Focus Longer to See Better: Recursively Refined Attention for Fine-Grained Image Classification

Prateek Shroff[1], Tianlong Chen[1], Yunchao Wei[2], Zhangyang Wang[1]
[1]Texas A&M University,[2]University of Technology Sydney
{prateek.shroff,wiwjp619,atlaswang}@tamu.edu,wychao1987@gmail.com

Abstract

Deep Neural Network has shown great strides in the coarse-grained image classification task. It was in part due to its strong ability to extract discriminative feature representations from the images. However, the marginal visual difference between different classes in fine-grained images makes this very task harder. In this paper, we tried to focus on these marginal differences to extract more representative features. Similar to human vision, our network repetitively focuses on parts of images to spot small discriminative parts among the classes. Moreover, we show through interpretability techniques how our network focus changes from coarse to fine details. Through our experiments, we also show that a simple attention model can aggregate (weighted) these finer details to focus on the most dominant discriminative part of the image. Our network uses only image-level labels and does not need bounding box/part annotation information. Further, the simplicity of our network makes it an easy plug-n-play module. Apart from providing interpretability, our network boosts the performance (up to 2%) when compared to its baseline counterparts. Our codebase is available at https://github.com/TAMU-VITA/Focus-Longer-to-See-Better

1. Introduction

Fine-grained image classification [30] has been an active research area that recognizes sub-categories within some meta-category [33, 14, 16]. This problem differs from generic image classification due to very small inter-class and large intra-class variations among the classes. This makes the task very challenging, as the recognition should be able to localize these fine variations and then represents these marginal visual differences. Though deep neural networks [30] have shown astounding performance in generic image classification [27], reaching similar-level performance on fine-grained recognition remains a challenge.

Deep learning approaches for fine-grained classification

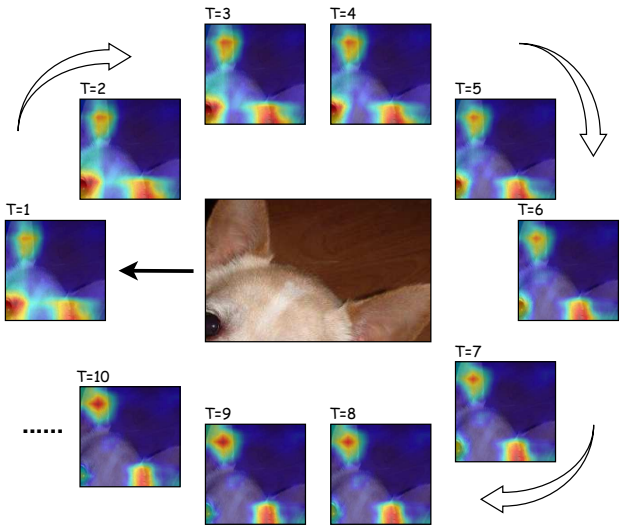

Figure 1. Example of a center patch of an image. The heat-maps around the image visualize the changes in attention, as we look longer at an image from temporal step (T) from 1 to 10. Looking recurrently at an image helps our model to progressively spot finer details like pointy ears from the coarser head region.

fall into two separate paradigms: localization-classification network [8, 38, 10], and end-to-end feature encoding [21, 5, 1]. The first category makes use of a separate localization network along with the classification network. The localization network is used to localize the discriminative image regions/parts. In order to localize these fine changes, earlier work [34, 22] has relied on human-annotated bounding box/par annotations (eg head, wings, feather color). But, all these human-based manual annotations make the process quite intensive, laborious, and subjective. Also, the manual annotation may be possible for small scale datasets like CUB200-2011 [33] , Stanford dataset [14, 16] but not feasible for large-scale image dataset say ImageNet [27]. Convolution neural networks (CNNs) were hence leveraged for weakly supervised part-learning with category-labels only, assuming no dependencies on bounding box/part annotations [6, 7, 17].

In the localization-classification network, the localization subnetwork focuses on learning the objects parts shared among the same classes while the classification subnetwork extracts discriminative features from these localize objects to make them different among classes. This complementary network architecture requires separate losses [34, 8] and tends to be computationally expensive.

The second category is to encode higher-order statistics of convolutional feature maps to enhance the feature presentation of the image [21, 9, 15, 18]. One of the first works in this category was the use of Bilinear CNNs [21] which computes pairwise feature interactions by two independent CNNs to capture the local differences in the image. Another work [26] proposed to encode CNN representation with Fisher Vector representation giving much superior performance on several datasets. But using higher-order dynamics makes the network less human-interpretable when compared to the localization-classification sub-network.

To overcome the above-mentioned challenges, we propose a novel attention-based recurrent convolutional neural network for fine-grained image classification. Our network recursively attends from coarse to the finer region of image or parts of the image to focus on the discriminative region more finely. Our model is simple, computationally inexpensive, and interpretable. Our motivation is that by processing an image or a part of the image recursively, we can focus on most discriminative details by continuously removing insignificant ones and other background noises. Further, by aggregating the finer regions from the image via suitable attention we can pinpoint the most discriminative region in the image. Additionally, the module is plug-and-play which greatly enhances its scalability and usability.

Our network consists of a weakly supervised patch extraction network which extracts different patches corresponding to an image. Another network attends to each patch by recurrently processing it via LSTMs. We use unidirectional stacked LSTMs to recurrently pass the patch through the time steps of LSTMs. Then, an attention layer is used to aggregate the finer representation from the output of the LSTMs. We append this network to the baseline image classifier giving way to a two-stream architecture. To leverage the power of ensembles, the representative features are fused and then passed to the end classifier.

Our contributions can be summarized as the following:

- We propose a novel recurrent attention network which progressively attends to and aggregate finer image details, for more discriminative representations.
- We show through various ablation studies the human interpretability of our attentions and features.
- We conduct experiments on two challenging benchmarks (CUB200-2011 birds [33], Stanford Dogs [14]), and show performance boosts over the baselines.

2. Related Work

Fine-grained Feature Learning. Learning discriminative features have been studied extensively in the field of image recognition and also for fine-grained classification. Due to the great success of deep learning, powerful deep convolutional based features [30, 11, 12, 31] forms the backbone for most of the recognition tasks. This has shown a great boost in performance when compared to hand-crafted features. To model subtle difference, a bilinear structure [21] is used to compute pairwise differences. The use of boosting to combine the representation of multiple learners also helps to improve classification accuracy [25]. Additionally, second-order information also helps in fine-grained feature extraction. Pooling methods that utilize second-order information [19, 18] have proven to enhance the extraction of more meaningful information.

Interpretable Deep Models for Fine-grained Recognition. Given the subtle differences between fine-grained categories, it becomes imperative to focus on and extract meaningful features from them. There has been extensive research [39, 13, 35, 29, 4] to develop interpretable models that visualize regions attended by the network. In [39] , Class Activation Maps (CAMs) are used to provide object-level attention but not providing much finer discriminative details. Over time, there have been variants developed [28, 23], that explores the backward propagation to identify salient image features. In [13, 35, 29], attention is at a finer level and focus more on the parts of the object rather than the whole body/object. In [4], the authors associate the prototypical aspect with the object part to reason out the classification prediction for an image. Our network uses a simple approach based on [28] to visualize the fine attention areas in the patches.

Attention. Attention has been incorporated in visual related tasks from a long time [36, 24, 2, 37, 3]. Attention models are aimed at identifying discriminative image parts that are most responsible for recognition. We follow on the same methodology of the visual-attention model to aggregate the output of LSTMs to have weighted attention to the most discriminative patch/part of the image. In [10], the author uses weakly supervised model to generate different patches of the same image containing different parts of images. We used a similar approach to extract patches from the images which is further used to look for finer details. This method does not use any external information like part annotations/bounding box information.

3. Our Proposal

Given an image I and its corresponding label c, our network aims to look longer via recurrently iterating through a patch of an image to extract more fine-grained information. A bottom-up weakly supervised object detection ap-

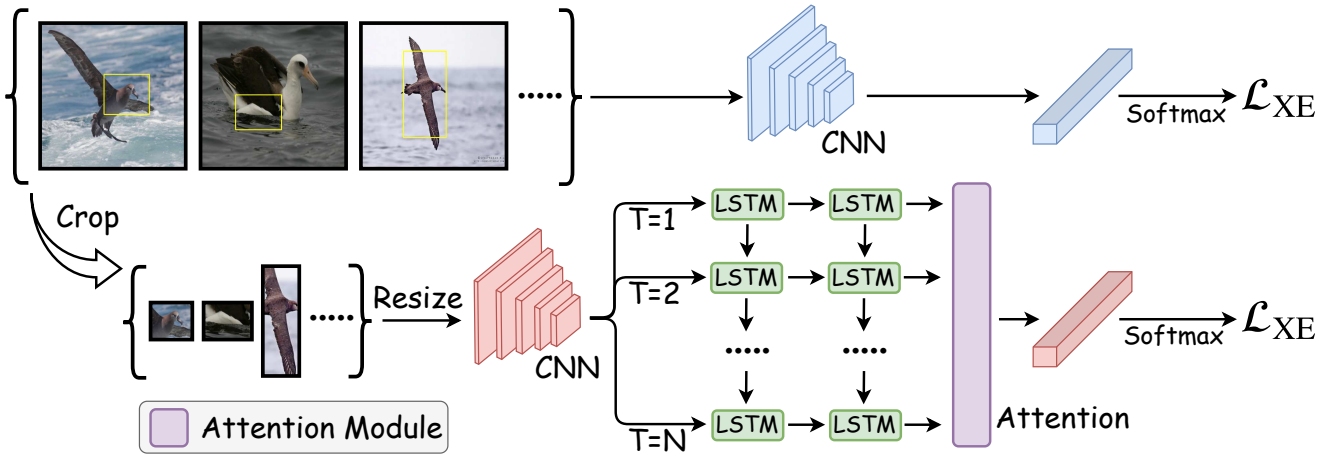

Figure 2. The pipeline of our two-stream architecture. The global stream (on the top) processes the entire image to provide global representation. While the local stream (on the bottom) processes a certain region of the image (say, patch). The features are generated by recurrently passing the patch through stacked LSTMs followed by an attention layer. Finally, the whole architecture is optimized via cross-entropy loss at each output.

proach is used to extract meaningful patches (parts of the images). This network uses only the category level labels and does not use any part annotations or bounding box information. Further, a two-stream feature extractor is used to extract global and object-level feature representations to boost the classification accuracy.

3.1. Two-Stream Architecture

Once we get a set of patches for each training image $\mathcal{I}$, we randomly select a patch $\mathcal{P}_\mathrm{i}$ from the set of patches $\mathcal{P}$ obtained. Hence, the input to the two-stream architecture consists of image $\mathcal{I}$ and a patch $\mathcal{P}_\mathrm{i}$ defined by a pair of coordinates

$$[(\mathrm{x}_\mathrm{i}^{\mathrm{tl}}, \mathrm{y}_\mathrm{i}^{\mathrm{tl}}), (\mathrm{x}_\mathrm{i}^{\mathrm{br}}, \mathrm{y}_\mathrm{i}^{\mathrm{br}})] \tag{1}$$

where tl and br represent top-left and bottom-right. The pair of coordinates denote top-left and bottom-right corners of the box over the part of an image. Assuming top-left corner in the original image as the origin of a pixel coordinate system, x-axis and y-axis is defined from left-to-right and top-to-bottom respectively.

As shown in figure 2, there are two streams in the architecture. The top stream consists of a convolution-based feature extractor followed by the classification layer. The second stream takes patch from images and extracts feature presentations via CNN. These features are recurrently passed through LSTMs to get better and finer representations focusing on fine discriminative regions within the patch. These finer patches are weight-aggregated to form a single most discriminative representation. Specific details about the architecture are shared in the following sections.

Global Stream Given an input image $\mathcal{I}$, we first extract deep features by passing the image through a convolution neural network. The neural network is pretrained on ImageNet [27]. The extracted representations can be written as $\mathbf{W}_\mathrm{g} * \mathcal{I}$, where $\mathbf{W}_\mathrm{g}$ denotes the representative weight of the whole neural network and * denotes all the convolution, pooling, and non-linear functions performed on the input image. The features are further passed through a softmax layer which outputs a probability distribution over fine-grained categories. Mathematically,

$$\mathbf{G}_\mathrm{I} = \mathcal{F}(\mathbf{W}_\mathrm{g} * \mathcal{I}) \tag{2}$$

where $\mathbf{G}_\mathrm{I}$ represents global representation for image and $\mathcal{F}(.)$ denotes the Global Pooling Layer (GAP) [20] followed by a fully connected softmax layer which transforms the deep features into probabilities. The global stream is used to extract global representative features of the images. The reasons for including this simple branch are two-fold. First, to provide global information to the network during the training since the patches/parts of the object extracted focus on the object itself. Second, it provides a simple baseline over which our local stream can be added, demonstrating the plug-n-play functionality of our main contribution.

Local Stream The output of weakly supervised patch extraction framework is dominant parts of an image as $\mathcal{P}$ = [$\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$, ..., $\mathcal{P}_\mathrm{n}$], where each $\mathcal{P}_\mathrm{i}$ could be defined as a pair of coordinates of the bounding box for a region of an image. The regions are cropped from the entire image as shown in the figure 2. The set of cropped image regions can be denoted as $\mathcal{I}(\mathcal{P})$ = [$\mathcal{I}(\mathcal{P}_1)$, $\mathcal{I}(\mathcal{P}_2)$, $\mathcal{I}(\mathcal{P}_3)$, ..., $\mathcal{I}(\mathcal{P}_\mathrm{n})$]. Once a region $\mathcal{I}(\mathcal{P}_\mathrm{i})$ (say i_{th} patch) is cropped from image $\mathcal{I}$, it is passed through the pre-trained convolution neural network as:

$$\mathbf{F}_{\mathrm{i}} = (\mathbf{W}_{\mathrm{g}} * \mathcal{I}(\mathcal{P}_{\mathrm{i}})) \tag{3}$$

where $\mathbf{W}_{\mathrm{g}}$ represents the overall weights of CNN and * denotes convolution, pooling, and other non-linear functions. The dimension of output feature $\mathbf{F}_{\mathrm{i}}$ is w x h x c where w, h, c represents the width, height, and channel of the feature map. Note that the CNN in the global stream and the local stream does not share weights. The feature map $\mathbf{F}_{\mathrm{i}}$ is recurrently passed through different time steps of stacked-LSTMs. The motivation of this step is to make the details finer as the feature map of patch passes through several time steps of LSTMs. So, the input to each time step is the same feature map $\mathbf{F}_{\mathrm{i}}$. The output of the first layer of LSTMs is passed as input to the second layer. The temporal representative function of stacked-LSTMs can be denoted as ϕ. Hence, the outputs of stacked-LSTMs can be modeled as

$$[\phi(\mathbf{F}_{\mathrm{i}}^{1}), \phi(\mathbf{F}_{\mathrm{i}}^{2}), ..., \phi(\mathbf{F}_{\mathrm{i}}^{\mathrm{T}})] \tag{4}$$

where t = 1,2,3 ... T denotes the time steps of stacked-LSTMs and ϕ denotes the function modelled after each time step by LSTMcell. $\phi(\mathbf{F}_{\mathrm{i}}^{\mathrm{t}}) \in \mathbb{R}^{\mathrm{D}}$ is the D dimensional vector denoting output of feature part(i_{th} patch) $\mathbf{F}_{\mathrm{i}}$ at time step t. Our experiments 4.2 validates our hypothesis about how feature changes over the time steps to focus on finer details of parts.

Once we have finer details of a patch through the LSTM, an attention network is used to perform a weighted aggregation over these finer features. We believe the advantages of attention is two-fold. First, the trainable weights of attention layer help to provide more weights to the discriminative finer scale of the patch. The attention network helps to focus on the scale of the patch which maximizes the classification accuracy by removing the noisy parts. Secondly, the weighted aggregation of these different time-step features aggregates fine details within the patch. The output of the attention layer can be written as:

$$\mathbf{A}_{\mathrm{i}} = \sum_{\mathrm{t}=1}^{\mathrm{T}} \alpha^{\mathrm{t}} \phi(\mathbf{F}_{\mathrm{i}}^{\mathrm{t}}) \tag{5}$$

where

$$\alpha^{t} = \frac{exp(\mathbf{W}^{\mathrm{t}} \cdot \phi(\mathbf{F}_{\mathrm{i}}^{\mathrm{t}}))}{\sum_{\mathrm{i}=1}^{\mathrm{T}} exp(\mathbf{W}^{\mathrm{t}} \cdot \phi(\mathbf{F}_{\mathrm{i}}^{\mathrm{t}}))} \tag{6}$$

where $\mathbf{A}_{\mathrm{i}}$ is the output of attention network and $\mathbf{W}^{\mathrm{t}} \in \mathbb{R}^{\mathrm{D}}$ is the trainable weight parameter assigned to feature at each time step. Finally, the D-dimensional output from attention layer is to pass through a network of fully-connected neural network and softmax to generate class probability vector for fine-grained categories given by:

$$\mathbf{L}_{\mathrm{I}} = \mathcal{F}^{'}(\mathbf{W}_{\mathrm{l}} * \mathbf{A}_{\mathrm{i}}) \tag{7}$$

where $\mathbf{L}_{\mathrm{I}}$ represents the probability distribution, $\mathbf{W}_{\mathrm{l}}$ encapsulates the weights of full-connected layer after attention, $\mathcal{F}^{'}(.)$ denotes the softmax layer, and $\mathbf{A}_{\mathrm{i}}$ denotes the output from the attention network. Such design enforces the network to gradually attend to the most discriminative region of patch/part of the image and boost confidence in the prediction of an image.

3.2. Classification Loss

The proposed architecture is optimized using classification based loss function. Here, we used two different instances of the same classification loss. So, for a given image the multi-scale loss function can be defined as follows:

$$\mathcal{L}_{\mathrm{total}} = \sum_{\mathrm{n}=1}^{\mathrm{N}} [\mathcal{L}_{\mathrm{XE}}(Y_{\mathrm{n}}^{\mathrm{g}}, Y) + \lambda * \mathcal{L}_{\mathrm{XE}}(Y_{\mathrm{n}}^{\mathrm{l}}, Y)] \tag{8}$$

where $\mathcal{L}_{\mathrm{XE}}$ represents classification loss for N training samples. $Y_{\mathrm{n}}^{\mathrm{g}}$ denotes predicted label from the probability distribution of global image $\mathbf{G}_{\mathrm{I}}$ and correspondingly $Y_{\mathrm{n}}^{\mathrm{l}}$ denotes the probability distribution of patch representation of local stream $\mathbf{L}_{\mathrm{I}}$. Y is the ground truth label vector for n^{th} training image. λ controls the amount of patch representation's influence on global representation. The specific classification loss used is the cross-entropy loss given by:

$$\mathcal{L}_{\mathrm{XE}}(Y_{\mathrm{n}}^{\mathrm{g}}, Y) = -\sum_{\mathrm{k}=1}^{\mathrm{C}} Y^{\mathrm{k}} log Y_{\mathrm{n}}^{\mathrm{g}}, \tag{9}$$

where C denotes the total number of classes. Such a design helps the network to learn both global and region-based local patch representative features simultaneously.

3.3. Joint Representation

Once the network is trained end-to-end, we obtain two feature representations of an image $\mathcal{I}$, one from the global stream $\mathbf{G}_{\mathrm{I}}$ and another from the local stream $\mathbf{L}_{\mathrm{I}}$. These descriptors are global and finer part-attention region representations. Hence, to boost the performance we merge the feature output from two-stream to evaluate the performance on the test set. The merge is weighted is the same way as the losses of both streams are weighted.

4. Experiment Results

4.1. Implementation Details

Datasets We evaluated the usability and interpretability of our network on the following two datasets:

- **CUB200-2011**[33] is one of the most used fine-grained classification dataset with 11,788 images from 200 classes. We followed the conventional split with 5,994 training images and 5,794 test images.

- **Stanford Dogs**[14] contains 120 breeds of dogs taken from ImageNet. It has 20,580 images from 120 classes with 12000 training images and 8,580 test images.

Architectures We initialize the Convolutional Neural Network of both the stream with ImageNet pre-trained VGG network [30]. We do not use any part annotation or bounding box information. We obtained patches of an image by following the procedure in [32]. Both the streams are trained end-to-end simultaneously. Implementation details of streams are as follows:

- **Global Stream** We have followed the standard practice as per the literature. The input to the global CNN is 448 x 448 image. To reduce computation, we removed the fully connected layer from the classifier layer of VGG19 [30] and replace them with Global Average Pooling (GAP) layer [20]. The classifier layer is a randomly initialized single fully-connected layer.

- **Local Stream** The output of the weakly supervised network is a set of multiple patches for an image. These patches have varying spatial dimensions. Hence, before passing into local stream's CNN it is resized to 224 x 224. Then, the patch is passed through a pre-trained VGG19 [30] network. All the layers after conv5_4 are removed. Therefore the output of the network is a feature map of 512 x 14 x 14. The feature map is passed through another Global Average Pooling (GAP) layer to output a 512 -dimensional feature. This feature vector is passed through stacked-LSTMs with a hidden size of 512. Note that the input feature is the same across all the time-steps of LSTM hence it is computed only once. The number of time steps used is 10. The output of each step is fed to the attention layer which creates a soft-score based on equation 5. These scores are weight-multiplied with LSTM's features and summed to produce a representative feature of the same dimension as hidden layer (512). Finally, two fully-connected layers are used to change the 512 dimensions to the number of classes in datasets (200 in CUB200-2011 and 120 in Stanford dogs).
  End-to-end training of both streams proceeds with global and local stream having softmax with cross-entropy losses with weight 1.0 and 1.0 respectively. At test time, these softmax layers are removed and the prediction is based on the same weighted combination of these two streams.

4.2. Visualization and Analysis

Attention Areas: Insights into the behavior of the local branch can be obtained by visualizing the features of the attention layer and drawing the attention heatmap around the attended regions within the patch. We ran Grad-CAM [28] on the output of the local stream to visualize the finer attended region within the patch.

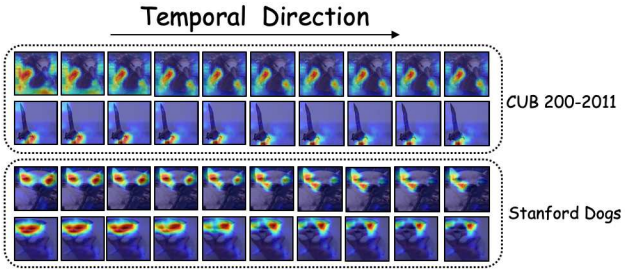

Figure 3. The diagram shows the image regions of the patch considered to be important via [28] by each of the 10 hidden representations of LSTMs for its prediction. As evident in the diagram, the attention regions in the patch become finer as we increase the number of time steps from 1 to 10.

Table 1. It shows the accuracy for our network over baseline for CUB200-2011 dataset[33]

Model	Accuracy(%)
VGG19 [30]	77.8
VGG19 + local-stream	**79.6**

The effect of hidden representations of LSTMs from various time-steps is shown in figure 3. Using Grad-CAM, [28] we can see the part of the image a time step's hidden representation attends to. Aligning with our motivation we can see that the attention in heatmap goes finer as we go further from the initial time step. As seen in figure 3 , the hidden representations in initial LSTMs focus on much broader areas of the patch, but as we recurrently pass the patch through the deeper LSTM cells the attention becomes finer and more discriminative. Moreover, in some cases 3 the attention spans changes from generic regions like the whole face to more subtle variations present in ears, feather, beak. This also shows that the representations at higher time steps are more discriminative producing higher responses.

Further, the simplicity of the module makes it possible to use it as a plug-n-play module. The local stream can be attached to any network which will be helpful to visualize how the network is attending to the various region of an image. It helps to inject interpretability and get a better understanding of the network evident from figure 3 . Also, we gain a boost in classification accuracy over the standard baseline as tabulated in Table 1 for CUB200 dataset and Table in 2 for Stanford dogs dataset.

Quantitatively, we tried to analyze the relationship between the level of finer details and being discriminative among the classes in table 3. As evident from the table, the feature representations of finer details become less discriminative as we pass it through more recurrent layers. We

processed a single patch repetitively and it started to overfit the finer details.

Table 2. It shows the accuracy for our network over baseline for Stanford Dogs dataset[14]

Model	Accuracy(%)
VGG19 [30]	77.2
VGG19 + local-stream	**78.7**

Table 3. Accuracy of fine detail representative feature at different time step t of LSTM

Feature at time step (t)	Accuracy(%)
1	78.90
2	79.22
3	**79.23**
4	79.11
5	79.04
6	79.09
7	79.08
8	79.03
9	79.03
10	79.04

Table 4. Effect of increasing LSTMs time step on classification accuracy for CUB200-2011 dataset[33]

Model	Accuracy(%)
VGG19 [30]	77.80
VGG19 + local-stream(CNN only)	77.79
VGG19 + local-stream(CNN + LSTM)	78.20
VGG19 + local-stream(CNN + LSTM + attention)	**79.60**

4.3. Ablation Study

We conducted the ablation studies to show how each component individually boost the accuracy of the overall model.

Effect of network components on classification As shown in Table 4, the presence of only Convolutional Neural Network in the local-stream doesn't add much performance benefit. Further, a stacked-LSTM layer is added in the local-stream. Here, the local-stream is trained using cross-entropy losses on the outputs of all the time steps. During inference, we only consider the output of the final step. This addition of the stacked-LSTM layer boosts the performance by a significant margin ($\sim$1%) , indicating the finer details are highly discriminative. Moreover, the attention layer provides extra gain to reach much better performance showing the effectiveness of weighted aggregation of the finer features.

Table 5. Effect of feature summation vs attention on CUB200-2011 dataset [33]. '$\sim$' denotes the summation of all the features between specific time steps.

Feature Summation	Accuracy(%)
1	78.90
$1 \sim 2$	78.78
$1 \sim 3$	79.18
$1 \sim 4$	78.75
$1 \sim 5$	77.04
$1 \sim 6$	75.32
$1 \sim 7$	72.97
$1 \sim 8$	71.01
$1 \sim 9$	69.21
$1 \sim 10$	67.64
Attention	**79.60**

Attention vs Summation We investigate the effect and importance of attention in the local-stream of the network, we tried to replace the attention layer with a simple summation of features. Table 5 shows the result of an experiment comparing simple summation of features from time step 1 to 10 with the attention layer. The results validate the claim that simple summing doesn't help to boost the accuracy while the attention layer explicitly learns the weights for each feature at the time steps. This helps the network to focus on the finer details which is most discriminative.

4.4. Hyperparameter Setting For Time Steps

We tried to see how the number of steps affects the overall classification accuracy of the network. We ran the network on CUB200-2011 dataset with different number of time steps in each run and recorded the results in Table 6 This shows that adding more time step does not necessarily increase the performance of the network. The results also align well with figure 3 showing diminishing difference in fine attention towards the end of the recurrent time steps.

Table 6. Effect of increasing LSTMs time step on the classification accuracy for CUB200-2011 dataset[33]

Time Steps	Accuracy(%)
5	77.72
10	**79.60**
15	79.12
20	79.34
25	78.98

5. Conclusion

In this paper, we propose a simple recurrent attention based module that extracts finer details from the image providing more discriminative features for fine-grained classification. The local stream of whole architecture aggregates these fine details into a representative and complementary feature vector. The proposed method does not need bounding box/part annotation for training and can be trained end-to-end. Moreover, the simplicity of the module makes it a plug-n-play module, thus, increasing its usability. Through the ablation study, we also show the effectiveness of each part of the network. Additionally, the interpretable nature of the module makes it easy to visualize learned discriminative patches.

References

[1] Sijia Cai, Wangmeng Zuo, and Lei Zhang. Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 511–520, 2017. 1

[2] Juan C Caicedo and Svetlana Lazebnik. Active object localization with deep reinforcement learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2488–2496, 2015. 2

[3] Yue Cao, Tianlong Chen, Zhangyang Wang, and Yang Shen. Learning to optimize in swarms. In *Advances in Neural Information Processing Systems*, pages 15018–15028, 2019. 2

[4] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, pages 8928–8939, 2019. 2

[5] Yin Cui, Feng Zhou, Jiang Wang, Xiao Liu, Yuanqing Lin, and Serge Belongie. Kernel pooling for convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2930, 2017. 1

[6] Jianlong Fu, Tao Mei, Kuiyuan Yang, Hanqing Lu, and Yong Rui. Tagging personal photos with transfer deep learning. In *Proceedings of the 24th International Conference on World Wide Web*, pages 344–354, 2015. 1

[7] Jianlong Fu, Yue Wu, Tao Mei, Jinqiao Wang, Hanqing Lu, and Yong Rui. Relaxing from vocabulary: Robust weakly-supervised deep learning for vocabulary-free image tagging. In *Proceedings of the IEEE international conference on computer vision*, pages 1985–1993, 2015. 1

[8] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4438–4446, 2017. 1, 2

[9] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 317–326, 2016. 2

[10] Weifeng Ge, Xiangru Lin, and Yizhou Yu. Weakly supervised complementary parts models for fine-grained image classification from the bottom up. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3034–3043, 2019. 1, 2

[11] 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. 2

[12] 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. 2

[13] Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang. Part-stacked cnn for fine-grained visual categorization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1173–1182, 2016. 2

[14] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011. 1, 2, 5, 6

[15] Shu Kong and Charless Fowlkes. Low-rank bilinear pooling for fine-grained classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 365–374, 2017. 2

[16] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 1

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[18] Peihua Li, Jiangtao Xie, Qilong Wang, and Zilin Gao. Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 947–955, 2018. 2

[19] Peihua Li, Jiangtao Xie, Qilong Wang, and Wangmeng Zuo. Is second-order information helpful for large-scale visual recognition? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2070–2078, 2017. 2

[20] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 3, 5

[21] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1457, 2015. 1, 2

[22] Xiao Liu, Tian Xia, Jiang Wang, and Yuanqing Lin. Fully convolutional attention localization networks: Efficient attention localization for fine-grained recognition. *arXiv preprint arXiv:1603.06765*, 1(2):4, 2016. 1

[23] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015. 2

[24] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014. 2

[25] Mohammad Moghimi, Serge J Belongie, Mohammad J Saberian, Jian Yang, Nuno Vasconcelos, and Li-Jia Li. Boosted convolutional neural networks. In *BMVC*, pages 24–1, 2016. 2

[26] Florent Perronnin and Diane Larlus. Fisher vectors meet neural networks: A hybrid classification architecture. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3743–3752, 2015. 2

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1, 3

[28] 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. 2, 5

[29] Marcel Simon and Erik Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1143–1151, 2015. 2

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014. 1, 2, 5, 6

[31] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 2

[32] Peng Tang, Xinggang Wang, Angtian Wang, Yongluan Yan, Wenyu Liu, Junzhou Huang, and Alan Yuille. Weakly supervised region proposal network and object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 352–368, 2018. 5

[33] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 1, 2, 4, 5, 6

[34] Dequan Wang, Zhiqiang Shen, Jie Shao, Wei Zhang, Xiangyang Xue, and Zheng Zhang. Multiple granularity descriptors for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision*, pages 2399–2406, 2015. 1, 2

[35] Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaxing Zhang, Yuxin Peng, and Zheng Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 842–850, 2015. 2

[36] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015. 2

[37] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018. 2

[38] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 5209–5217, 2017. 1

[39] 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. 2