Deep Model Based Transfer and Multi-Task Learning for Biological Image Analysis

Wenlu Zhang[1], Rongjian Li[1], Tao Zeng[1], Qian Sun[2], Sudhir Kumar[3,4,5], Jieping Ye[6,7] and Shuiwang Ji[1]

[1]Dept. of Computer Science, Old Dominion University, Norfolk, VA, 23529
[2]Dept. of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287
[3]Institute for Genomics and Evolutionary Medicine, Temple University, Philadelphia, PA 19122
[4]Dept. of Biology, Temple University, Philadelphia, PA 19122
[5]Center of Excellence in Genomic Medicine Research, King Abdulaziz University, Jeddah, Saudi Arabia
[6]Dept. of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI 48109
[7]Dept. of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109

ABSTRACT

A central theme in learning from image data is to develop appropriate image representations for the specific task at hand. Traditional methods used handcrafted local features combined with high-level image representations to generate image-level representations. Thus, a practical challenge is to determine what features are appropriate for specific tasks. For example, in the study of gene expression patterns in *Drosophila melanogaster*, texture features based on wavelets were particularly effective for determining the developmental stages from *in situ* hybridization (ISH) images. Such image representation is however not suitable for controlled vocabulary (CV) term annotation because each CV term is often associated with only a part of an image. Here, we developed problem-independent feature extraction methods to generate hierarchical representations for ISH images. Our approach is based on the deep convolutional neural networks (CNNs) that can act on image pixels directly. To make the extracted features generic, the models were trained using a natural image set with millions of labeled examples. These models were transferred to the ISH image domain and used directly as feature extractors to compute image representations. Furthermore, we employed multi-task learning method to fine-tune the pre-trained models with labeled ISH images, and also extracted features from the fine-tuned models. Experimental results showed that feature representations computed by deep models based on transfer and multi-task learning significantly outperformed other methods for annotating gene expression patterns at different stage ranges. We also demonstrated that the intermediate layers of deep models produced the best gene expression pattern representations.

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.
© 2015 ACM. ISBN 978-1-4503-3664-2/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2783258.2783304.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications - Data Mining

General Terms

Algorithms

Keywords

Deep learning; transfer learning; multi-task learning; image analysis; bioinformatics

1. INTRODUCTION

A general consensus in image-related research is that different recognition and learning tasks may require different image representations. Thus, a central challenge in learning from image data is to develop appropriate representations for the specific task at hand. Traditionally, a common practice is to hand-tune features for specific tasks, which is time-consuming and requires substantial domain knowledge. For example, in the study of gene expression patterns in *Drosophila melanogaster*, texture features based on wavelets, such as Gabor filters, were particularly effective for determining the developmental stages from *in situ* hybridization (ISH) images [24]. Such image representation, often referred to as "global visual features", is not suitable for controlled vocabulary (CV) term annotation because each CV term is often associated with only a part of an image, thereby requiring an image representation of local visual features [8, 27]. Current state-of-the-art systems for CV term annotation first extracted local patches of an image and computed local features which are invariant to certain geometric transformations (e.g., scaling and translation). Each image was then represented as a bag of "visual words", known as the "bag-of-words" representation [7], or a set of "sparse codes", known as the "sparse coding" representation [9, 19, 25].

In addition to being problem-dependent, a common property of traditional feature extraction methods is that they are "shallow", because only one or two levels of feature extraction was applied, and the parameters for computing features are usually not trained using supervised algorithms.

Given the complexity of patterns captured by biological images, these shallow models of feature extraction may not be sufficient. Therefore, it is desirable to develop a multi-layer feature extractor, alleviating the tedious process of manual feature engineering and enhancing the representation power.

In this work, we proposed to employ the deep learning methods to generate representations of ISH images. Deep learning models are a class of multi-level systems that can act on the raw input images directly to compute increasingly high-level representations. One particular type of deep learning models that have achieved practical success is the deep convolutional neural networks (CNNs) [13]. These models stack many layers of trainable convolutional filters and pooling operations on top of each other, thereby computing increasingly abstract representations of the inputs. Deep CNNs trained with millions of labeled natural images using supervised learning algorithms have led to dramatic performance improvement in natural image recognition and detection tasks [6, 10, 18].

However, learning a deep CNN is usually associated with the estimation of millions of parameters, and this requires a large number of labeled image samples. This bottleneck currently prevents the application of CNNs to many biological problems due to the limited amount of labeled training data. To overcome this difficulty, we proposed to develop generic and problem-independent feature extraction methods , which involves applying previously obtained knowledge to solve different but related problems. This is made possible by the initial success of transferring features among different natural image data sets [4, 17, 26]. These studies trained the models on the ImageNet data set that contains millions of labeled natural images with thousands of categories. The learned models were then applied to other image data sets for feature extraction, since layers of the deep models are expected to capture the intrinsic characteristics of visual objects.

In this article, we explored whether the transfer learning property of CNNs can be generalized to compute features for biological images. We proposed to transfer knowledge from natural images by training CNNs on the ImageNet data set. To take this idea one step further, we proposed to fine-tune the trained model with labeled ISH images, and resumed training from already learned weights using multi-task learning schemes. The two models were then both used as a feature extractors to compute image features from *Drosophila* gene expression pattern images. The resulting features were subsequently used to train and validate our machine learning method for annotating gene expression patterns. The overall pipeline of this work is given in Figure 1.

Experimental results show that our approach of using CNNs outperformed the sparse coding methods [19] for annotating gene expression patterns at different stage ranges. In addition, our results indicated that the transfer and fine-tuning of knowledge by CNNs from natural images is very beneficial for producing high-level representations of biological images. Furthermore, we showed that the intermediate layers of CNNs produced the best gene expression pattern representations. This is because the early layers encode very primitive image features that are not enough to capture gene expression patterns. Meanwhile, the later layers captured features that are specific to the training natural image set, and these features may not be relevant to gene expression pattern images.

2. DEEP MODELS FOR TRANSFER LEARNING AND FEATURE EXTRACTION

Deep learning models are a class of methods that are capable of learning hierarchy of features from raw input images. Convolutional neural networks (CNNs) are a class of deep learning models that were designed to simulate the visual signal processing in central nervous systems [1, 10, 13]. These models usually consist of alternating combination of convolutional layers with trainable filters and local neighborhood pooling layers, resulting in a complex hierarchical representations of the inputs. CNNs are intrinsically capable of capturing highly nonlinear mappings between inputs and outputs. When trained with millions of labeled images, they have achieved superior performance on many image-related tasks [10, 13, 18].

A key challenge in applying CNNs to biological problems is that the available labeled training samples are very limited. To overcome this difficulty and develop a universal representation for biological image informatics, we proposed to employ transfer learning to transfer knowledge from labeled image data that are problem-independent. The idea of transfer learning is to improve the performance of a task by applying knowledge acquired from different but related task with a lot of training samples. This approach of transfer learning has already yielded superior performance on natural image recognition tasks [4, 14, 17, 23, 26].

In this work, we explored whether this transfer learning property of CNNs can be generalized to biological images. Specifically, the CNN model was trained on the ImageNet data containing millions of labeled natural images with thousands of categories and used directly as feature extractors to compute representations for ISH images. In this work, we applied the pre-trained VGG model [18] that was trained on the ImageNet data to perform several computer vision tasks, such as localization, detection and classification. There are two pre-trained models in [18], which are "16" and "19" weight layers models. Since these two models generated similar performance on our ISH images, we used the "16" weight layers model in our experiment. The VGG architecture contains 36 layers. This network includes convolutional layers with fixed filter sizes and different numbers of feature maps. It also applied rectified non-linearity, max-pooling to different layers.

More details on various layers in the VGG weight layer model are given in Figure 2. Since the output feature representations of layers before the third max pooling layer involve larger feature vectors, we used each *Drosophila* ISH image as input to the VGG model and extracted features from layers 17, 21, 24, and 30 to reduce the computational cost. We then flattened all the feature maps and concatenated them into a single feature vector. For example, the number of feature maps in layer 21 is 512, and the corresponding size of feature maps is 28×28. Thus, the corresponding size of feature vector for this layer is 401,408.

3. DEEP MODELS FOR MULTI-TASK LEARNING

In addition to the transfer learning scheme described above, we also proposed a multi-task learning strategy in which a CNN is first trained in the supervised mode using the ImageNet data and then fine-tuned on the labeled ISH *Drosophila* images. This strategy is different from the pre-trained model

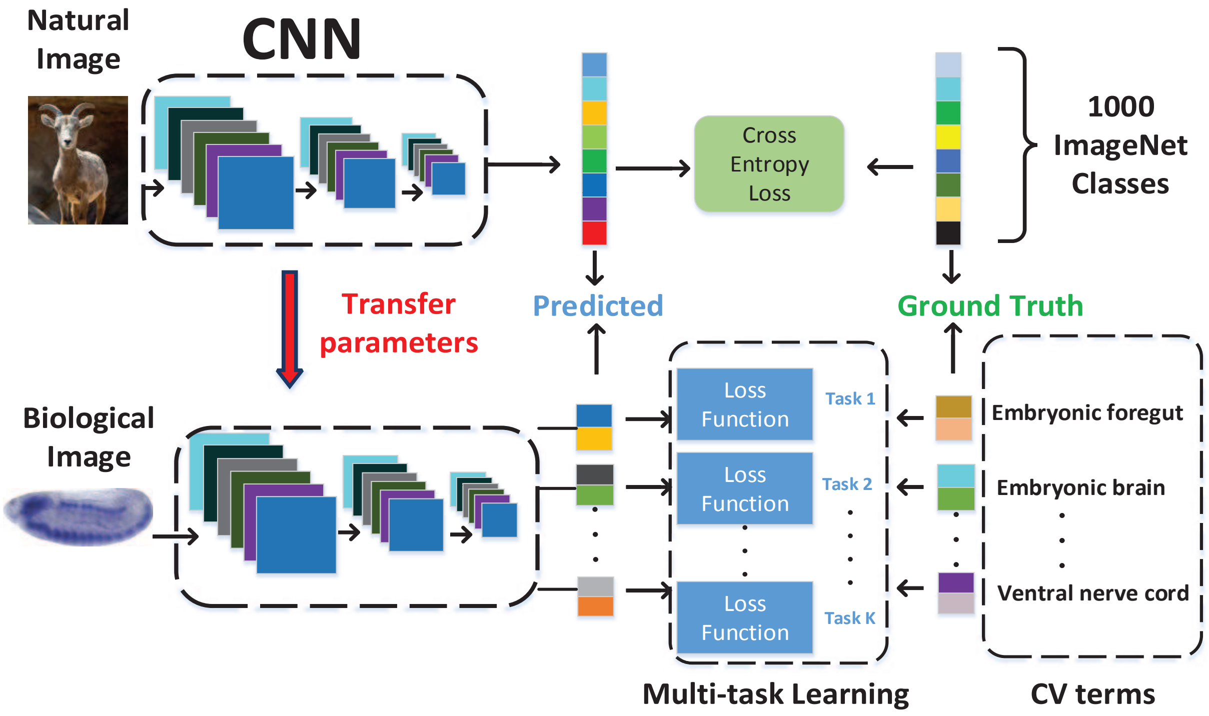


Figure 1: Pipeline of deep models for transfer learning and multi-task learning. The network was trained on the ImageNet data containing millions of labeled natural images with thousands of categories (top row). The pre-trained parameters are then transferred to the target domain of biological images. We first directly used the pre-trained model to extract features from *Drosophila* gene expression pattern images. We then fine-tuned the trained model with labeled ISH images. We then employed the fine-tuned model to extract features to capture CV term-specific discriminative information (bottom row).

we used above. To be specific, the pre-trained model is designed to recognize objects in natural images while we studied the CV term annotation of *Drosophila* images instead. Although the leveraged knowledge from the source task could reflect some common characteristics shared in these two types of images such as corners or edges, extra efforts are also needed to capture the specific properties of ISH images. The *Drosophila* gene expression pattern images are organized into groups, and multiple CV term annotations are assigned to multiple images in the same group. This multi-image multi-label nature poses significant challenges to traditional image annotation methodologies. This is partially due to the fact that there are ambiguous multiple-to-multiple relationships between images and CV term annotations, since each group of images are associated with multiple CV term annotations.

In this paper, we proposed to use multi-task learning strategy to overcome the above difficulty. To be specific, we first employed a CNN model that is pre-trained on natural images to initialize the parameters of a deep network. Then, we fine-tuned this network using multiple annotation term prediction tasks to obtain CV term-specific discriminative representation. The pipeline of our method is illustrated in Figure 1. We have a single pre-trained network with the same inputs but with multiple outputs, each of which corresponds to a term annotation task. These outputs are fully connected to a hidden layer that they share. Because all outputs share a common layer, the internal representations learned by one task could be used by other tasks. Note that the back-propagation is done in parallel on these outputs in the network. For each task, we used its individual loss function to measure the difference between outputs and the ground truth. In particular, we are given a training set of k tasks $\{X_i, y_i^j\}_{i=1}^m$, $j = 1, 2, \ldots, k$, where $X_i \in R^n$ denotes the i-th training sample, m denotes the total number of training samples. Note that we used the same groups of samples for different tasks, which is a simplified version of traditional multi-task learning. The output label y_i^j denotes the CV term annotation status of training sample, which is binary with the form

$$y_i^j = \begin{cases} 1 & \text{if } X_i \text{ is annotated with the } j\text{-th CV term,} \\ 0 & \text{otherwise.} \end{cases}$$

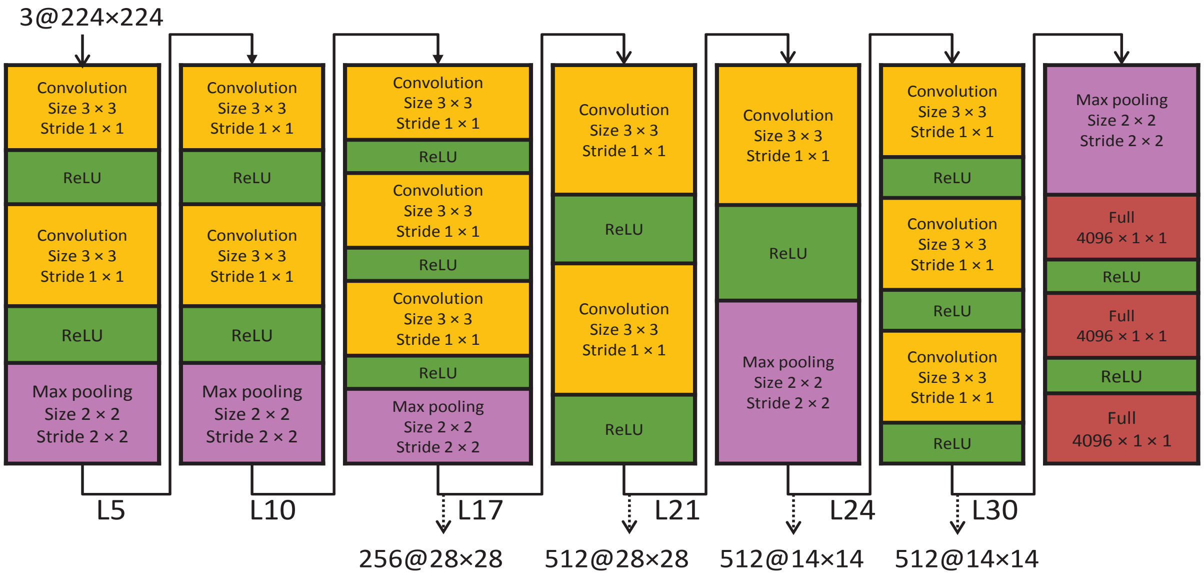
To quantitatively measure the difference between the predicted annotation results and ground truth from human experts, we used a loss function in the following form:

$$loss(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{m}\sum_{j=1}^{k}\left(y_i^j log f(\hat{y}_i^j) + (1 - y_i^j) log(1 - f(\hat{y}_i^j))\right),$$

where

$$f(q) = \begin{cases} \frac{1}{1+e^{-q}} & \text{if} \quad q \geq 0 \\ 1 - \frac{1}{1+e^{-q}} & \text{if} \quad q < 0, \end{cases}$$

and $\mathbf{y} = \{y_i^j\}_{i,j=1}^{m,k}$ denotes the ground truth label matrix over different tasks, and $\hat{\mathbf{y}} = \{y_i^j\}_{i,j=1}^{m,k}$ is the output matrix


3@224×224
Convolution Size 3 × 3 Stride 1 × 1
ReLU
Convolution Size 3 × 3 Stride 1 × 1
ReLU
Max pooling Size 2 × 2 Stride 2 × 2
Convolution Size 3 × 3 Stride 1 × 1
ReLU
Convolution Size 3 × 3 Stride 1 × 1
ReLU
Max pooling Size 2 × 2 Stride 2 × 2
Convolution Size 3 × 3 Stride 1 × 1
ReLU
Convolution Size 3 × 3 Stride 1 × 1
ReLU
Convolution Size 3 × 3 Stride 1 × 1
ReLU
Max pooling Size 2 × 2 Stride 2 × 2
Convolution Size 3 × 3 Stride 1 × 1
ReLU
Convolution Size 3 × 3 Stride 1 × 1
ReLU
Convolution Size 3 × 3 Stride 1 × 1
ReLU
Max pooling Size 2 × 2 Stride 2 × 2
Convolution Size 3 × 3 Stride 1 × 1
ReLU
Convolution Size 3 × 3 Stride 1 × 1
ReLU
Convolution Size 3 × 3 Stride 1 × 1
ReLU
Max pooling Size 2 × 2 Stride 2 × 2
Full 4096 × 1 × 1
ReLU
Full 4096 × 1 × 1
ReLU
Full 4096 × 1 × 1
L5
L10
L17
L21
L24
L30
256@28×28
512@28×28
512@14×14
512@14×14

Input size	L17	L21	L24	L30
224×224	200704	401408	100352	100352

Figure 2: Detailed architecture of the VGG model. "Convolution", "Max pooling" and "ReLU" denote convolutional layer, max pooling layer and rectified linear unit function layer, respectively. This model consists of 36 layers. We extracted features from layers 17, 21, 24, and 30.

of our network through feedforward propagation. Note that $\hat{y}_i^j$ denotes the network output before the softmax activation function. This loss function is a special case of the cross entropy loss function by using sigmoid function to induce probability representation [2, 3]. Note that our multi-task loss function is the summation of multiple loss functions, and all of them are optimized simultaneously during training.

4. BIOLOGICAL IMAGE ANALYSIS

The *Drosophila melanogaster* has been widely used as a model organism for the study of genetics and developmental biology. To determine the gene expression patterns during *Drosophila* embryogenesis, the Berkeley *Drosophila* Genome Project (BDGP) used high throughput RNA *in situ* hybridization (ISH) to generate a systematic gene expression image database [20, 21]. In BDGP, each image captures the gene expression patterns of a single gene in an embryo. Each gene expression image is annotated with a collection of anatomical and developmental ontology terms using a CV term annotation to identify the characteristic structures in embryogenesis. This annotation work is now mainly carried out manually by human experts, which makes the whole process time-consuming and costly. In addition, the number of available images is now increasing rapidly. Therefore, it is desirable to design an automatic and systematic annotation approach to increase the efficiency and accelerate biological discovery [5, 8, 11, 12, 15, 16].

Prior studies have employed machine learning and computer vision techniques to automate this task. Due to the effects of stochastic process in development, every embryo develops differently. In addition, the shape and position of the same embryonic part may vary from image to image. Thus, how to handle local distortions on the images is crucial for building robust annotation methods. The seminal work in [28] employed the wavelet-embryo features by using the wavelet transformation to project the original pixel-based embryonic images onto a new feature domain. In subsequent work, local patches were first extracted from an image and local features which are invariant to certain geometric transformations (e.g., scaling and translation) were then computed from each patch. Each image was then represented as a bag of "visual words", known as the "bag-of-words" representation [7], or a set of "sparse codes", known as the "sparse coding" representation [19, 25]. All prior methods used handcrafted local features combined with high-level methods, such as the bag-of-words or sparse coding schemes, to obtain image representations. These methods can be viewed as two-layer feature extractors. In this work, we proposed to employ the deep CNNs as a multi-layer feature extractor to generate image representations for CV term annotation.

We showed here that a universal feature extractor trained on problem-independent data set can be used to compute feature representations for CV term annotation. Furthermore, the model trained on problem-independent data set, such as the ImageNet data, can be fine-tuned on labeled data from specific domains using the error back propagation algorithm. This will ensure that the knowledge transferred from problem-independent images is adapted and tuned to capture domain-specific features in biological images. Since generating manually annotated biological images is both time-consuming and costly, the transfer of knowledge from other domains, such as the natural image world, is essential in achieving competitive performance.

Table 1: Statistics of the data set used in this work. The table shows the total number of images for each stage range and the numbers of positive samples for each term.

Stages	Number	# of positive samples for each term									
	of images	No. 1	No. 2	No. 3	No. 4	No. 5	No. 6	No. 7	No. 8	No. 9	No. 10
4-6	4173	953	438	1631	1270	1383	1351	351	568	582	500
7-8	1953	782	741	748	723	753	668	510	340	165	209
9-10	2153	899	787	778	744	694	496	559	452	350	264
11-12	7441	2945	2721	2056	1932	1847	1741	1400	1129	767	1152
13-17	7564	2572	2169	2062	1753	1840	1699	1273	1261	891	1061

5. EXPERIMENTS

5.1 Experimental setup

In this study, we used the *Drosophila* ISH gene expression pattern images provided by the FlyExpress database [12, 22], which contains genome-wide, standardized images from multiple sources, including the Berkeley *Drosophila* Genome Project (BDGP). For each *Drosophila* embryo, a set of high-resolution, two-dimensional image series were taken from different views (lateral, dorsal, and lateral-dorsal and other intermediate views). These images were then subsequently standardized semi-manually. In this study, we focused on the lateral-view images only, since most of images in FlyExpress are in lateral view.

In the FlyExpress database, the embryogenesis of *Drosophila* has been divided into six discrete stage ranges (stages 1-3, 4-6, 7-8, 9-10, 11-12, and 13-17). We used those images in the later 5 stage ranges in the CV term annotation, since only a very small number of keywords were used in the first stage range. One characteristic of these images is that a group of images from the same stage and same gene are assigned with the same set of keywords. Prior work in [19] has shown that image-level annotation outperformed group-level annotation using the BDGP images. In this work, we focused on the image-level annotation only and used the same top 10 keywords that are most frequently annotated for each stage range as in [19]. The statistics of the numbers of images and most frequent 10 annotation terms for each stage range are given in Table 1.

For CV term annotation, our image data set is highly imbalanced with much more negative samples than positive ones. For example, there are 7564 images in stages 13-17, but only 891 of them are annotated the term "dorsal prothoracic pharyngeal muscle". The commonly-used classification algorithms might not work well for our specific problem, because they usually aimed to minimizing the overall error rate without paying special attention to the positive class. Prior work in [19] has shown that using under-sampling with ensemble learning could produce better prediction performance. In particular, we selectively under-sampled the majority class to obtain the same number of samples as the minority class and built a model for each sampling. This process was performed many times for each keyword to obtain a robust prediction. Following [19], we employed classifier ensembles built on biased samples to train robust models for annotation. In order to further improve the performance, we produced the final prediction by using majority voting, since this sample scheme is one of the widely used methods for fusion of multiple classifiers. For comparison purpose, we also implemented the existing sparse coding image representation method studied in [19]. The annotation performance was measured using accuracy, specificity, sensitivity and area under the ROC curve (AUC) for CV term annotation. For all of these measures, a higher value indicates better annotation performance. All classifiers used in this work are the ℓ_2-norm regularized logistic regression.

5.2 Comparison of features extracted from different layers

The deep learning model consists of multiple layer of feature maps for representing the input images. With this hierarchical representation, a natural question is which layer has the most discriminative power to capture the characteristics of input images. When such networks were trained on natural image data set such as the ImageNet data, the features computed in lower layers usually correspond to local features of objects such as edges, corners or edge/color conjunctions. In contrast, the features encoded at higher layers mainly represent class-specific information of the training data. Therefore, for the task of natural object recognition, the features extracted from higher layers usually yielded better discriminative power [26].

In order to identify the most discriminative features for the gene expression pattern annotation tasks, we compared the features extracted from various layers of the VGG network. Specifically, we used the ISH images as inputs to the pre-trained VGG network and extracted features from layers 17, 21, 24, and 30 for each ISH image. These features were used for the annotation tasks, and the results are given in Figure 3. We can observe that for all stage ranges, layer 21 features outperformed other features in terms of overall performance. Specifically, the discriminative power increases from layer 17 to layer 21, and then drops afterwards as the depth of network increases. This indicates that gene expression features are best represented in the intermediate layers of CNN that was trained on natural image data set. One reasonable explanation about this observation is the lower layers compute very primitive image features that are not enough to capture gene expression patterns. Meanwhile, the higher layers captured features that are specific to the training natural image set, and these features may not be relevant for gene expression pattern images.

Then we proposed to use multi-task learning strategy to fine-tune the pre-trained network with labeled ISH images. In order to show the gains through fine-tuning on pre-trained model, we extracted features from the same hidden layers that are used for the pre-trained model. We reported the predictive performance achieved by features of different layers in the proposed fine-tuned model in Figure 4. It can be observed from the results that the predictive performance

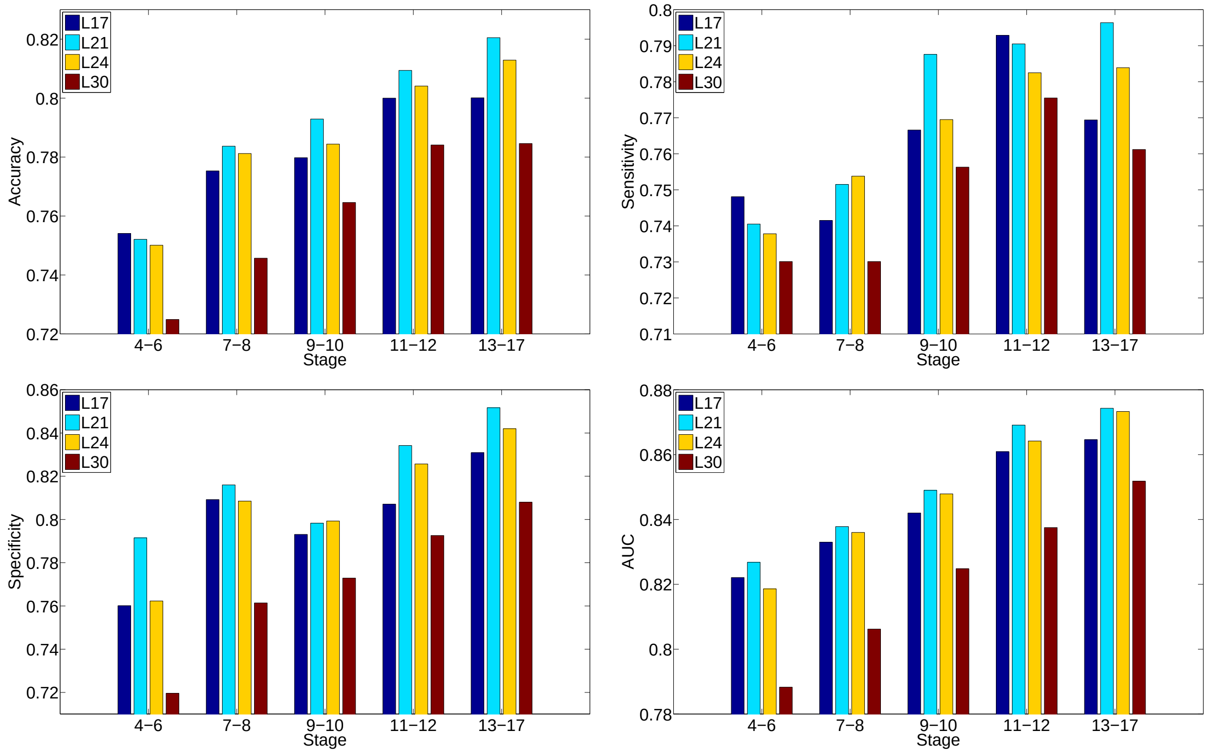


Figure 3: Comparison of annotation performance achieved by features extracted from different layers of deep models for transfer learning over five stage ranges. "Lx" denotes the hidden layer from which the features were extracted.

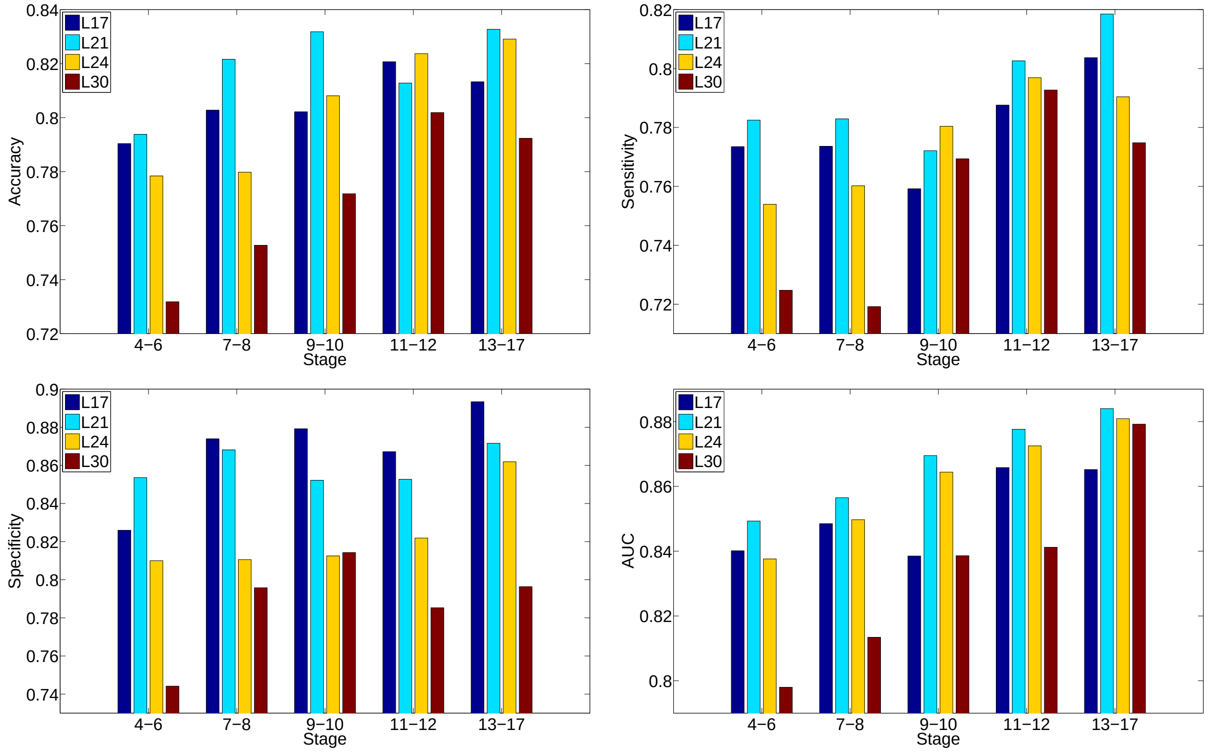


Figure 4: Comparison of annotation performance achieved by features extracted from different layers of the deep models for multi-task learning over five stage ranges. "Lx" denotes the hidden layer from which the features were extracted.

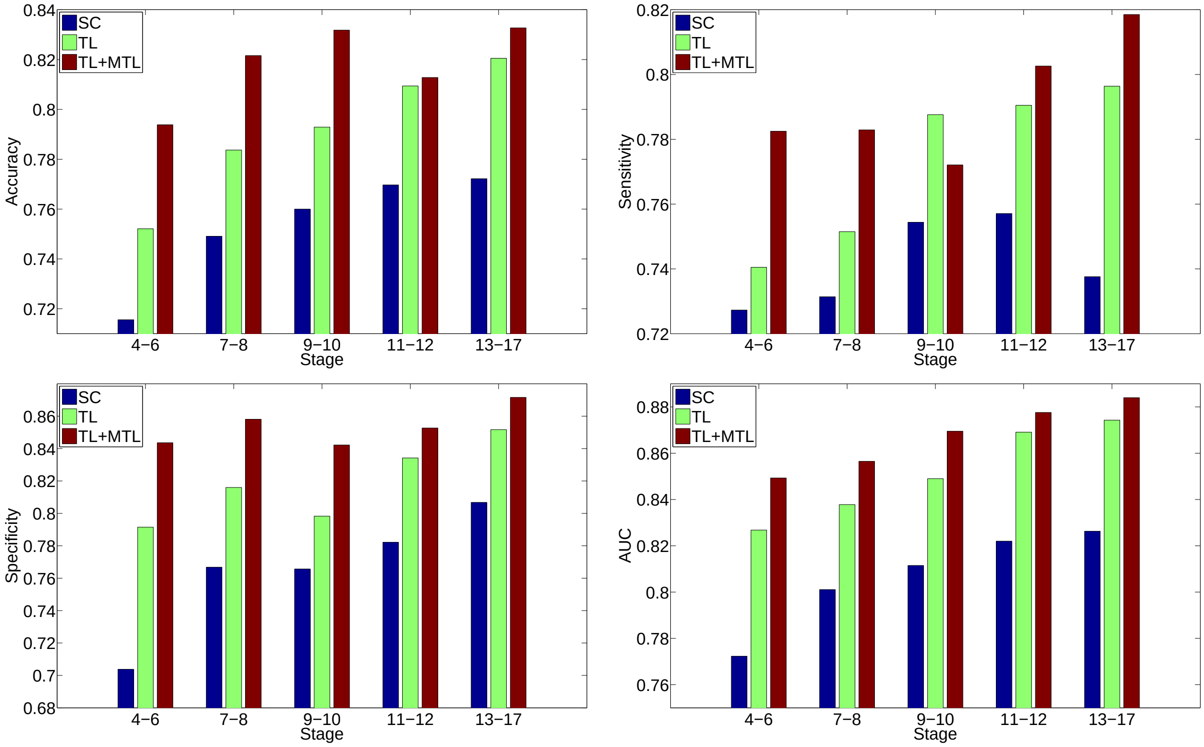


Figure 5: Performance comparison of different methods. "SC" denotes sparse coding. "TL" and "TL + MTL" denote the performance achieved by transfer learning and multi-task learning models, respectively. We only consider the features extracted from layer 21 of these two deep models.

was generally higher on middle layers in the deep architecture. In particular, layer 21 outperformed other layers significantly. This result is consistent with the observation found on the pre-trained model.

5.3 Comparison with prior methods

We also compared the performance achieved by different methods including sparse coding, transfer learning model and multi-task learning. These results demonstrated that our deep model with multi-task learning were able to accurately annotate gene expression images over all embryogenesis stage ranges. To compare our generic features with the domain-specific features used in [19], we compared the annotation performance of our deep learning features with that achieved by the domain-specific sparse coding features. Deep learning models include transfer learning and multi-task learning. In this experiment, we only considered the features extracted from layer 21 since they yielded the best performance among different layers. The performance of these three types of features averaged over all terms is given in Figure 5 and Table 2. We can observe that the deep model for multi-task learning features outperformed the sparse coding features and transfer learning features consistently and significantly in all cases. To examine the performance differences on individual anatomical terms, we showed the AUC values on each term in Figure 6 for different stage ranges. We can observe that our features extracted from layer 21 of the VGG networks for transfer learning and multi-task learning outperformed the sparse coding features over all stage ranges for all terms consistently. These results demonstrated that our generic features of deep models were better at representing gene expression pattern images than the problem-specific features based on sparse coding.

In Figure 7, we provided a term-by-term and image-by-image comparison between the results of the deep model for multi-task learning and the sparse coding features for the 10 terms in stages 13-17. The x-axis corresponds to the 10 terms. The y-axis corresponds to a subset of 50 images in stages 13-17 with the largest numbers of annotated terms. Overall, it is clear that the total number of green and blue entries is much more than the number of red and pink entries, indicating that, among all predictions disagreed by these two methods, the predictions by the multi-task learning features were correct most of the time.

6. CONCLUSIONS AND FUTURE WORK

In this work, we proposed to employ the deep convolutional neural networks as a multi-layer feature extractor to generate generic representations for ISH images. We used the deep convolutional neural network trained on large natural image set as feature extractors for ISH images. We first directly used the model trained on natural images as feature extractors. We then employed multi-task classification methods to fine-tune the pre-trained model with labeled ISH images. Although the number of annotated ISH images is small, it nevertheless improved the pre-trained model. We compared the performance of our generic approach with the problem-specific methods. Results showed that our proposed approach significantly outperformed prior methods on ISH image annotation. We also showed that the intermediate layers of deep models produced the best gene expression pattern representations.

Table 2: Performance comparison in terms of accuracy, sensitivity, specificity, and AUC achieved by CNN models and Sparse Coding features for all stage ranges. "TL+MTL" and "TL" denote the features extracted from layer 21 of the deep model for multi-task learning and transfer learning. "SC" denotes the performance of the sparse coding features.

Measures	Methods	Stage 4-6	Stage 7-8	Stage 9-10	Stage 11-12	Stage 13-17
Accuracy	TL+MTL	0.7938±0.0381	0.8216±0.0231	0.8318±0.0216	0.8128±0.0325	0.8327±0.0256
	TL	0.7521±0.0326	0.7837±0.0269	0.7929±0.0231	0.8094±0.0331	0.8205±0.0304
	SC	0.7217±0.0352	0.7401±0.0351	0.7549±0.0303	0.7659±0.0326	0.7681±0.0231
Sensitivity	TL+MTL	0.7825±0.0372	0.7829±0.0368	0.7721±0.0412	0.8026±0.0401	0.8185±0.0259
	TL	0.7405±0.0293	0.7515±0.0342	0.7876±0.0401	0.7905±0.0389	0.7964±0.0317
	SC	0.7321±0.0408	0.7190±0.0331	0.7468±0.0298	0.7576±0.0329	0.7328±0.0235
Specificity	TL + MTL	0.8436±0.0376	0.8581±0.0380	0.8422±0.0284	0.8527±0.0252	0.8716±0.0256
	TL	0.7915±0.0247	0.8160±0.0316	0.7983±0.0315	0.8342±0.0237	0.8517±0.0306
	SC	0.7140±0.0389	0.7605±0.0392	0.7629±0.0298	0.7749±0.0329	0.8005±0.0298
AUC	TL + MTL	0.8493±0.0427	0.8565±0.0279	0.8695±0.0276	0.8776±0.0291	0.8824±0.0197
	TL	0.8344±0.0439	0.8401±0.0346	0.8508±0.0257	0.8702±0.0271	0.8746±0.0299
	SC	0.7687±0.0432	0.7834±0.0358	0.7921±0.0294	0.8061±0.0342	0.8105±0.0280

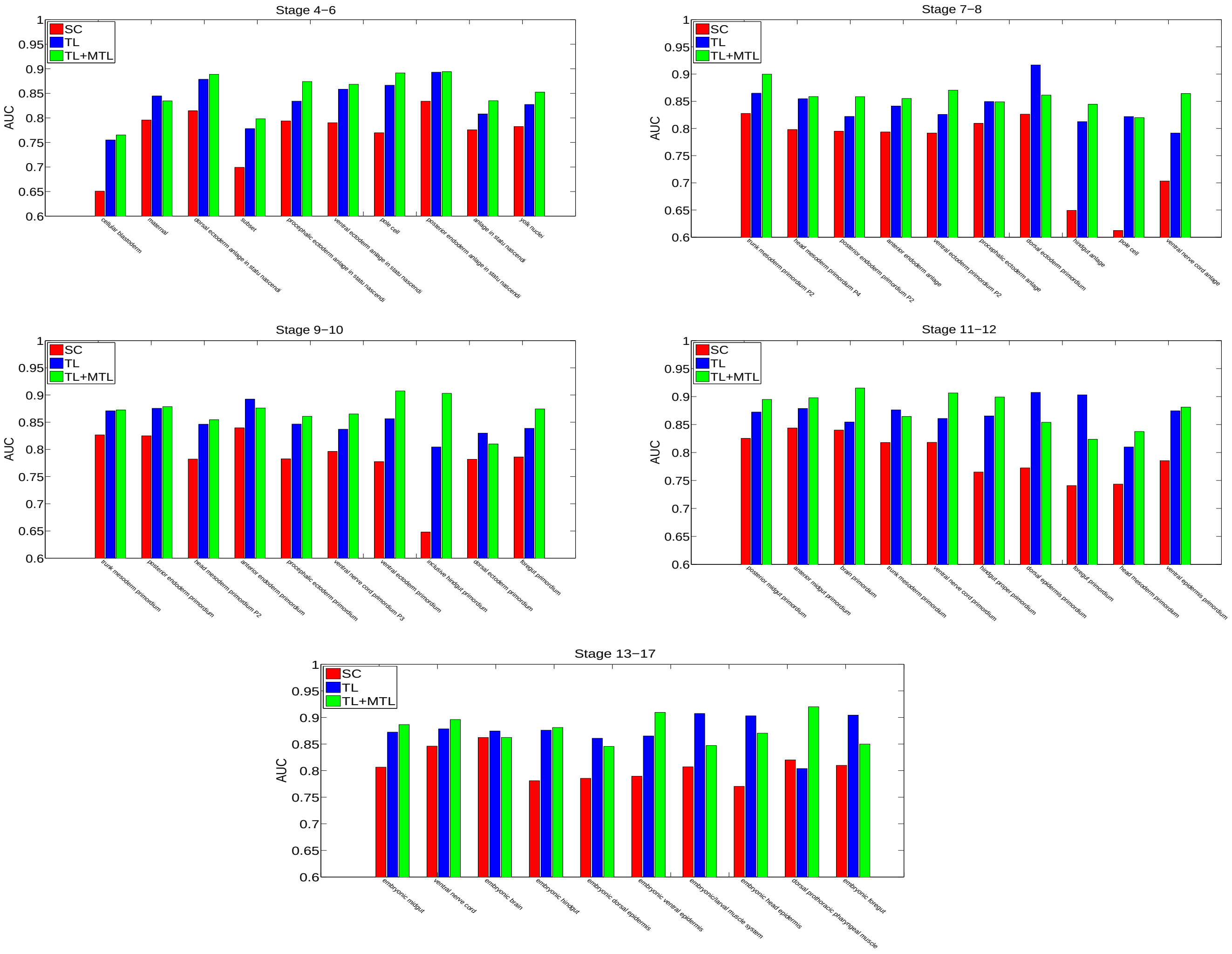


Figure 6: Performance comparison of different methods for all stage ranges. "SC", "TL" and "TL + MTL" denote sparse coding, transfer learning and multi-task learning models, respectively.

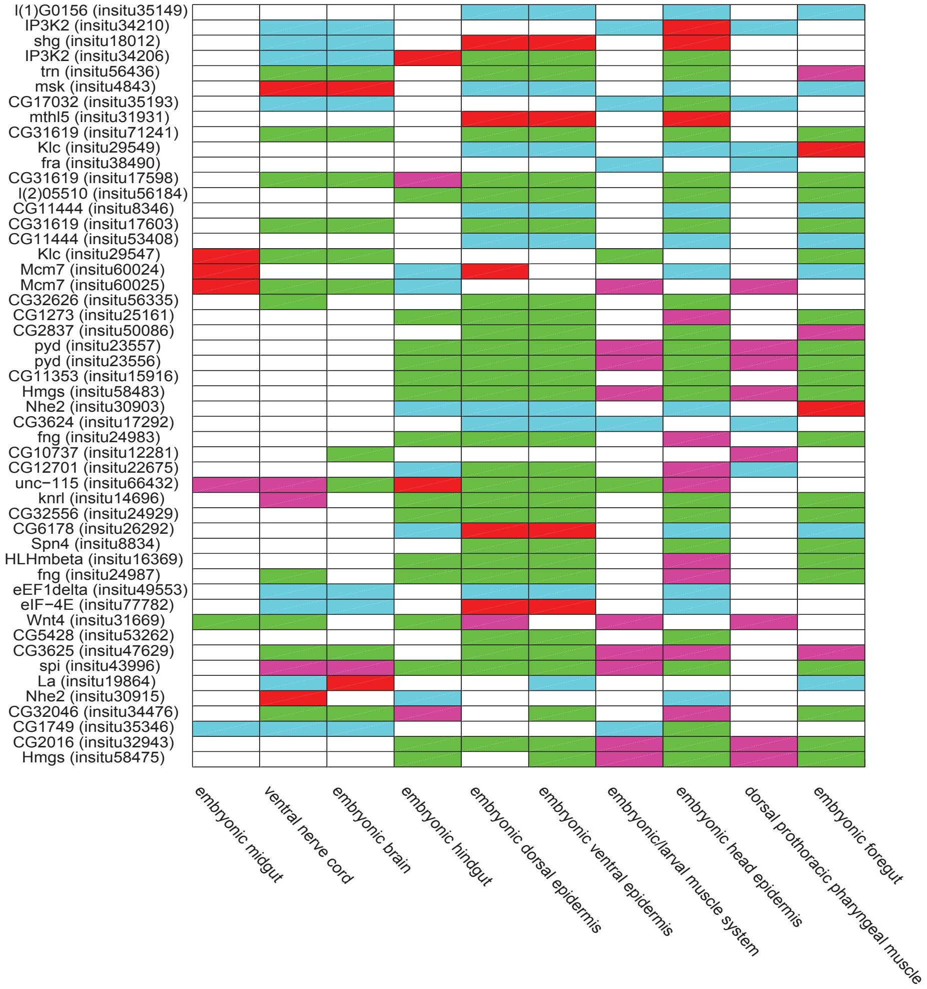


Figure 7: Comparison of prediction results between the deep models for multi-task learning and the sparse coding features for the 10 terms in stages 13-17. The x-axis shows the 10 terms. The y-axis corresponds to a subset of 50 images in stages 13-17 with the largest numbers of annotated terms. The gene names and the FlyExpress image IDs in parentheses are displayed. The prediction results of different methods compared with the ground truth are distinguished by different colors. The white entries correspond to predictions agreed upon by these two methods, while non-white entries were used to denote different types of disagreements. Specifically, the green and blue entries correspond to correct predictions by the multi-task learning features but incorrect predictions by the sparse coding features. Green and blue indicate positive and negative samples, respectively, in the ground truth. Similarly, the red and pink entries correspond to incorrect predictions by the multi-task learning features but correct predictions by the sparse coding features. Red and pink indicate positive and negative samples, respectively, in the ground truth.

In the current study, we focus on using deep models for CV annotation. There are many other biological image analysis tasks that require appropriate image representations such as developmental stage prediction. We will consider broader applications in the future. In this work, we considered a simplified version of the problem in which each term is associated with all images in the same group. We will extend our model to incorporate the image group information in the future.

7. ACKNOWLEDGMENTS

This work was supported in part by research grants from NSF (IIS-0953662, DBI-1147134 and DBI-1350258), NIH (R01 LM010730, HG002516-09), and the NVIDIA Corporation with the donation of the Tesla K40 GPU.

8. REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[2] C. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[3] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th*

international conference on Machine learning, pages 160–167. ACM, 2008.

[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the 31st International Conference on Machine Learning*, pages 647–655, 2014.

[5] E. Frise, A. S. Hammonds, and S. E. Celniker. Systematic image-driven analysis of the spatial *Drosophila* embryonic expression landscape. *Molecular Systems Biology*, 6:345, 2010.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[7] S. Ji, Y.-X. Li, Z.-H. Zhou, S. Kumar, and J. Ye. A bag-of-words approach for *Drosophila* gene expression pattern annotation. *BMC Bioinformatics*, 10(1):119, 2009.

[8] S. Ji, L. Sun, R. Jin, S. Kumar, and J. Ye. Automated annotation of *Drosophila* gene expression patterns using a controlled vocabulary. *Bioinformatics*, 24(17):1881–1888, 2008.

[9] S. Ji, L. Yuan, Y.-X. Li, Z.-H. Zhou, S. Kumar, and J. Ye. *Drosophila* gene expression pattern annotation using sparse features and term-term interactions. In *Proceedings of the Fifteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 407–416, 2009.

[10] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1106–1114. 2012.

[11] S. Kumar, K. Jayaraman, S. Panchanathan, R. Gurunathan, A. Marti-Subirana, and S. J. Newfeld. BEST: a novel computational approach for comparing gene expression patterns from early stages of *Drosophila melanogaster* develeopment. *Genetics*, 169:2037–2047, 2002.

[12] S. Kumar, C. Konikoff, B. Van Emden, C. Busick, K. T. Davis, S. Ji, L.-W. Wu, H. Ramos, T. Brody, S. Panchanathan, J. Ye, T. L. Karr, K. Gerold, M. McCutchan, and S. J. Newfeld. FlyExpress: visual mining of spatiotemporal patterns for genes and publications in *Drosophila* embryogenesis. *Bioinformatics*, 27(23):3319–3320, 2011.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[14] M. Oquab, I. Laptev, L. Bottou, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[15] I. Pruteanu-Malinici, D. L. Mace, and U. Ohler. Automatic annotation of spatial expression patterns via sparse Bayesian factor models. *PLoS Comput Biol*, 7(7):e1002098, 07 2011.

[16] K. Puniyani, C. Faloutsos, and E. P. Xing. SPEX2: automated concise extraction of spatial gene expression patterns from fly embryo ISH images. *Bioinformatics*, 26(12):i47–56, 2010.

[17] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014.

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[19] Q. Sun, S. Muckatira, L. Yuan, S. Ji, S. Newfeld, S. Kumar, and J. Ye. Image-level and group-level models for *Drosophila* gene expression pattern annotation. *BMC Bioinformatics*, 14:350, 2013.

[20] P. Tomancak, A. Beaton, R. Weiszmann, E. Kwan, S. Shu, S. E. Lewis, S. Richards, M. Ashburner, V. Hartenstein, S. E. Celniker, and G. M. Rubin. Systematic determination of patterns of gene expression during *Drosophila* embryogenesis. *Genome Biology*, 3(12):research0088.1–0088.14, 2002.

[21] P. Tomancak, B. Berman, A. Beaton, R. Weiszmann, E. Kwan, V. Hartenstein, S. Celniker, and G. Rubin. Global analysis of patterns of gene expression during *Drosophila* embryogenesis. *Genome Biology*, 8(7):R145, 2007.

[22] B. Van Emden, H. Ramos, S. Panchanathan, S. Newfeld, and S. Kumar. Flyexpress: an image-matching web-tool for finding genes with overlapping patterns of expression in drosophila embryos. *Tempe, AZ*, 85287530, 2006.

[23] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pages 3320–3328, 2014.

[24] L. Yuan, C. Pan, S. Ji, M. McCutchan, Z.-H. Zhou, S. Newfeld, S. Kumar, and J. Ye. Automated annotation of developmental stages of *Drosophila* embryos in images containing spatial patterns of expression. *Bioinformatics*, 30(2):266–273, 2014.

[25] L. Yuan, A. Woodard, S. Ji, Y. Jiang, Z.-H. Zhou, S. Kumar, and J. Ye. Learning sparse representations for fruit-fly gene expression pattern image annotation and retrieval. *BMC Bioinformatics*, 13:107, 2012.

[26] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Proceedings of the European Conference on Computer Vision*, pages 818–833. Springer, 2014.

[27] W. Zhang, D. Feng, R. Li, A. Chernikov, N. Chrisochoides, C. Osgood, C. Konikoff, S. Newfeld, S. Kumar, and S. Ji. A mesh generation and machine learning framework for *Drosophila* gene expression pattern image analysis. *BMC Bioinformatics*, 14:372, 2013.

[28] J. Zhou and H. Peng. Automatic recognition and annotation of gene expression patterns of fly embryos. *Bioinformatics*, 23(5):589–596, 2007.