DRAGN: Deep Recusively AggreGation Network

Anonymous CVPR submission

Paper ID ****

Abstract

A lot of computer vision tasks exhibit multi-instance property, in which the learning is performed at the bag-level instead of instance-level and each bag contains of multiple instances, i.e. images. Compared to traditional multi-instance learning algorithms and recent 3D deep neural networks, feature aggregation methods have the following advantages: i)more flexible, as it can be achieved in both traditional and deep learning frameworks, and ii)more versatile, as it can deal with various types of bags without temporal or spatial dependency between images.

In this study, we proposed a deep learning-based feature aggregation model, called DRAGN(Deep Recursively AggreGation Network). It consists of two components, feature aggregation unit and feature aggregation module. The feature aggregation module uses feature aggregation unit to perform iterative and stacked convolution aggregation of multiple instances, and finally output an aggregated feature.

We assess the model performance on two biomedical image processing tasks. One is the protein subcellular localization using immunofluorescence images for human cells, and the other is gene annotation using spatial gene expression images. In both the two tasks, DRAGN outperforms the existing feature aggregation methods and the state-of-the-art models for addressing these two tasks.

1. Introduction

In traditional computer vision tasks, like image classification [1, 2, 3], object detection [4, 5], and semantic segmentation [6, 7], the inputs of computational learning models are single images. As image processing techniques develop rapidly and the storage capacity for multi-media data grows dramatically, there is an increasing need for handling higher dimensional data, such as videos and series of images. Especially, in many image processing applications, the input consists of multiple images, which determine the output jointly. For example, when we recognize person identity based on videos, each video input can be regarded as a time-series of images. Another example is the automatic diagnosis system based on MRI (magnetic resonance imaging) data, which consists of multiple slices presenting the area of interest being scanned. In these two cases, there are either temporal or spatial dependency between images; whereas for a lot more applications, their inputs are neither videos nor 3D models, but just bags of images. There may be no defined spatio-temporal relationship between images, but they have common attributes associated with the bag.

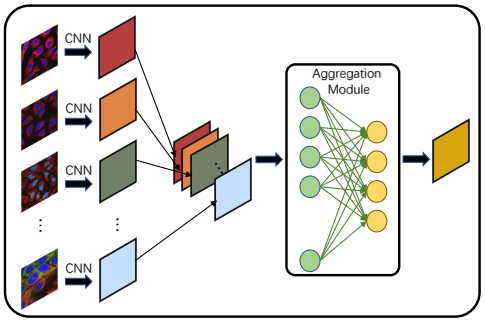

Figure 1: A simple schematic of feature aggregation,The pre-trained CNN model is used to extract features from the input images, and then the extracted features are input into the feature aggregation module to obtain the final aggregation output.

Such multi-image-input (MII) tasks are especially common in the biomedical field [8, 9, 10, 11]. Benefitting from recent advances of microscopic imaging technology, various types of biomedical images have accumulated at an unprecedented rate for the past decade [12, 13, 14, 15]. Image analysis has become a common approach in not only medical diagnosis but also fundamental biological researches from tissue level down to the cell level. Unlike natural images, biomedical images are often much harder to obtain due to the difficulty in preparing the specimen or stringent experimental conditions. Therefore, during the imaging procedure, it is common to capture multiple images for a specimen in a single trial of experiment and perform multiple trails for repeatability. To infer the functions of genes or characteristic of molecules, all the captured im-

ages should be considered comprehensively to give a more accurate judgement on the final output, as single images may only contain partial information and it is important to capture the correlation between images, e.g. the 3D structure inference of biomacromolecules using 2D projection images captured from multiple orientations. It is difficult to develop computational methods for addressing MII tasks. The challengs mainly come from variable-sized input and enven image quality.

The existing approaches to deal with MII learning tasks fall into 3 categories.

i) **Multi-instance learning (MIL) models.** The MII learning tasks can be regarded as a special kind of problems in machine learning, namely multi-instance learning [16]. As MIL is a general learning framework, it can deal with any type of multi-instance input, such as image [17], text [18], and chemical structures [19]. MIL algorithms compute pair-wise similarity or loss function at the bag-level instead of instance-level. Till now, a lot of MIL methods have been developed, like multi-instance KNN [20] and multi-instance support vector machines (SVMs) [21]. Before using these traditional learning models, image features should be extracted separately. Recently, a deep-learning-based MIL model has been proposed, called DeepMIML [22], but it was designed for single-image input and each image is regarded as a bag of objects.

ii) **Feature aggregation methods.** Instead of adapting learning models to multi-instance inputs, this type of methods focuses on feature representation. There are a few ways to combine instance-level features to bag-level representation. For instance, in image retrieval, researchers aggregated the patch-based local descriptors into a global descriptor [23, 24, 25]. Moreover, deep learning models can conveniently achieve feature aggregation by simply using pooling, summation or averaging operations [26].

iii) **3D data learning models.** Recent advances of deep learning methods allow a more straightforward processing of high-dimensional data. For instance, instead of regarding the input as a bag of images, a 3D convolutional neural network (CNN) model learns 3D data samples directly [27].

Apparently, the third type of methods is more suitable for processing video or 3D structural data, in which the images have strong correlations with each other. The first type of methods are mostly traditional shallow learning models. For dealing with image data, extra feature engineering is required. By contrast, the second type is more versatile, including both traditional and deep learning-based models, and can be applied to learn representations for various bags of images.

In this paper, we focus on feature aggregation, because of its flexibility in dealing with various types of image sets, especially originated from biomedical studies, and propose a model, called DRAGN (Deep Recusively AggreGation Network). The model is featured by a recursive aggregation protocol and simple but effective aggregation units, which can address the learning of input bags with varying size. We assess the performance of DRAGN on two biomedical image processing applications, namely the prediction of protein subcellular localization and spatial gene expression data annotation. On both tasks, DRAGN achieves significant improvement on prediction accuracy compared with other feature aggregation methods and the state-of-the-art predictors for these two tasks

2. Relate Work

Feature aggregation is not a new problem in computer vision. It is used to be a part in feature engineering in traditional methods, and it also benefits from recent advances of deep learning techniques.

2.1. Traditional methods

In traditional image processing, feature aggregation aims to fuse features extracted from single images into a comprehensive feature representation before feeding to a learning model. There are three typical feature fusion methods, namely the bag of visual words (BOV) [23], Fisher vector [24], and vector of locally aggregated descriptors (VLAD) [25]. BOV regards image features as words, builds a vocabulary of local image features and generates a vector of their occurrence counts. The Fisher vector method stores the mixing coefficients of the Gaussian mixture model (GMM) as well as the mean and covariance deviation vectors of the individual components. The VLAD method computes the distance of each feature point to the cluster center closest to it. All of these three methods produce a fixed-length feature vector for the input image set, which can work with traditional machine learning models, like SVMs.

2.2. Deep learning-based methods

For the past decade, feature encoding via deep neural networks has almost replaced traditional feature engineering in image processing [28, 29, 30]. Deep learning-based feature aggregation methods have also emerged, as it is natural to fuse features by using the common operations, like pooling, summation or averaging in convolutional neural networks (CNNs). Su et al. pioneered the studies that integrate feature aggregation into CNNs [26]. To achieve 3D recognition by using photos captured from different angles, they designed a Multi-view CNN (MVCNN) model [26] to extract photo features and fuse information from multiple orientations. The feature aggregation strategy in MVCNN is simply a maximization operation but very effective in the task. Later, Yang *et al*. proposed a neural aggregation network for video face recognition, NAN[31], which exploited multiple face images to train the network model. Instead of a simple integration procedure, NAN aggregates multiple

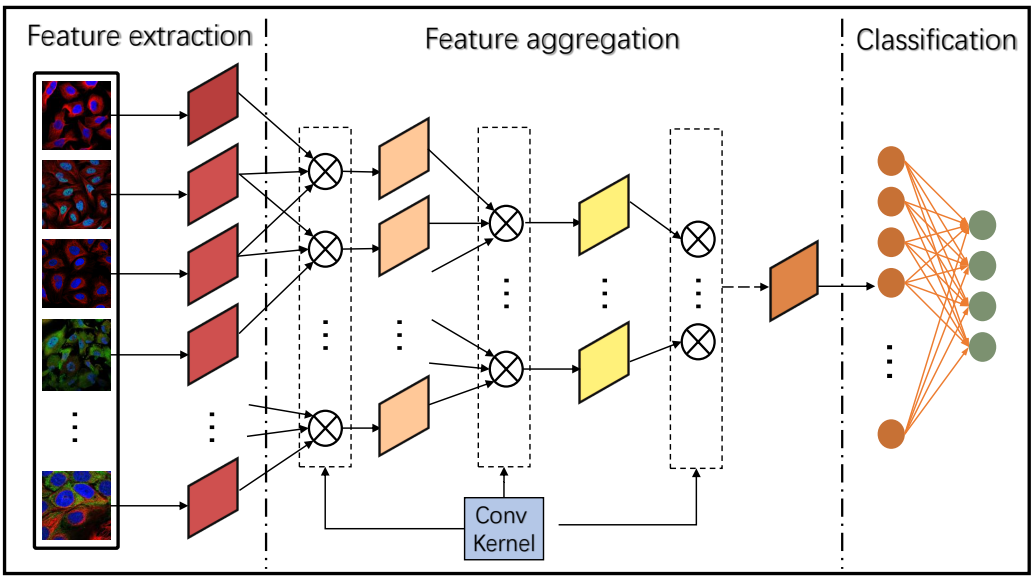

Figure 2: Flowchart of post-aggregation methods

features by an attention module, which assigns high attention scores to important features. Besides, by considering the feature vector of each instance as the input at a time step, Yang *et al*. designed a model called Annofly [8] based on recurrent neural network, which also implemented an attention mechanism to make the model focus on high-quality images.

In summary, these three methods aim to weight multiple input features by suitable weighting coefficients, so as to obtain final aggregated features, based on either CNN or RNN network architecture.

3. Methods

3.1. Problem Description

For a multi-input-image (MII) classification task, we aim to learn a model whose input samples are represented by bags of images (i.e., each sample corresponds to a bag of images) and output vectors denote label probabilities. Let $\mathcal{X}$ denote the sample set, i.e. $\mathcal{X} = \{X_i\}$, where $i \in \{1, 2, ..., n\}$, n is the number of samples, and X_i is a sample; $X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,m}\}$, where m is the number of images of the ith sample, $x_{i,j}$ ($j \in \{1, 2, ..., m\}$) denotes an image of X_i. Let $\mathcal{Y} = \{Y_i\}$ be the output space, and $Y_i = \{y_1, y_2, \ldots, y_k\}$ corresponds to the label set of X_i. The goal is to learn a mapping function $f : \mathcal{X} \mapsto \mathcal{Y}$.

Generally, there are two frameworks to aggregate multiple instances into a fixed-length input for learning models, namely pre-aggregation and post-aggregation, which differ in the order of performing feature extraction and feature aggregation.

Pre-aggregation first combines multiple instances, by organizing them into a certain data structure, and then extract features for the composite data samples. FlyIt [11] is a typical pre-aggregation model, which first stitch images belong to the same bag into large images and then extract features from the large images. Since the combination of multiple raw data samples are very complex in most cases, there is often a strong restriction on the number of combined samples. By contrast, post-aggregation first extract features for raw instances and then perform the aggregation operation, as shown in Fig. 2. For instance, MIMT-CNN [9] employs 10 VGG models to learn feature representation from 10 images respectively, and concatenate them into a bag representation for the following classification.

DRAGN belongs to the second type, and it integrates feature learning, aggregation and classification in a deep recursive neural network.

3.2. Model architecture

For MII tasks, a major challenge is how to effectively represent the multi-instance input. Most of the existing methods, no matter pre-aggregation or post-aggregation, set a fixed number of instances to feed the model and perform feature fusion in a single run. For example, MIMT-CNN concatenates feature representations from 10 images, Annofly designs an RNN model of 10 time-steps and feeds at most 10 images to the RNN to learn an embedding for the images, and FlyIt stitches 4 raw images into large images.

An obvious limitation of these methods is the lack of

flexibility in the model architecture and the weakness in handling highly variable sizes of input bags. In contrast to the fixed-input and single-run schema, DRAGN implements a new protocol and sets no limit on the number of input instances. It recursively runs feature aggregation, where the running times depends on the number of instances. The model architecture is shown in Fig. 2.

By regarding the input bag as a sequence of images, DRAGN slides a fixed-width window over the sequence with step size 1. Within each window, a kernel function (defined in Section 3.3) is called to yield an aggregated feature representation for the images within the window. Then these aggregated feature vectors compose a new sequence. The feature aggregation operation is performed in a recursive way until the sequence contains feature representations less than L, then an averaging operation is used to get the final representation. The formal description of the recursive procedure is provided in Algorithm 1.

Algorithm 1

Input: The set of images for an input bag X_i.
$X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,m}\}$,
Output: Feature representation O_i for the bag X_i.
1: Let w be the window size, L be the sequence length ($L = m$), and f_{agg} be the aggregation function.
2: $L = L - w + 1$.
3: **for** each $k \in [1, L]$ **do**
4: $O_k = f(x_k, \ldots, x_{k+w-1})$.
5: **end for**
6: **if** $L < w$ **then**
7: O_i= mean(O_k), $k \in [1, L]$, return.
8: **end if**
9: Goto Step 2.

3.3. Feature aggregation unit

The aggregation units in DRAGN mainly involve convolution computations. Formally, let $x_1, x_2, \ldots, x_n$ be the feature maps to be aggregated, where n is the number of feature maps. The aggregation is defined in Eq. (1),

$$\begin{aligned} \mathbf{X} &= [x_1, x_2, \ldots, x_n], \\ O &= \mathbf{X} * \mathbf{W} + b, \end{aligned} \tag{1}$$

where $\mathbf{X}$ is a tensor composed by the feature maps, $*$ is the convolution operator, $\mathbf{W}$ is the convolutional filter, b is the bias, and O is the aggregated feature map.

We call the feature aggregation unit as L1Agg. To meet the needs from different tasks, in addition to L1Agg, we design two other more complex aggregation units, namely L2Agg and L3Agg, defined in Eqs. 2 and 3, respectively.

$$O = \mathbf{W} * f(g(\mathbf{X} * \mathbf{W} + b)) + b, \tag{2}$$

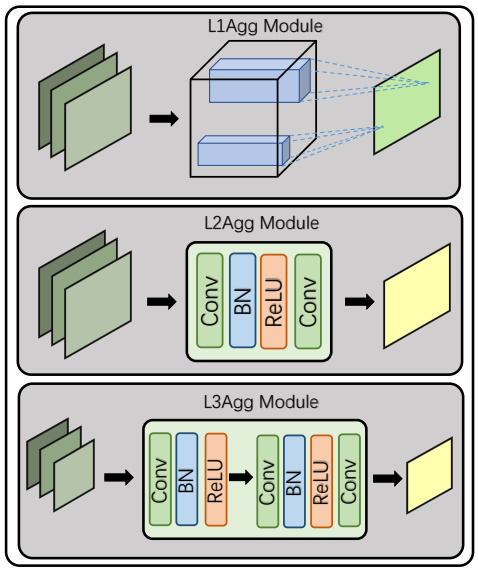

Figure 3: Feature aggregation unit

where $g(\cdot)$ is a normalization function, $f(\cdot)$ denotes the ReLU function.

$$O = \mathbf{W} * f(g(\mathbf{W} * f(g(\mathbf{X} * \mathbf{W} + b)) + b)) + b, \tag{3}$$

As shown in Fig. 3, these three types of units differ in complexity, whereas their working mechanism is the same. And, during the recursive procedure, the convolutional kernel parameters are shared by all aggregation operations.

4. Experiments

We assess the performance of DRAGN on two MII tasks. The first task aims to predict protein subcellular localization based on immunofluorescence (IF) microscopic images. As it is known that, the location information of proteins in the cells is very important for revealing protein functions. Microscopic images can visualize complex localization patterns of proteins, thus make the prediction more accurate and allow tissue-specific studies. This is a typical MII task. Each protein corresponds to multiple microscopic images, and the labels, i.e. cellular locations, are predicted based on all localization patterns implied in these images. However, most of the existing methods convert it into a common image-annotation task. As in the Kaggle competition for classifying subcellular protein patterns in human cells held in January 2019 (www.kaggle.com/c/human-proteinatlas- image-classification), the contest task is to assign location labels to single images. Here we predict labels at the protein-level via the proposed feature aggregation method, and compare with both single-instance learning models and other feature aggregation models.

The second task aims to assign functional annotation terms to genes based on their expression images, which visualize spatial distribution patterns of gene expression in tissues and help to reveal gene functions. Similar to the first task, the annotation is at the gene-level, and each gene has a set of expression images captured in different angles or experimental trials. We compare the performance of DRAGN on a large-scale gene expression dataset of *Drosophila* embryos with the state-of-the-art methods designed for this annotation task.

4.1. Task 1: Prediction of protein subcellular localization

4.1.1 Data source

The datasets of Task 1 are collected from the cell atlas of human protein atlas (HPA) database [32]. We download 1600 bags of IF images, each of which corresponds to a protein. The location annotations of proteins are hand-crafted in the database. The total number of images is 19,777. The sizes of raw images include 800×800, 1728×1728 to 2048×2048. Obviously, the large size of images and small numbers of bags bring great computation difficulties. Considering that there are usually multiple cells within an image, we adopt the selective search algorithm [33] to pick the cellular patches from raw images, for reducing image size and making the model focus on localization patterns within cells in the meantime. As shown in Fig. 4, the algorithm segments out key regions of cells from the non-informative background. We resize the selected patches into a fixed size 512×512, and construct a new dataset. Besides, for data augmentation, we also split the patches of the same protein into several bags. Finally, we obtain a total of 24,052 bags and 185,169 patches.

4.1.2 Experimental Setting

We implement DRAGN under the pytorch framework on a PC with 2.4GHz CPU, 32GB RAM, and two 2080Ti graphics cards. When optimizing the network, we adopt Adam optimizer [34], we set the learning rate of the network, β_1, β_2 to 0.0001, 0.9 and default value respectively. The batch size of the network is set as 16. In this experiment, the loss function is the standard BCE loss. The selected feature extraction network is ResNet50 [35], and the final output layer consists of 10 nodes (to predict 10 major cellular organelles). As it is a multi-label classification task (a protein may locate at multiple organelles), we use 5 common multi-label classification metrics, namely AUC value, macro precision, macro recall, macro F_1 and micro F_1.

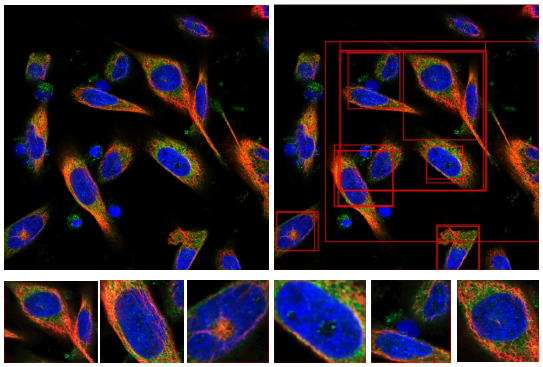

Figure 4: Human protein atlas image has size of 2048×2048, and each image contained numerous identical organelles(top left).We use selective search algorithm to pick out the organelle in the original image(top right), and the captured iamges are shown in the bottom.

4.1.3 Experimental results

To assess the performance of DRAGN, we compare it with two series models, including two baseline models and 3 other deep learning-based feature aggregation models.

The two baseline models share basic model component and settings with DRAGN. The first baseline model is a single-instance learning model, denoted by SI, i.e., the input is a single image. As previous works, by transferring the bag labels to all the images in the bag, the model learns a mapping from single images to location labels. In the test phase, for each bag, the images within the bag are predicted one by one, and their results are integrated (maximum values are extracted) to yield the final labels for the bag. The second baseline model adopts the same training mode as DRAGN, except that it simply averages multiple features for feature aggregation, named MI.

The deep learning-based feature aggregation models, include MVCNN [26], NAN [31], and SPoc [36]. MVCNN takes the maximum value of multi-instance features for aggregation, NAN learns adaptive weights between multi-instance features by introducing an attention mechanism, SPoc uses radial basis functions to aggregate multi-instance features.

The experimental results are shown in Table 1. As can be seen, the performance of SI is extremely poor on this task, suggesting the advantages of feature aggregation for multiple images over single-instance learning on the MII tasks. MI, MVCNN and DRAGN adopt different aggregation strategy. The first two methods use the average and maximum value in the aggregation, respectively. MI performs the worst, while MVCNN and DRAGN have very close performance. However, MVCNN has a

Method	AUC(%)	Macro F_1(%)	Micro F_1(%)
SI	95.56	27.95	68.32
MI	95.59	62.56	**81.96**
MVCNN	96.08	62.7	81.79
NAN	95.86	58.18	77.82
SPoc	93.35	54.72	75.97
DRAGN	**96.56**	**65.15**	81.69

Table 1: Performance comparison on Task 1

Method	AUC(%)	Macro F_1(%)	Micro F_1(%)
ML_{LS}	80.92	54.99	60.17
PMK_{comp}	76.66	50.02	55.86
LR_{SOFT}	82.95	57.72	62.28
E-MIMLSVM	84.60	59.80	64.00
HMIML	85.30	63.98	66.73
AnnoFly	93.67	70.17	71.27
FlyIT	93.59	71.81	71.08
DRAGN	**94.77**	**74.74**	**74.79**

Table 2: Performance comparison on Task 2

much lower macro F_1 than DRAGN. The potential reason is that MVCNN may have an uneven prediction performance for major and minor classes (macro F_1 is average F_1 over classes). As MVCNN takes maximum values of features, it focuses more on the dominant patterns, whereas the minor classes usually correspond to infrequent patterns, thus has relatively low accuracy.

4.2. Task 2: Annotation of spatial gene expression patterns

4.2.1 Data source

In this study, we use standard gene expression images of *Drosophila* embryos from the FlyExpress database [37]. All the images were extracted from the Berkeley *Drosophila* Genome Project (BDGP) (`www.fruitfly.org`) [38, 39] and preprocessed to a uniform size 180×320. We download more than 4000 bags including over 10,000 images. Each bag of images presents the expression distribution on the *Drosophila* embryo for a single gene, in which the images were captured from different orientations, i.e. dorsal, lateral and ventral. We use the dataset as FlyIT [11], where the dataset is divided (at bag-level) into training, validation and test sets according to the ratio of 4:1:5. The total number of labels is 10, each of which is an ontology term describing anatomical and developmental properties. This task is also a multi-label classification problem, as each gene may have multiple developmental terms.

4.2.2 Experimental Setting

The experimental settings are almost the same as Task 1, except that we adopt a more lightweight network ResNet18 for feature extraction, due to the smaller size of images and fewer images in bags.

4.2.3 Experimental results

As BDGP is the largest gene expression data source, a lot of computational methods have been proposed for the automatic image annotation in the database. Here we compare DRAGN with the existing approaches, including both shallow learning models and deep neural networks.

The experimental results are shown in Table 2. The first 4 methods are traditional models while the last 4 models are deep learning models. ML_{LS} is a method utilizing the shared subspace multi-label formulation and bag-of-words feature extraction [40]. PMK_{comp} is the pyramid match kernel (PMK) algorithms, where the subscript "comp" denotes the composite kernels [40]. LR_{soft} is a feature representation scheme based on sparse coding and incorporating the correlations among terms [41]. E-MIMLSVM is an MIML algorithm which adapts the kernel function of support vector machines to the MIML scenario [42]. HMIML and FlyIT[11] both adopt image stitching to combine raw images but use different CNN architectures for learning, thus belong to pre-aggregation methods. AnnoFly [8] first extracts features via ResNet, then employs RNN to deal with the multi-instance input, thus it is a post-aggregation model. As can been seen, the deep learning models improve the prediction accuracy significantly compared with shallow models. Among the traditional methods, E-MIMLSVM, which is the only multi-instance model, performs the best. AnnoFly and FlyIT have very close performance. DRAGN outperforms the second best method FlyIT by around 3% on F_1 values and over 1% on AUC. This experiment again demonstrates the advantage of the proposed recursive feature aggregation method over previous multi-instance learning schemas.

4.3. Visualized analysis

To further investigate the quality of aggregated features by DRAGN, we perform two kinds of visualization analysis. The results are described in the following subsections.

4.3.1 Aggregated features are more differentiable

In the first analysis, we visualize the features representations learned by deep CNN before and after aggregation, by projecting the features onto a 2D space via tSNE algorithm [43]. We use the dataset from Task 2. Fig. 5 shows the 2D distribution of features learned by ResNet. We perform a feature aggregation based on the same features, and Fig. 6 shows the 2D distribution of aggregated features. Different colors denote different classes. Note that as Fig. 5 shows

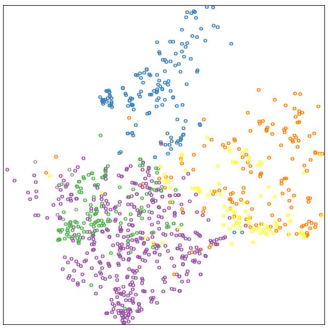

Figure 5: Raw image features

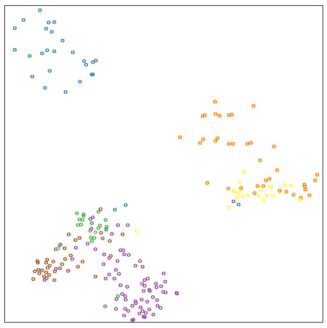

Figure 6: Aggregated features

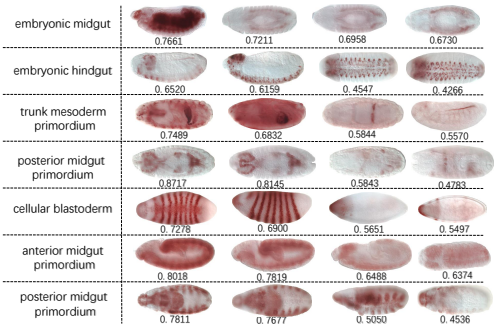

Figure 7: The weight distribution between the aggregated feature and the original images.

the distribution of single images, while Fig. 6 shows the distribution of bags, thus the number of points in Fig. 6 are much fewer than in Fig. 5. It can be seen that the features of different categories before aggregation have small margin and large overlap, which brings great difficulties to subsequent classifiers. After aggregation, samples of different categories are much more separated, with small overlap, which is beneficial for subsequent classification.

4.3.2 Aggregated features retain more information from high-quality images

From Section 4.3.1, we can observe that the feature aggregation make the image patterns more distinguishable. In order to get more insight on the aggregation operation, we compute the cosine similarity between aggregated features and the features of input images, and take the similarity value as the score of the input images. Fig. 7 visualizes the images with different scores. Each row shows images from the same bag, the first column lists typical labels for the bag. As can be seen, the images with high scores contain relatively obvious local patterns corresponding to the target labels, suggesting the feature aggregation retain more information from the high-quality images, i.e. the features aggregated by DRGAN will automatically focus on important features.

5. Discussion and conclusion

This paper proposes a general network framework DRAGN for dealing with the MII problems. DRAGN is featured by a deep recursive architecture and convolution-based aggregation unit. The advantages of DRAGN are summarized as follows.

i) High accuracy and good applicability to various multi-instance image classification tasks; ii) The flexibility of allowing variable sizes of input bags without a restriction on the bag size. iii) Simplicity in aggregating features.

Although our model has shown excellent performance in both of these two tasks, but the scope of our model is not limited to these two tasks. The DRAGN we proposed is a general feature aggregation network, we can easily change its components to make it suitable for other tasks and improve their performance. As a future work, we will investigate the impact of recursion depth on the aggregated features, and explore the potential of fusing features generated in different recursion depth to obtain more comprehensive feature representations.

References

[1] I. Sutskever A. Krizhevsky and G.E. Hinton. Imagenet classification with deep convolutional neural netowrk. *In NIPS*, 2012. 1

[2] K. Simony and A.Zisserman. Very deep convolutional networks for large-scale image recognition. *In ICLR*, 2015. 1

[3] Y.Jia P.Sermanet S. Reed D. Anguelov D.Erhan V. Vanhoucke C. Szegedy, W.Liu and A. Rabinovich. Going deeper with convolutions. *In CVPR*, 2015. 1

[4] T.Darrell R. Girshick, J.Donahue and J.Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *In CVPR*, 2014. 1

[5] L.Bourdev S. Maji B.Hariharan, P.Arbelaez and J.Malik. Semantic contours from inverse detectors. *In ICCV*, 2011. 1

[6] L.Najman C.Farabet, C.Couprie and Y.LeCun. Learning hierarchical features for scene labeling. *TPAMI*, 35(8):1915–1929, 2013. 1

[7] P.Yadollaphour M.Mostajabi and G.Shakhnarovish. Feedforward semantic segmentation with zoom-out features. *In CVPR*, 2015. 1

[8] Yang Yang, Mingyu Zhou, Qingwei Fang, and Hong-Bin Shen. Annofly: annotating drosophila embryonic images based on an attention-enhanced rnn model. *Bioinformatics (Oxford, England)*, 35:2834–2842, 08 2019. 1, 3, 6

[9] Tao Zeng and Shuiwang Ji. Deep convolutional neural networks for multi-instance multi-task learning. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 579–588. IEEE, 2015. 1, 3

[10] Ying-Xin Li, Shuiwang Ji, Sudhir Kumar, Jieping Ye, and Zhi-Hua Zhou. Drosophila gene expression pattern annotation through multi-instance multi-label learning. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 9(1):98–112, 2012. 1

[11] Yang Yang Wei Long, Tiange Li and Hong-Bin Shen. Flyit: Drosophila embryogenesis image annotation based on image tiling and convolutional neural networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2019. 1, 3, 6

[12] Akihiko Yoshizawa Hiroki Tokunaga, Yuki Teramoto and Ryoma Bise. Adaptive weighting multi-field-of-view cnn for semantic segmentation in pathology. *In CVPR*, 2019. 1

[13] Xiaofei Wang Lai Jiang Liu Li, Mai xu and Hanruo Liu. Attention based glaucoma detection: A large-scale database and cnn model. *In CVPR*, 2019. 1

[14] Xinwei Sun Zhen Zhou Fandong Zhang, Ling Luo and Xiuli Li. Cascaded generative and discriminative learning for microcalcification detection in brease mammograms. *In CVPR*, 2019. 1

[15] Lei Huang Li Liu Fan Zhu Shanshan Cui Yi Zhou, Xiaodong He and Ling Shao. Collaborative learning of semi-supervised segmentation and classification for medical images. *In CVPR*, 2019. 1

[16] Zhi-Hua Zhou. Multi-instance learning: A survey. *Department of Computer Science & Technology, Nanjing University, Tech. Rep*, 2004. 2

[17] Zheng-Jun Zha, Xian-Sheng Hua, Tao Mei, Jingdong Wang, Guo-Jun Qi, and Zengfu Wang. Joint multi-label multi-instance learning for image classification. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 2

[18] Zhi-Hua Zhou, Kai Jiang, and Ming Li. Multi-instance learning based web mining. *Applied Intelligence*, 22(2):135–147, 2005. 2

[19] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997. 2

[20] Jun Wang and Jean-Daniel Zucker. Solving multiple-instance problem: A lazy learning approach. 2000. 2

[21] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Advances in neural information processing systems*, pages 577–584, 2003. 2

[22] Ji Feng. Deep miml network. 01 2017. 2

[23] Jun Yang, Yu-Gang Jiang, Alexander Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. pages 197–206, 01 2007. 2

[24] J. Sanchez F.Perronnin, Y.Liu and H. Poirier. Large-scale image retrieval with compressed fisher vectors. *In The Twenty-Third IEEE Conference on Computer vision and Pattern Recognition, CVPR,*, pages 3384–3391, 2010. 2

[25] C.Schmid H. Jegou, M. Douze and P.Perez. Aggregating local descriptors into a compact image representation. *In The Twenty-Third IEEE Conference on Computer vision and Pattern Recognition, CVPR,*, pages 3304–3311, 2010. 2

[26] Evangelos Kalogerakis Hang Su, Subhransu Maji and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. *In ICCV*, 2015. 2, 5

[27] Konstantinos Kamnitsas, Christian Ledig, Virginia Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical Image Analysis*, 36:61–78, 2017. 2

[28] J.Sullivan A.S. Razavin, H.Azizpour and S.Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. *In DeepVision workshop*, 2014. 2

[29] I.Laptev [3] M.Oquad, L.Bottou and J.Sivic. Learning and transferring mid-level image representatons using convolutional neural networks. *In IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1717–1724, 2014. 2

[30] A. Vedaldi K.Chatfield, K.Simonyan and A.Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *In British Machine Vision Conference, MNVC*, 2014. 2

[31] D.Chen F.Wen H.Li J.Yang, P.Ren and G.Hua. Neural aggregation network for video face recognition. *arxiv preprint arxiv:1603.05474*, 2016. 2, 5

[32] Mathias Uhlen et al. Towards a knowledge-based human protein atlas. *Nature biotechnology*, 28(12):1248, 2010. 5

[33] Jasper Uijlings, K E Sande, Theo Gevers, and Arnold W M Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013. 5

[34] D.Kingma and J.Ba. Adam: A method for stochastic optimization. *Arxiv preprint arxiv:1312.6117*, 2013. 5

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 2016. 5

[36] Artem Babenko and Victor Lempitsky. Aggregating deep convolutional features for image retrieval. *In ICCV*, 2015. 5

[37] Sudhir Kumar, Charlotte Konikoff, Bernard Van Emden, Christopher Busick, Kailah T Davis, Shuiwang Ji, Lin Wei Wu, Hector Ramos, Thomas Brody, Sethuraman Panchanathan, et al. Flyexpress: visual mining of spatiotemporal patterns for genes and publications in drosophila embryogenesis. *Bioinformatics*, 27(23):3319–3320, 2011. 6

[38] Pavel Tomancak, Amy Beaton, Richard Weiszmann, Elaine Kwan, Shengqiang Shu, Suzanna E Lewis, Stephen Richards, Michael Ashburner, Volker Hartenstein, Susan E Celniker, et al. Systematic determination of patterns of gene expression during drosophila embryogenesis. *Genome Biology*, 3(12):1–14, 2002. 6

[39] Pavel Tomancak, Benjamin P Berman, Amy Beaton, Richard Weiszmann, Elaine Kwan, Volker Hartenstein, Susan E Celniker, and Gerald M Rubin. Global analysis of patterns of gene expression during drosophila embryogenesis. *Genome Biology*, 8(7):1–24, 2007. 6

[40] Shuiwang Ji, Ying-Xin Li, Zhi-Hua Zhou, Sudhir Kumar, and Jieping Ye. A bag-of-words approach for drosophila gene expression pattern annotation. *BMC bioinformatics*, 10(1):119, 2009. 6

[41] Shuiwang Ji, Lei Yuan, Ying-Xin Li, Zhi-Hua Zhou, Sudhir Kumar, and Jieping Ye. Drosophila gene expression pattern annotation using sparse features and term-term interactions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 407–416. ACM, 2009. 6

[42] Ying-Xin Li, Shuiwang Ji, Sudhir Kumar, Jieping Ye, and Zhi-Hua Zhou. Drosophila gene expression pattern annotation through multi-instance multi-label learning. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 9(1):98–112, January 2012. 6

[43] Laurens Van Der Maaten and Geoffrey E Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 6