86.92 (↓ **0.38**)

Table 3: Classification accuracies (Acc%) of ResNet-50 and DenseNet-121 using different image augmentation approaches with on ImageCLEF2013, ImageCLEF2015, and ImageCLEF2016 datasets. Augmentation approaches that achieved lower classification accuracy than the baseline are marked by a downward-facing arrow (↓ **lower than baseline**). Augmentation approaches that achieved higher classification accuracy than the baseline are marked by an upward-facing arrow (↑ **higher than baseline**). Geo. and Photo. stands for geometric and photometric augmentations, respectively.

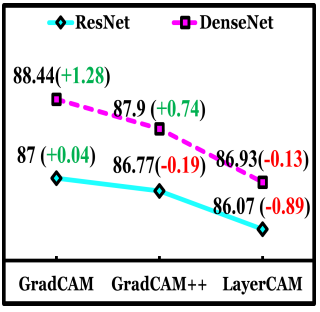
4.2 Quantitative Experimental Results of Our Approach

4.2.1 Performance on Image Classification

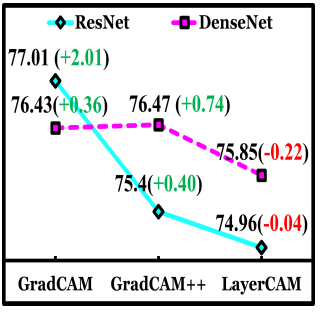
To evaluate our approach we used GradCAM [23], GradCAM++ [3], and LayerCAM [17] to provide the visually explainable prior (Stage I). The task network was either ResNet-50 or DenseNet-121. Then, we re-trained these models from scratch using our data augmentation approach (Stage II). Results are reported in Figure 4. First, we observed that the initialization mode improved the baseline on all data sets for both models. In particular, the attention priors from GradCAM significantly improved the baseline on the ImageCLEF2015 (**2.01%** on the ResNet model). Higher accuracies than the baseline were observed with all attention priors besides the ones obtained from LayerCAM. Our approach in refinement mode also gave higher accuracies than the baseline. Both models obtained higher accuracies on almost all datasets, even after using different visually explainable priors.

4.2.2 Space Complexity

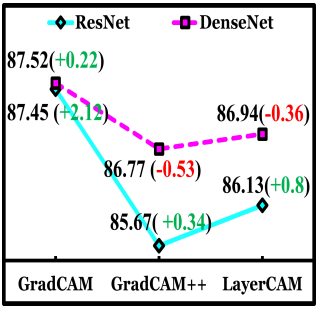
The size of the matrix A_{IJ} in Equation 1 is 1024×1024 for the original implementation of AIM. In our approach, this size reduces to 7×7. This reduction in the size of A_{IJ} can be measured in terms of GPU memory usage. On a single NVIDIA RTX 2080 Ti graphics card, for a batch size of 1 on the DenseNet model, the GPU memory allocated for AIM is 338 MB. In comparison, 304 MB of GPU memory is allocated for our approach in the refinement mode. Thus, our approach reduced the spatial complexity of AIM by nearly 10%. Compared to AIM, the space complexity for our approach further decreased with an increase in batch size. For example, for a batch size of 32, the GPU memory allocated for AIM is 6.3 GB. In comparison, 5.2GB of GPU memory is allocated for our approach in the refinement mode. In the initialization mode of our approach, 4.5GB of GPU memory is allocated.



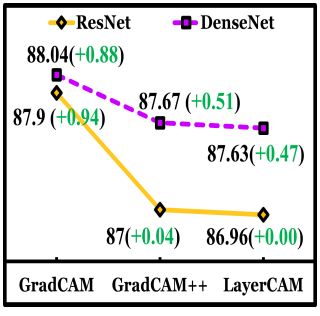
(a) Initialization CLEF2013



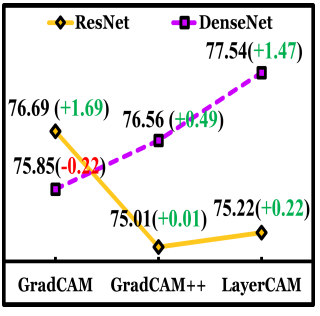
(b) Initialization CLEF2015



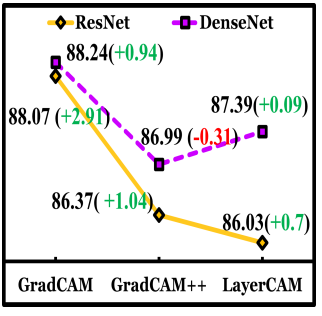
(c) Initialization CLEF2016



(d) Refinement CLEF2013



(e) Refinement CLEF2015



(f) Refinement CLEF2016

Figure 4: **Results of our approach.** The top row reports results using our data augmentation approach in the initialization mode. The bottom row reports results using our data augmentation approach in the refinement mode. In both modes, higher accuracies than the baseline were observed. Naturally, our augmentation approach outperforms all other state-of-the-art image augmentation approaches as well. In particular, results in the refinement mode gave consistently higher accuracies than the baseline. The numbers in the brackets next to the classification accuracy represent the increase (+) or decrease (-) in classification accuracy compared to the baseline. [Best viewed in color.]

5 Discussion and Conclusion

Data augmentation approaches have shown that they can improve the performance of neural networks in classifying images of natural scenes. However, the role of data augmentation techniques for biomedical document images has not been completely explored. Designing augmentation approaches that generalize to biomedical document images can improve the performance of biomedical document retrieval systems. In this work, we introduced a retargeting-based image augmentation approach. We found that accurate attention prior is vital to the performance of the task-based image retargeting techniques. In this work, attention priors were obtained from techniques that provide a visual explanation of decisions by Convolutional Neural Networks. Our data augmentation approach outperformed seven state-of-the-art augmentation approaches on three datasets containing biomedical document images. However, further work will be necessary to demonstrate that our approach generalizes to other tasks. We also demonstrated that our data augmentation approach reduces the space complexity of an existing retargeting algorithm called AIM from $\mathcal{O}(n)$ to $\mathcal{O}((\log n)^2)$.

Acknowledgement

This work was supported by the National Institutes of Health/ National Library of Medicine award R01LM012527.

References

[1] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. In *ACM SIGGRAPH 2007 papers*, pages 10–es. 2007.

[2] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018.

[3] Wenjie Chen, Shuang Ran, Tian Wang, and Lihong Cao. Learning how to zoom in: Weakly supervised roi-based-dam for fine-grained visual classification. In *International Conference on Artificial Neural Networks*, pages 118–130. Springer, 2021.

[4] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[5] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[6] Alba G Seco De Herrera, Stefano Bromuri, Roger Schaer, and Henning Müller. Overview of the medical tasks in imageclef 2016. *CLEF Working Notes. Evora, Portugal*, 2016.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[8] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[9] Yao Ding, Yanzhao Zhou, Yi Zhu, Qixiang Ye, and Jianbin Jiao. Selective sparse sampling for fine-grained image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6599–6608, 2019.

[10] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[11] Ruigang Fu, Qingyong Hu, Xiaohu Dong, Yulan Guo, Yinghui Gao, and Biao Li. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. *arXiv preprint arXiv:2008.02312*, 2020.

[12] Alba Garcia Seco De Herrera, Jayashree Kalpathy-Cramer, Dina Demner-Fushman, Sameer Antani, and Henning Müller. Overview of the imageclef 2013 medical tasks. *CLEF working notes 2013*, 1179:219–232, 2014.

[13] Alba Garcia Seco De Herrera, Henning Müller, and Stefano Bromuri. Overview of the imageclef 2015 medical classification task. In *Working Notes of CLEF 2015–Cross Language Evaluation Forum, CEUR*. CEUR Workshop Proceedings, 2015.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] 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.

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

[17] Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei. Layercam: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 30:5875–5888, 2021.

[18] Chen Jin, Ryutaro Tanno, Thomy Mertzanidou, Eleftheria Panagiotaki, and Daniel C Alexander. Learning to downsample for segmentation of ultra-high resolution images. *arXiv preprint arXiv:2109.11071*, 2021.

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

[20] 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.

[21] Harish Guruprasad Ramaswamy et al. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 983–991, 2020.

[22] Adria Recasens, Petr Kellnhofer, Simon Stent, Wojciech Matusik, and Antonio Torralba. Learning to zoom: a saliency-based sampling layer for neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 51–66, 2018.

[23] 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 *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[24] Vidya Setlur, Saeko Takagi, Ramesh Raskar, Michael Gleicher, and Bruce Gooch. Automatic image retargeting. In *Proceedings of the 4th international conference on Mobile and ubiquitous multimedia*, pages 59–68, 2005.

[25] Vinit Veerendraveer Singh and Chandra Kambhamettu. Feature map retargeting to classify biomedical journal figures. In *International Symposium on Visual Computing*, pages 728–741. Springer, 2020.

[26] Vinit Veerendraveer Singh and Chandra Kambhamettu. Aim: An auto-augmenter for images and meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 722–731, 2022.

[27] Daniel Vaquero, Matthew Turk, Kari Pulli, Marius Tico, and Natasha Gelfand. A survey of image retargeting techniques. In *Applications of Digital Image Processing XXXIII*, volume 7798, pages 328–342. SPIE, 2010.

[28] Xiaohan Xing, Yixuan Yuan, and Max Q-H Meng. Zoom in lesions for better diagnosis: Attention guided deformation network for wce image classification. *IEEE Transactions on Medical Imaging*, 39(12):4047–4059, 2020.

[29] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

[30] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[31] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3791–3800, 2018.