SEARCH FOR FINER ATTENTION DETAILS FOR FINE-GRAINED IMAGE CLASSIFICATION

A Thesis

by

PRATEEK SHROFF

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Zhangyang (Atlas) Wang
Co-Chair of Committee,	Nima Khademi Kalantari
Committee Member,	Tie Liu
Head of Department,	Scoff Schaefer

August 2020

Major Subject: Computer Science

ABSTRACT*

The task of general object classification via images has been studied extensively. It involves distinguishing very different object categories like a dog or a cat. On the other hand, the task of fine-grained classification deals with the recognition of images having subtle visual differences among the classes or categories. The marginal visual difference between different classes in fine-grained images makes this very task harder.

The work of this thesis is inspired by how the human visual system looks for fine attention details to recognize an object in the image. Our brain is trained to look for some particular fine discriminative details by repetitively scanning through the image. Through our work, we tried to focus on these marginal differences to extract more representative latent features via deep learning models. Similar to human vision, our network recurrently focuses on the parts of images to spot small discriminative parts among the classes. Moreover, we show through interpretability techniques how our network focus changes from coarser to finer details. Our network uses only image-level labels and does not need bounding box/part annotation information to spot these changes. Further, the simplicity of our network makes it an easy plug-n-play module increasing its usability in other applications.

*Part of this section is adapted from the published paper, "Focus Longer to See Better: Recursively Refined Attention For Fine-Grained Image Classification" by Prateek Shroff, Tianlong Chen, Yunchao Wei, and Zhangyang Wang, Conference on Computer Vision and Pattern Recognition (CVPR) Workshops ©2020 IEEE.

DEDICATION

To my mother, my father, and my entire family who supported me throughout my career.

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to my supervisor, Prof. Zhangyang (Atlas) Wang of Department of Computer Science and Engineering at Texas A&M University for the continuous support of my work and research through-out my masters even when the road got tough. He has always encouraged me to challenge the status quo and has provided numerous valuable life-long learning that would surely help me in the future. I would also like to thank my co-advisor Prof. Nima Khademi Kalantari, and committee member Tie Liu for their guidance during the work.

I would especially like to thank my fellow Vita Informatics Group (VITA) lab member Tianlong Chen without whose technical and intellectual guidance; this thesis would never have taken this shape.

Finally, and most importantly, I would like to acknowledge the support of my family and friends for always encouraging me throughout my years of study.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a thesis committee consisting of Professor Zhangyang (Atlas) Wang [advisor], Professor Nima Khademi Kalantari of the Department of Computer Science and Engineering and Professor Tie Liu of the Department of Electrical and Computer Engineering. All other work conducted for the thesis was completed by the student with timely advise from fellow colleague Tianlong Chen.

Funding Sources

This work did not have any funding.

NOMENCLATURE

RCNN	Region-Convolutional Neural Network
CNN	Convolutional Neural Network
FG	Fine Grained
DNN	Deep Neural Network
RNN	Recurrent Neural Network
FFT	Fast Fourier Transform
BN	Batch Normalization
LSTM	Long-Short Term Memory
MLP	Multi Layer Perceptron
ML	Machine Learning
VGG	Visual Geometry Group
GAP	Global Average Pooling
CAM	Class Activation Maps
Grad-CAM	Gradient Class Activation Map

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION*

1.1 Overview

Image classification is one of the most researched area in the field of computer vision and machine learning [1, 2, 3]. The aim of this problem is to identify a object within an image like identification of dog or cat in an given image. The various methods developed in the area of image classification forms a solid bedrock over which more more advanced problems like object detection [4, 5, 6], visual question answering [7], image segmentation [8, 9] are developed. Figure

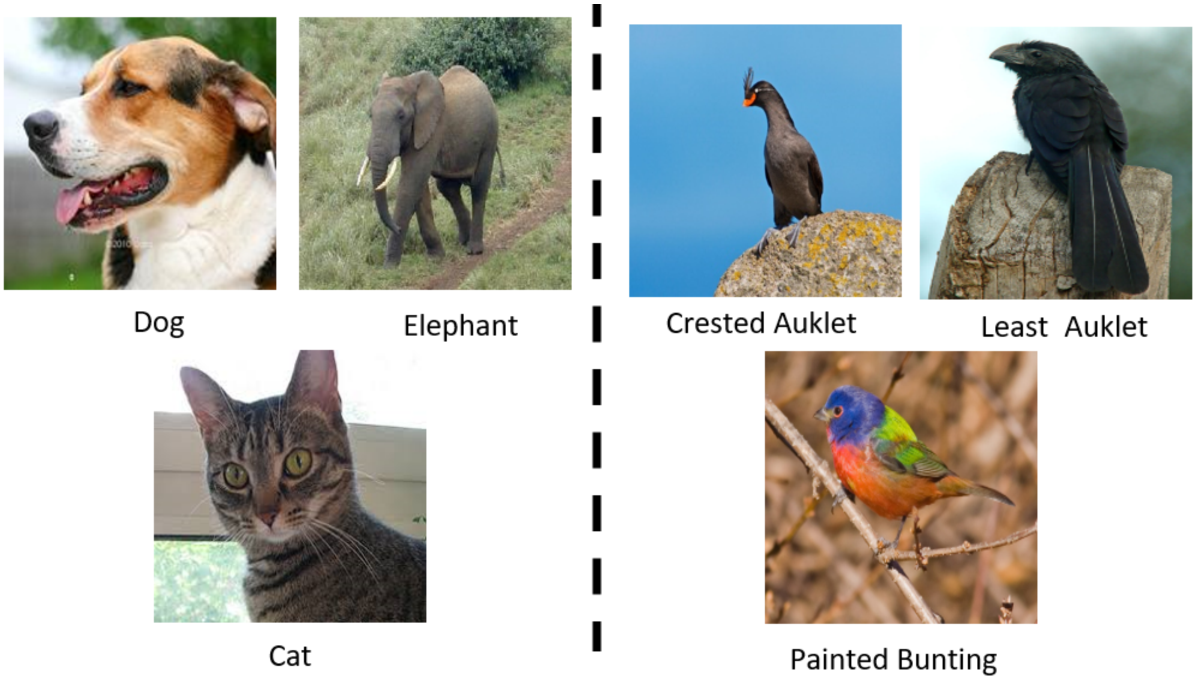


Figure 1.1: Figure shows the distinction between generic image classification and fine-grained classification. The generic image classification usually distinct very different image categories like dog, cat, elephant but fine-grained recognition deals with fine categories like bird species.

*Part of this section is adapted from the published paper, "Focus Longer to See Better: Recursively Refined Attention For Fine-Grained Image Classification" by Prateek Shroff, Tianlong Chen, Yunchao Wei, and Zhangyang Wang, Conference on Computer Vision and Pattern Recognition (CVPR) Workshops ©2020 IEEE.

1.1 shows the difference between generic and fine grained image classification. Generic image classification deals with recognizing a broader category like dog, cat, person. On the other hand, the fine-grained image classification deals with recognizing identity of an object at much finer level like bird species, dog species. This makes the fine grained recognition problem much more intricate. For example, answering whether an image has cat or dog is much easier than telling which specie of bird is present in the image. Fine-grained image classification is challenging due to very small inter-class but large intra-class variations among the classes (identities). The presence of small inter-class variation is due to very subtle differences within these classes as shown in fig 1.2 while there could be large intra-class variations due to difference in illumination, background, pose, scales, and viewpoint as evident in figure 1.3.

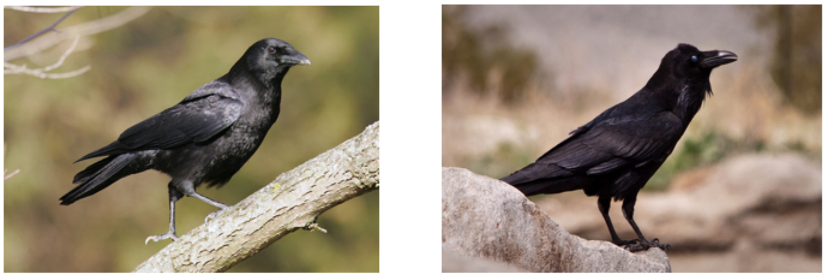


Figure 1.2: Figure shows the presence of very small and subtle variations between different classes/categories in fine-grained dataset.

Fine-grained image classification has been an active research area that recognizes sub-categories (SUV, jeep, sedan) within some meta-category(cars). Deep neural networks have shown astounding performance in generic image classification. It has become the de-facto tool to extract powerful representations from the images. Deep learning approaches for fine-grained classification fall into two separate paradigms: localization-classification network [10, 11, 12], and end-to-end feature encoding [13, 14, 15]. 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

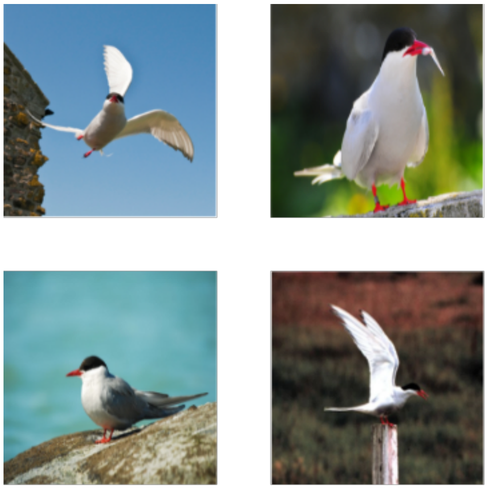


Figure 1.3: Figure shows the presence of variations in form of scale, view, winged/non-winged pose among the images of a same class.

regions/parts. In order to localize these fine changes, earlier work [16, 17] 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 [18] and Stanford dataset [19, 20] but not feasible for large-scale image dataset say ImageNet [1]. Convolution neural networks (CNNs) were hence leveraged for weakly supervised part-learning with category-labels only, assuming no dependencies on bounding box/part annotations [21, 22, 23].

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. The localization network is used to extract very fine grained details which are discriminative among classes. Figure 1.4 shows an example of presence of a (only) fine detail which is different in two very similar yet different classes. This complementary network architecture requires separate losses [16, 10] 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 [13, 24, 25, 26]. One of the first works in this category was the use of Bilinear CNNs [13] which computes pairwise feature interactions by two independent CNNs to capture the local differences in the image. Another work [27] 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.

Following the paradigm of localization-classification network, we tried to study the problem of fine grained recognition specifically in the context of finding these fine subtle variation and the level of granularity where they are most discriminative.

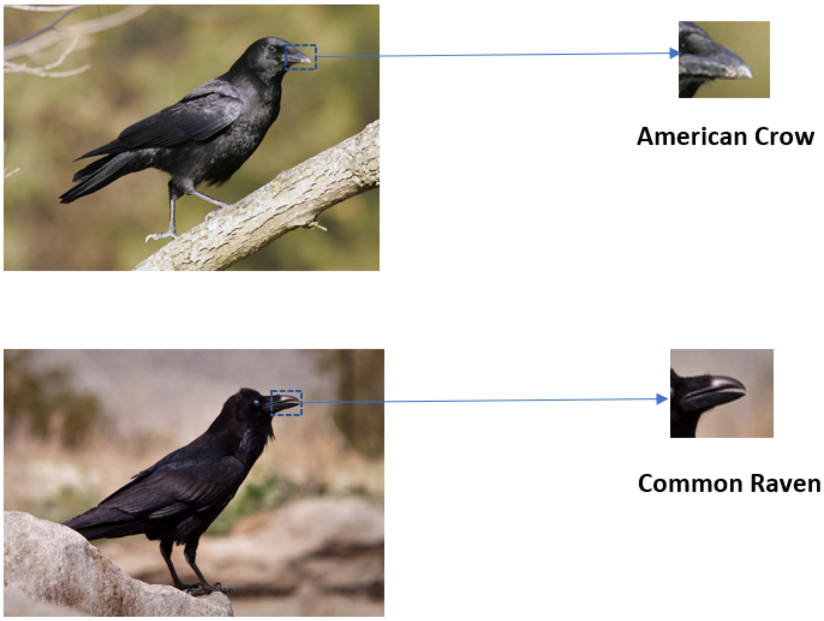


Figure 1.4: Figure shows the fine difference between two very visually similar yet different categories of images. Only focusing on discriminative region (in this case peck) can help to identify correct class.

1.2 Research Questions

The common thread between various designs and architectures in the field of fine grained image recognition is the presence of localize network to extract the marginal discriminative representations of the parts of image. But they don't explore the scale at which these marginal differences provide most information helpful in discrimination. Moreover, very less research has been done to make them explainable.

Through our work on this topic, we aim to tackle following two questions:

- What is the level of granularity of distinguishing parts that provide most benefit to the classification accuracy?
- How to view these marginal differences in human-interpretable form?
- We also analyzed the "What" and "Where". "What" features are most dicriminative and network attend to. Also, "Where" is it present in whole image.

1.3 Contributions

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 uni-directional 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. Specifically, we used the 'soft' attention methodology that discredits irrelevant areas and focuses on discriminative finer scale. 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 (weighted) 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 [18], Stanford Dogs [19]), and show performance boosts over the baselines.

1.4 Outline

The remainder of the thesis is organized into following sections. Section 2 provides a brief survey of various parts of model referred. Section 3 discusses in detail the approach and methodology of our design. This includes the architecture, design setup, datasets, and other experimental details. Section 4 discuss the ablation studies to support the results and how it answers our research problem. Section 5 quantifies our results. Section 6 sheds some light on future direction. Finally, section 7 discuss the strength and limitation of approach leading to the conclusion.

2. RELATED WORK*

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 [28, 29, 30, 31] forms the backbone for most of the recognition task. This has shown a great boost in performance when compared to hand-crafted features. To model subtle difference, a bilinear structure [13] is used to compute pairwise differences. The use of boosting to combine the representation of multiple learners also helps to improve classification accuracy [32].

Many models [33, 34] falls under the localization-classification paradigm. The main idea behind these approaches is to first find the discriminative regions and then compare their appearance. The localization framework requires the semantic parts like (head, body) to be shared among objects in the same class yet be discriminative to be different across other classes. Also, another paradigm uses second-order information in fine-grained feature extraction. Pooling methods that utilize second-order information[35, 26] have proven to enhance the extraction of more meaningful information.

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 [36, 34, 37, 38, 39] to develop interpretable models that visualize regions attended by the network. In [36], Class Activation Maps (CAMs) are used to provide object-level attention thus not providing much finer discriminative details. Over time, there have been variants developed [40, 41], that explore the backward propagation to identify salient image features. In [34, 37, 38], attentions are at a much finer level where the focus is more on the parts of the object that are discriminative rather than the whole body/object. In [39], the authors associate the prototypical aspect to the object part to reason out the classification prediction for an image. Our network makes a simple approach based

*Part of this section is adapted from the published paper, "Focus Longer to See Better: Recursively Refined Attention For Fine-Grained Image Classification" by Prateek Shroff, Tianlong Chen, Yunchao Wei, and Zhangyang Wang, Conference on Computer Vision and Pattern Recognition (CVPR) Workshops ©2020 IEEE.

on [40] to visualize the fine attention areas in the patches.

Attention has been incorporated in visual related tasks from long time [42, 43, 44, 45]. 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 [12], 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. METHODOLOGY*

3.1 Our Proposal

3.1.1 Overview

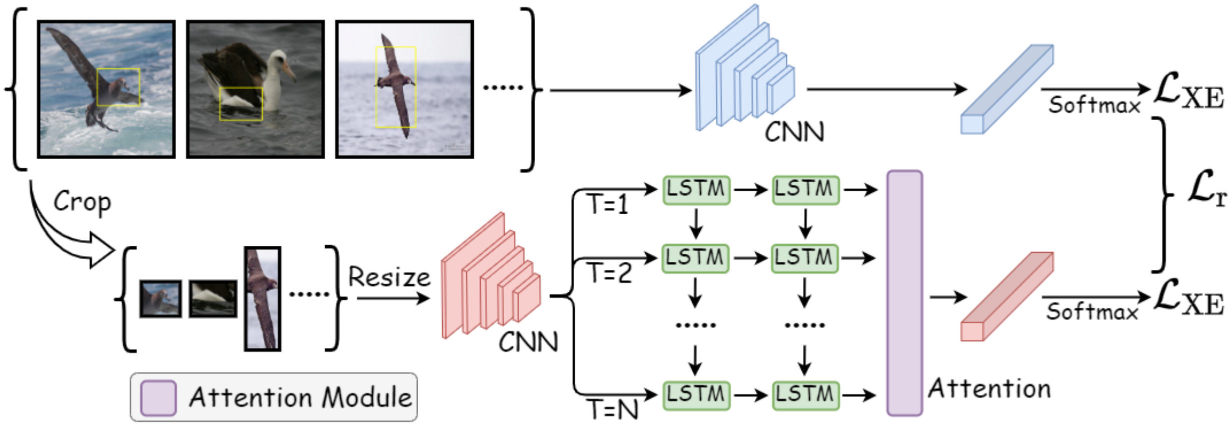


Figure 3.1: The pipeline of our two-stream architecture.

Figure 3.1 shows the pipeline of our overall architecture. 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 approach is used to extract meaningful patches (parts of the images) [46]. 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. The global branch is a simple CNN based feature extractor that extract features from whole image providing global representational context. On the other hand, the local branch takes in a part of image and recurrently process it top-down to extract fine representations at several levels. This is

*Part of this section is adapted from the published paper, "Focus Longer to See Better: Recursively Refined Attention For Fine-Grained Image Classification" by Prateek Shroff, Tianlong Chen, Yunchao Wei, and Zhangyang Wang, Conference on Computer Vision and Pattern Recognition (CVPR) Workshops ©2020 IEEE.

followed by an attention layer to attend to most discriminative attention details in view to boost the overall classification accuracy.

3.1.2 Weakly Supervised Patch Extraction

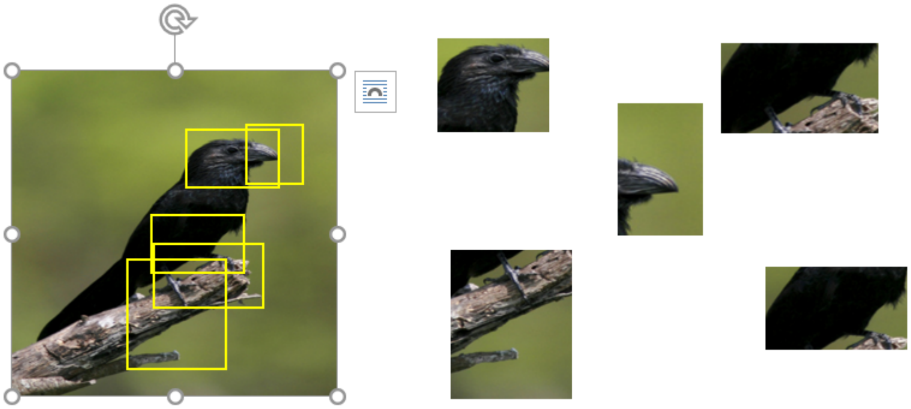


Figure 3.2: Output of weakly supervised patch detection

The local stream of the architecture (defined here 3.1.3.2) uses a part of an image specifically the region of the image representative of some part of the image. Many datasets provide the part-annotation information but we wanted our method to be generalize to the datasets even when those annotation information is not available. Hence, we generated our patches in weakly supervised fashion. That means to extract the significant parts of image only category level information is utilized. There has been many methods to extract the patches in weakly supervised manner [47, 12, 48]. We specifically used the methodology in [39]. The methodology defines the prototypical patches for each of the images according to the highest activation of feature maps during the training phase. We tune this architecture to extract N patches for each image as shown in figure 3.2.

3.1.3 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.

$$[(\mathrm{x}_{\mathrm{i}}^{\mathrm{tl}}, \mathrm{y}_{\mathrm{i}}^{\mathrm{tl}}), (\mathrm{x}_{\mathrm{i}}^{\mathrm{br}}, \mathrm{y}_{\mathrm{i}}^{\mathrm{br}})] \tag{3.1}$$

where tl and br represents top-left and bottom-right. The pair of coordinates denotes top-left and bottom-right corners of the box over the part of 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 3.1, there are two streams in the architecture. The top stream consists of a convolution-based feature extractor followed by the classification layer and softmax layer. The second stream takes patch from images and extracts feature representations via CNN. These features are recurrently passed through Long-Short Term Memory (LSTMs) to get better and finer representations focusing on fine discriminative regions within the patch. These finer patches are weight-aggregated via an attention layer. Finally, the output of attention layer is passed through classification and softmax layer. Both of the stream are optimized via the weighted-loss function formed by adding the cross-entropy losses from the both the stream. Both of these streams are discussed in detailed in the following subsections.

3.1.3.1 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 [1]. 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,

$$\mathbf{G}_{\mathrm{I}} = \mathcal{F}(\mathbf{W}_{\mathrm{g}} * \mathcal{I}) \tag{3.2}$$

where $\mathbf{G}_{\mathrm{I}}$ represents global representation for image and $\mathcal{F}(.)$ denotes the Global Pooling Layer (GAP) [49] followed by a full-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, the motivation is to provide more global information into the network during the training since the patches/part 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.

3.1.3.2 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 bounding box for a region of an image. The image regions are cropped from the image as shown in the figure 3.1. 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:

$$\mathbf{F}_{\mathrm{i}} = (\mathbf{W}_{\mathrm{g}} * \mathcal{I}(\mathcal{P}_{\mathrm{i}})) \tag{3.3}$$

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 feature map. Note that the CNN in global stream and 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

$$[\phi(\mathbf{F}_{\mathrm{i}}^{1}), \phi(\mathbf{F}_{\mathrm{i}}^{2}), ..., \phi(\mathbf{F}_{\mathrm{i}}^{\mathrm{T}})] \tag{3.4}$$

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 5 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 attention layer can be written as:

$$\mathbf{A}_{\mathrm{i}} = \sum_{\mathrm{t}=1}^{\mathrm{T}} \alpha^{\mathrm{t}} \phi(\mathbf{F}_{\mathrm{i}}^{\mathrm{t}}) \tag{3.5}$$

where

$$\alpha^{t} = \frac{exp(\mathbf{W}^{\mathrm{t}} \cdot \phi(\mathbf{F}_{\mathrm{i}}^{\mathrm{t}}))}{\sum_{\mathrm{t}=1}^{\mathrm{T}} exp(\mathbf{W}^{\mathrm{t}} \cdot \phi(\mathbf{F}_{\mathrm{i}}^{\mathrm{t}}))} \tag{3.6}$$

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:

$$\mathbf{L}_{\mathrm{I}} = \mathcal{F}^{'}(\mathbf{W}_{\mathrm{l}} * \mathbf{A}_{\mathrm{i}}) \tag{3.7}$$

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 output from 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.

3.1.4 Loss Functions

The proposed dual-stream architecture is optimized via two different types of loss functions, i.e., classification loss and margin-based ranking loss. Specifically, we minimize the following joint multi-task loss function:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{c}}(\mathbf{Y}_{\mathrm{n}}^{\mathrm{g}}, \mathbf{Y}_{\mathrm{n}}^{\mathrm{l}}, \mathbf{Y}) + \lambda * \mathcal{L}_{\mathrm{r}}(p_t^{\mathrm{g}},\ p_t^{\mathrm{l}}) \tag{3.8}$$

Where λ limits contribution of ranking loss to the overall objective function and is empirically set. The meaning of the terms involved in loss function is discussed in the following section.

3.1.4.1 Classification Loss

One of the loss function used to optimize network is the cross-entropy based classification loss. 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:

$$\mathcal{L}_{\mathrm{c}} = \sum_{\mathrm{n}=1}^{\mathrm{N}} [\mathcal{L}_{\mathrm{XE}}(\mathbf{Y}_{\mathrm{n}}^{\mathrm{g}}, \mathbf{Y}) + \lambda^{'} * \mathcal{L}_{\mathrm{XE}}(\mathbf{Y}_{\mathrm{n}}^{\mathrm{l}}, \mathbf{Y})] \tag{3.9}$$

where $\mathcal{L}_{\mathrm{XE}}$ represents classification loss over n^{th} sample. $\mathbf{Y}_{\mathrm{n}}^{\mathrm{g}}$ denotes predicted label from the probability distribution of global image $\mathbf{G}_{\mathrm{I}}$ and correspondingly $\mathbf{Y}_{\mathrm{n}}^{\mathrm{l}}$ denotes the predicted label from probability distribution of patch representation of local stream $\mathbf{L}_{\mathrm{I}}$. $\mathbf{Y}$ is the ground truth label vector for n^{th} training image. The λ_1 controls the amount of patch representation's influence on overall optimization. The specific classification loss used is the cross-entropy loss given by:

$$\mathcal{L}_{\mathrm{XE}}(\mathbf{Y}_{\mathrm{n}}^{\mathrm{g}}, \mathbf{Y}) = -\sum_{\mathrm{k}=1}^{\mathrm{C}} \mathbf{Y}^{\mathrm{k}} log \mathbf{Y}_{\mathrm{n}}^{\mathrm{g}} \tag{3.10}$$

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.

3.1.4.2 Ranking Loss

Another loss function used is inspired from [10]. The margin based loss function is denoted by $\mathcal{L}_{\mathrm{r}}$ and given by:

$$\mathcal{L}_{\mathrm{r}}(p_t^{\mathrm{g}},\, p_t^{\mathrm{l}}) = \max\{0, p_t^{\mathrm{g}} - p_t^{\mathrm{l}} + \mathrm{margin}\} \tag{3.11}$$

where p_t denotes the probability corresponding to correct label t. This enforces that difference between the probability of correct label for global branch and for local branch is not greater than some margin. The global branch uses whole image context and learns the higher level semantics which produces better results with much higher confidence. Therefore, we want the local-branch to also take predictions from global branch as reference. This leads the local-branch to generate results with much more confidence.

3.1.5 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 DETAILS*

4.1 Dataset

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

- **CUB200-2011**[18] 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**[19] contains 120 breeds of dogs taken from ImageNet. It has 20,580 images with 12,000 training images and 8,580 test images.

4.2 Implementation Details

We initialize the Convolutional Neural Network of both the stream with ImageNet pre-trained VGG network, specifically VGG19 variant. We do not use any part annotation or bounding box information. The patches are extracted in a weakly supervised manner. For the global stream, We have followed the standard practice as per literature. The input to the global-branch CNN is 448x448 image. To reduce computation, we removed all the fully connected layers before the classifier layer of VGG19 and replace them with the Global Average Pooling (GAP) layer followed by a classifier layer and softmax.

For 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 the patches are resized to 224 x 224. The feature map from the final convolutional layer is passed through another Global Average Pooling (GAP) layer to output a 512 -dimensional feature(D). This feature vector is passed through stacked-LSTMs with a hidden size of 512. Note that the feature vector is the same across all the time-steps of LSTMs hence it is computed only

*Part of this section is adapted from the published paper, "Focus Longer to See Better: Recursively Refined Attention For Fine-Grained Image Classification" by Prateek Shroff, Tianlong Chen, Yunchao Wei, and Zhangyang Wang, Conference on Computer Vision and Pattern Recognition (CVPR) Workshops ©2020 IEEE.

once. The output of each step is fed to the attention layer which creates a soft-score for each of the time steps. These scores are multiplied with LSTM's features and summed to produce a representative feature of the same dimension (512).

The global and local branch VGG19 uses different base learning rate. The initial learning rate for global branch is 0.001 and for local branch is 0.01 . The every 40 epochs , the learning of both stream is reduced by a factor of 0.1.

End-to-end training of both streams proceeds with global and local stream having softmax with cross-entropy losses and ranking losses. The margin for ranking loss is empirically set to 0.5. The λ is set to 0.5 and $\lambda^{'}$ is set to 1.0 . At test time, these softmax layers are removed and the prediction is based on the same weighted combination of the features from two stream.

5. ANALYSIS AND RESULTS*

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 [40] on the output of the local stream to visualize the finer attended region within the patch.

The effect of hidden representations of LSTMs from various time-steps is shown in figure 5.1. Using Grad-CAM, [40] 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 deeper from the initial time step. As seen in figure 5.1 , the initial hidden representations in 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. Moreover, in some cases 5.1 the attention spans changes from generic regions like the whole face to more subtle variations in ears feather, beak 5.2. But finer does not mean it is more discriminative. Different image/patches might require focusing on different level of fineness in order to be most discriminative. Hence, a soft attention layer is used that provides more weights to representative scale of patch. This helps to attend to most discriminative level or region of patch which maximizes the classification accuracy.

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 5.2

Quantitatively, We tried to see the effect of the addition of this local stream (providing fine discriminative features) representations to the global stream (providing the higher level semantic) on the overall classification accuracy. We ran over two datasets described in section 4.1. The results show we gain a boost in classification accuracy over the standard baseline as tabulated in

*Part of this section is adapted from the published paper, "Focus Longer to See Better: Recursively Refined Attention For Fine-Grained Image Classification" by Prateek Shroff, Tianlong Chen, Yunchao Wei, and Zhangyang Wang, Conference on Computer Vision and Pattern Recognition (CVPR) Workshops ©2020 IEEE.

Table 5.1: It shows the accuracy for our network over baseline for CUB200-2011 dataset ©2020 IEEE.

Model	**Accuracy(%)**
VGG19 [28]	77.8
VGG19 + local-stream	**79.8**

table 5.1 for CUB200 dataset and table in 5.2 for Stanford dogs dataset.

Table 5.2: It shows the accuracy for our network over baseline for Stanford Dogs dataset ©2020 IEEE.

Model	**Accuracy(%)**
VGG19 [28]	77.2
VGG19 + local-stream	**78.7**

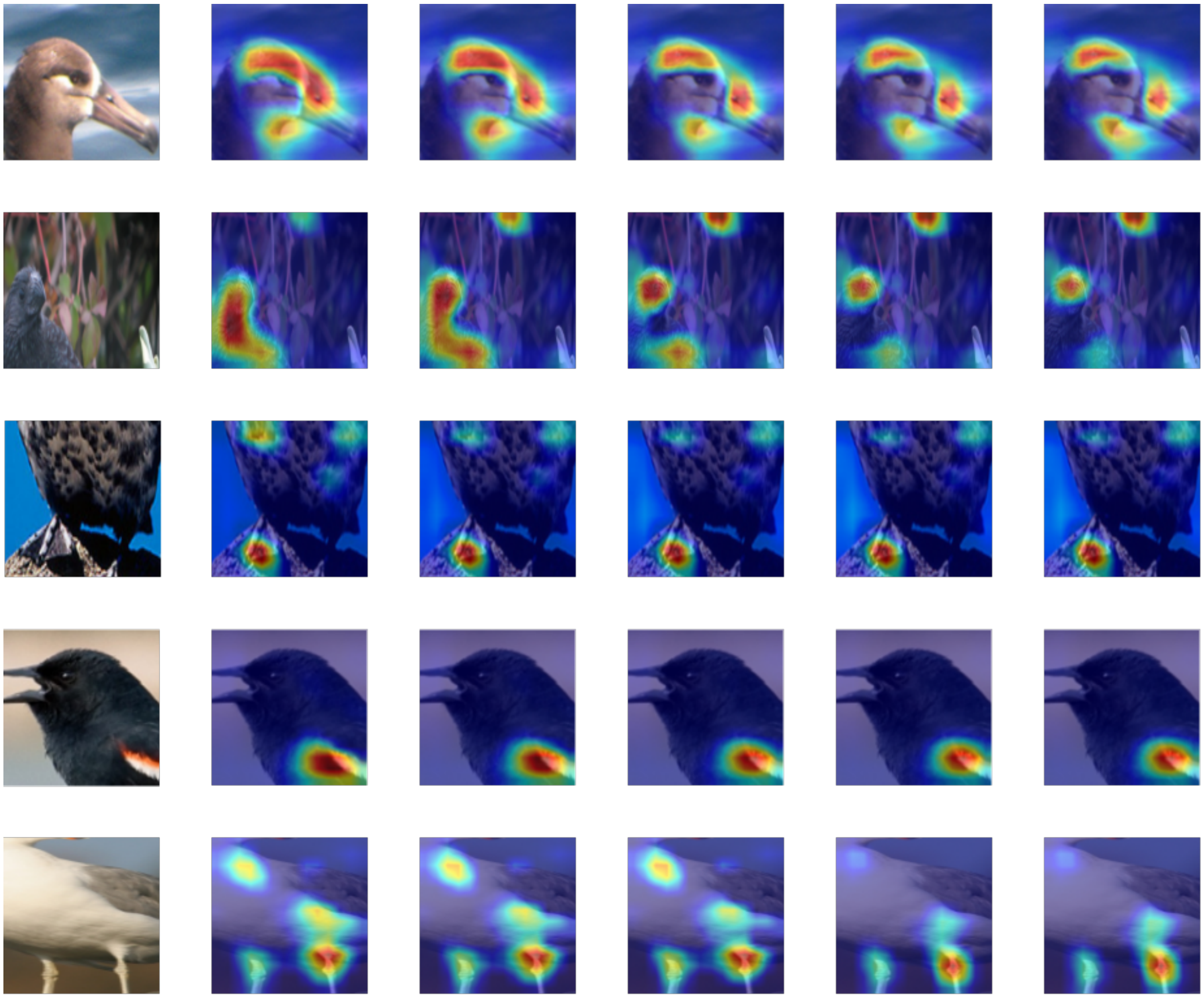


Figure 5.1: Attention maps corresponding to hidden representations of LSTMs for CUB200-2011 dataset.

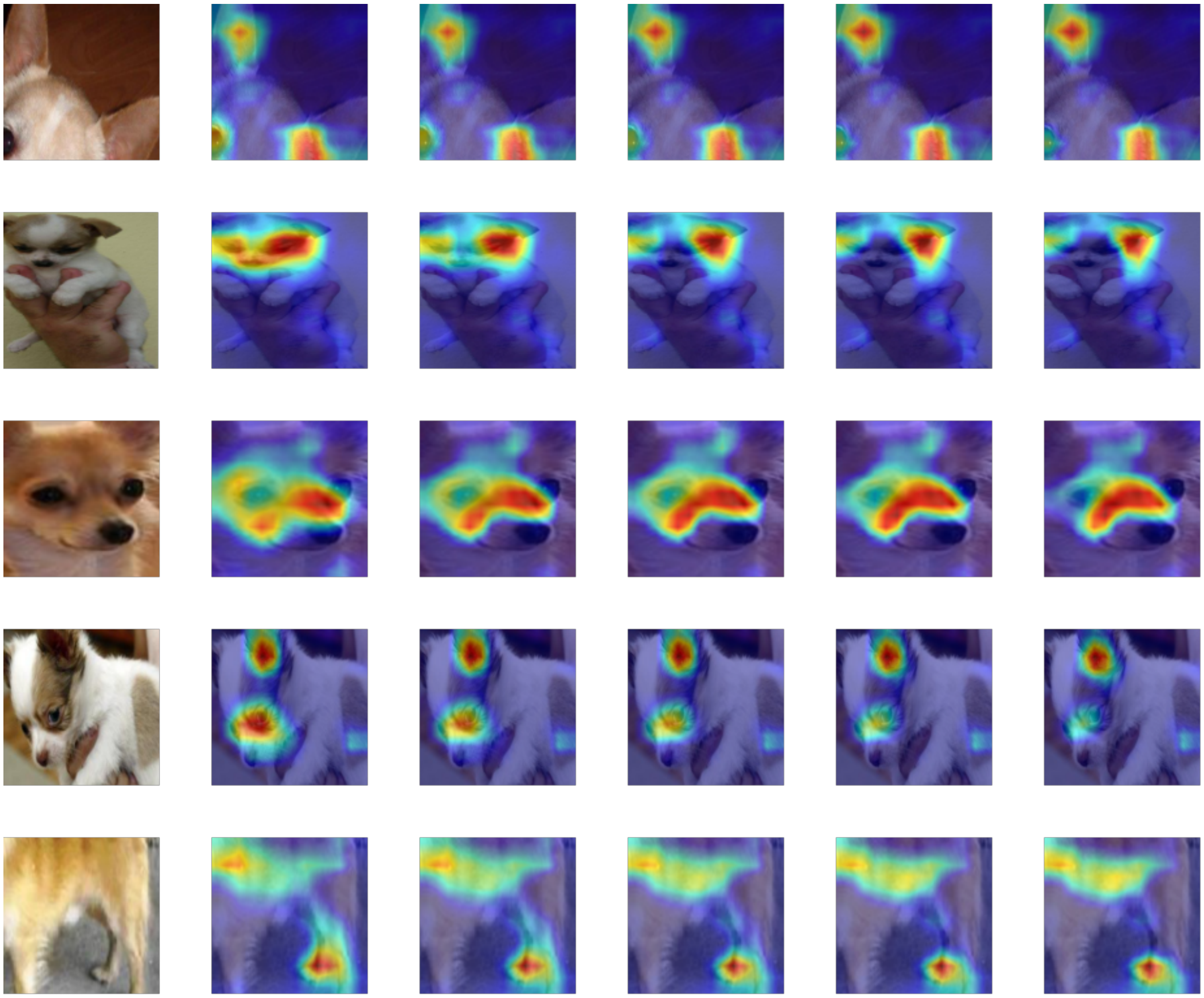


Figure 5.2: Attention maps corresponding to hidden representations of LSTMs for Stanford Dog dataset.

6. ABLATION STUDIES[*]

We conducted the following ablation studies to see how different components aid for the overall increase in classification accuracy.

6.1 Effect of Network Components on Classification

Table 6.1: Effect of different components on classification accuracy for CUB200-2011 dataset ©2020 IEEE.

Model	**Accuracy(%)**
VGG19 [28]	77.80
VGG19 + local-stream(CNN only)	77.79
VGG19 + local-stream(CNN + LSTM)	78.20
VGG19 + local-stream(CNN + LSTM + attention)	**79.60**

As shown in table 6.1, presence of only Convolutional Neural Network in 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 final step. This addition of stacked-LSTM layer boost the performance by a significant margin ($\sim 1\%$) , indicating the finer details provide the discriminative information. Moreover, attention layer provides extra gain to reach much better performance showing the effectiveness of weighted aggregation of the finer features which puts more attention to the fine scale providing better discriminative features.

[*] Part of this section is adapted from the published paper, "Focus Longer to See Better: Recursively Refined Attention For Fine-Grained Image Classification" by Prateek Shroff, Tianlong Chen, Yunchao Wei, and Zhangyang Wang, Conference on Computer Vision and Pattern Recognition (CVPR) Workshops ©2020 IEEE.

6.2 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. This can be viewed as provided equal weightage ($= 1$) to each scale of fine features. Table 6.2 shows the result of an experiment comparing simple summation of features from time step 1 to 10 with attention layer. Each row ($t_1 \sim t_2$) represents the summation of feature from time step t_1 to t_2. We clearly see that the simple summation of the feature does not help in classification score. In fact, in some cases it decreases the overall accuracy. The result validates 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. Hence, attention plays a pivotal role in the network.

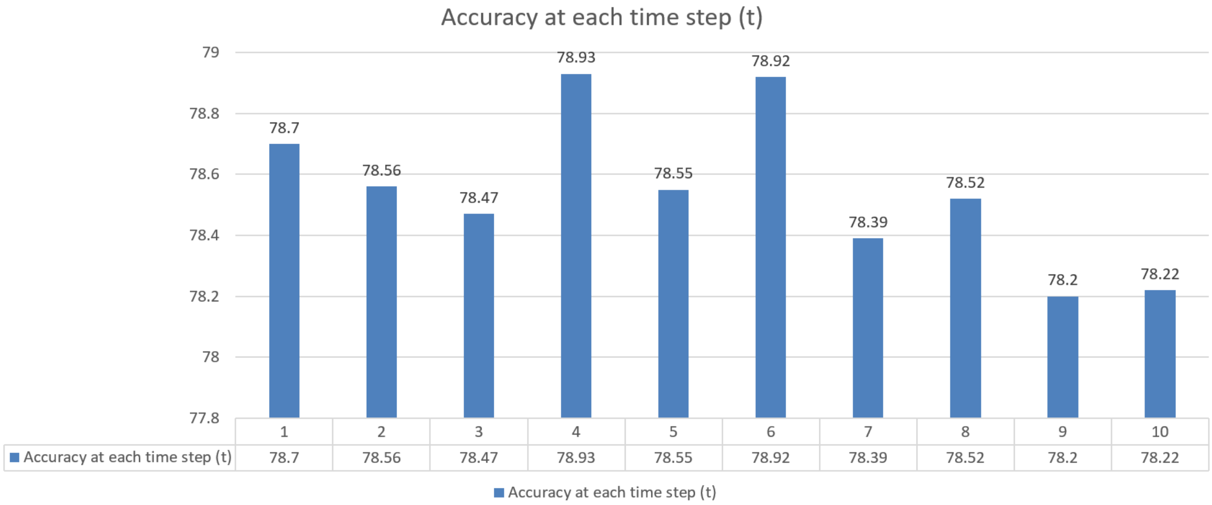
Table 6.2: Effect of feature summation vs attention on CUB200-2011 dataset. '$\sim$' denotes the summation of all the features between specific time steps ©2020 IEEE.

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**

6.3 Are All Scale Equally Discriminative?

The features at different time-steps of LSTMs attend to different scale of the discriminative region. We wanted to see if all different scales of these regions are equally discriminative? We also want to explore if there is any dominant scale of representation.

We hypothesis that even though the region of discriminative parts become finer that does not automatically lead to better translation in the classification accuracy. We also believe that there is no one dominant scale of representation that always gives the best classification score.



	1	2	3	4	5	6	7	8	9	10
■ Accuracy at each time step (t)	78.7	78.56	78.47	78.93	78.55	78.92	78.39	78.52	78.2	78.22

Figure 6.1: Accuracy corresponding to using feature at different time steps of LSTM

Figure 6.1 shows the accuracy of using the feature of only a certain time step. There are two things to observe in this case. First, there is quiet a variations in the accuracy of different time steps. Also, there is no dominant feature that is better than rest of the features. The different between time step 4 and 6 is negligible.

Second observation is there is no upward or downward trend. This states that there is no relationship between the fineness of the region and the scale being discriminative. For a different region, a certain level of fine-ness may be "most" discriminative at time step where another region may be not. Hence, each region is discriminative at a certain or multiple fine level(s). This is also evident from the figure 6.2. Each row shows the attention maps corresponding to features at different time steps and the classification output (correct/incorrect) when corresponding time step feature is used. As seen in figure, the top row has correct classification score at 1,5,6,9 while incorrect at 2. On the other hand, the bottom row has correct classification score at 2,5 but incorrect at 1,6,9 . So, we don't find see any dominant trend between time steps of LSTMs and classification accuracy.

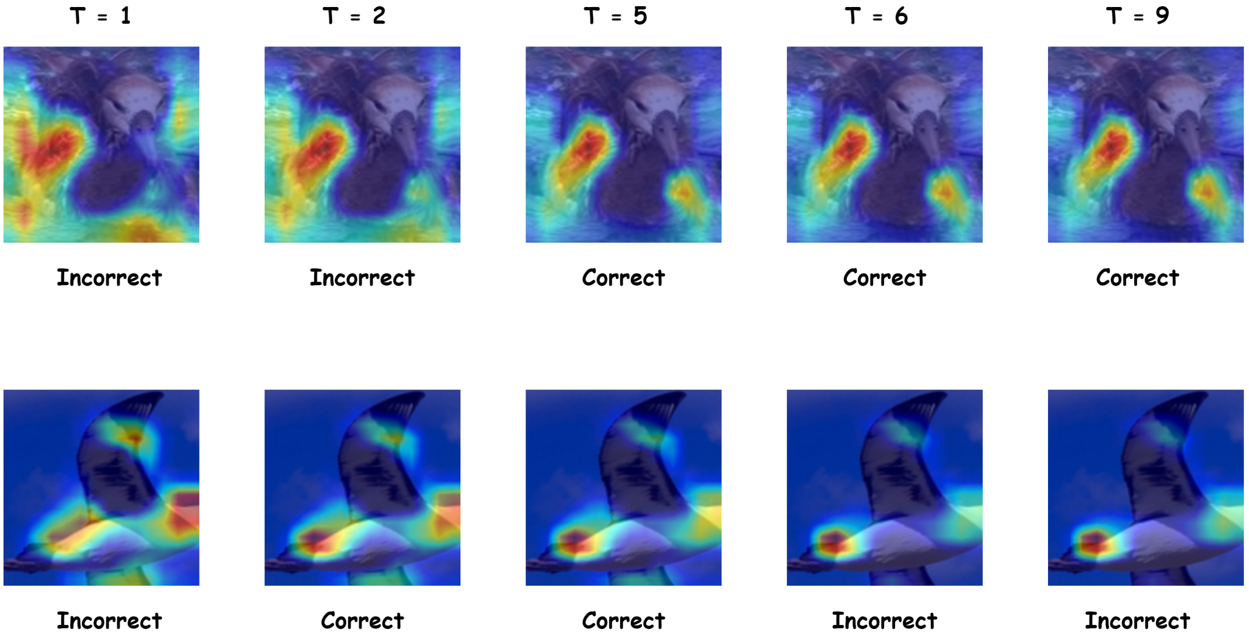


Figure 6.2: The figure shows the correct/incorrect classification when feature corresponding to time step 1,2,5,6,9 is used only for classification.

7. CAN WE GET FINER? ADDING SELF-ATTENTION

With the emergence of transformers and its other variants in language modelling field, multi-headed self-attention has become ubiquitous in the field. The variants acting on images based feature map has helped to increase the attention accuracy in the field of vision too [50, 51]. The major component is the self-attention module. It forgos the recurrence in LSTM module in favor of simple scaled dot-product multiplication between the representations. The input is viewed as set of Key-Value pairs (K,V) and the previous output (or context) is encoded as query (Q). Under this methodology , the output is produced by combining mapping the query with keys and values and is called answers. Succinctly,

$$Attention(K, Q, V) = \text{softmax}\left(\frac{Q \cdot K}{\sqrt{n}}\right) V \tag{7.1}$$

The module used in our network has been inspired from the work of [52, 53, 54]. The self-attention is inserted between the two LSTM. Self attention module works on the feature map hence gains the advantage of working on the higher spatial semantics. So the global average pooling layer is removed after the last convolution layer. The feature map of last convolution layer forms the input to the self-attention module. The self -attention module hast takes feature map and previous hidden state of LSTM and outputs the input for next LSTM. This is different than directly passing the same patch to all time steps of LSTM. The placement of self-attention module is show in figure 7.1.

The figure 7.2 shows the detailed version inside the self-attention module. The output of the local stream convolution neural network (used resnet34 here) forms the input and is computed only once. The same feature map is used at each step of self-attention. The output activation map $\mathbf{F} \in \mathbb{R}^{h \times w \times c}$ is split along channel representing keys $\mathbf{K} \in \mathbb{R}^{h \times w \times c_k}$ and values $\mathbf{V} \in \mathbb{R}^{h \times w \times c_v}$ such that $c = c_k + c_v$. To maintain the spatial representation, a spatial bias map of dimension $h \times w \times c_s$ is added to both key and value. The query vectors are the function of the hidden states of the LSTM cell only. The hidden states of LSTM is passed through the two hidden layer MLP of dimension 512 to output a D dimensional vector of size $c_k + c_s$. The output represents a query vector. This query vector is dot product with the key feature map along the channel dimension and sum along the channel to produce a two dimensional feature map of size $h \times w$. The output is spatially softmax which forms the attention map. The attention map is then broadcast along the channel dimension to have channel of size $c_s + c_v$ and element-wise multiplied with the value feature map. The resultant feature map is spatially sum to output a vector of dimension $c_v + c_s$. This is called answer vector. Next, another two layer MLP is used to output the 512 dimension vector. This forms the input to the next time-step LSTM. The whole network makes use of "soft" attention and hence can be trained end-to-end via backpropogation.

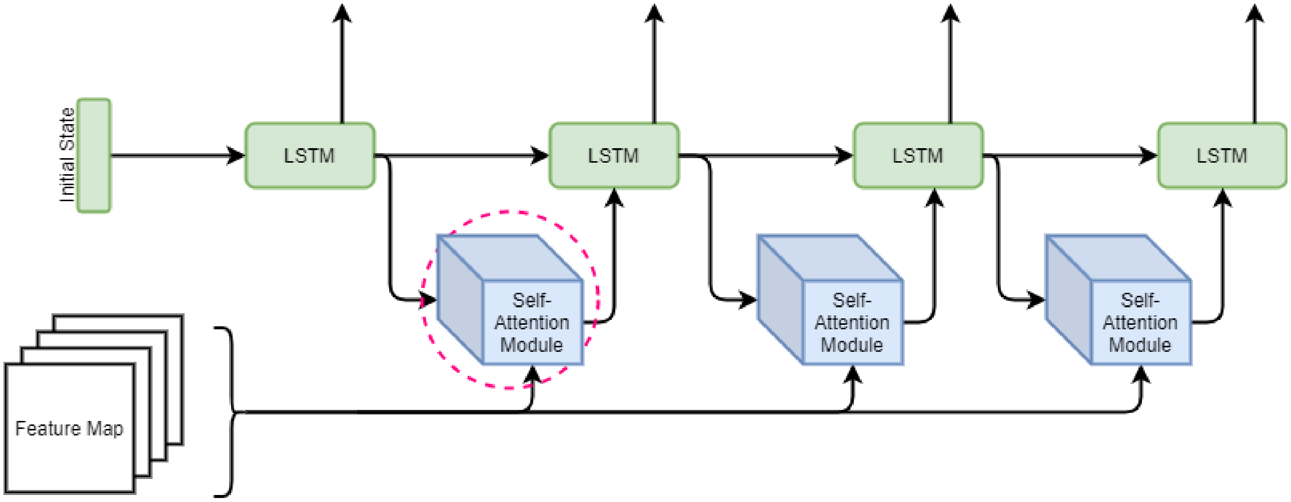


Figure 7.1: Figure shows the placement of self-attention module with the recurrent LSTMs.The feature map represents output from last convolution layer of local-stream.

We believe that the self-attention will aid in refining some of those visual marginal difference and provide better representative fine grained features. Some of the examples of use of self-attention module before and after attention is shown in figure 7.3.

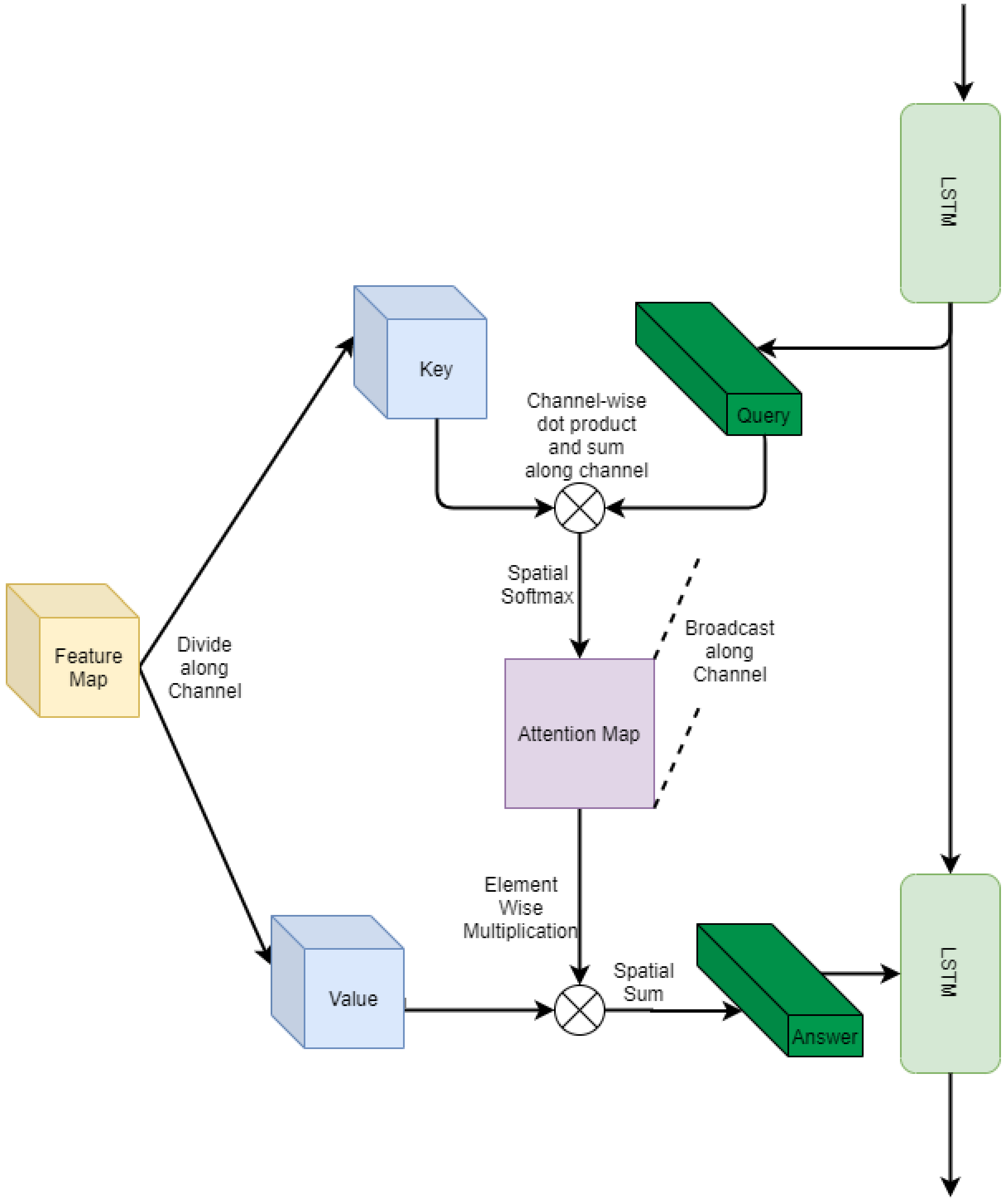
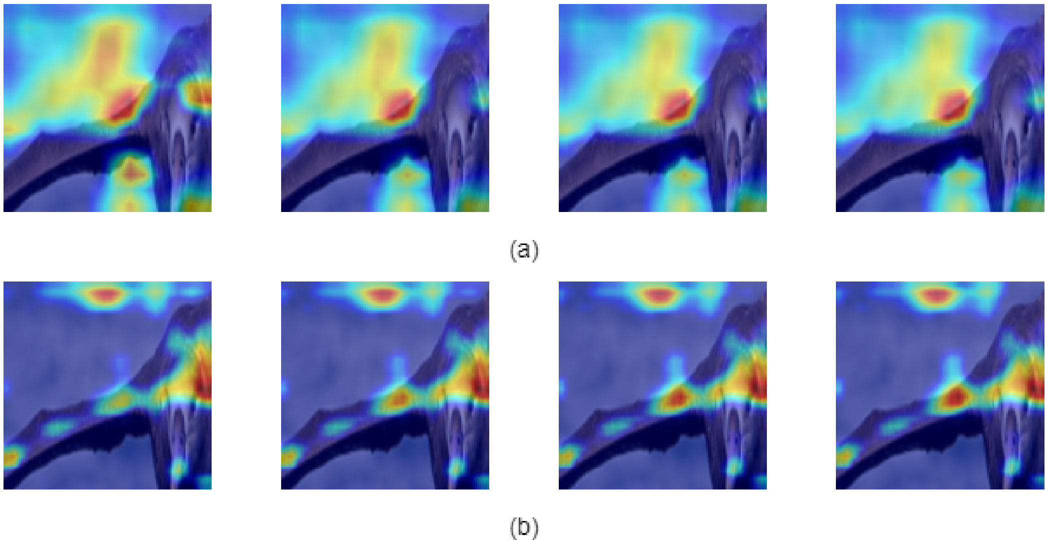


Figure 7.2: The figure shows the zoomed details of the self-attention block. The green blocks shows the MLP layer to produce query vectors and answer vectors. The Key and Values comes from feature map.



(a)

(b)

Figure 7.3: The figure shows an example where self-attention improves the attention area. It denotes the grad-cam visualization of hidden representation of LSTMs (a) without self-attention module and (b) with self-attention.

8. FUTURE WORK

Currently, we use the output of a weakly-supervised framework to crop out the patches from the images. Since our network highly relies on the patches to be representative parts of the image. Our network's efficiency is limited by the weakly supervised framework used. So, in future, we try to embed the localization network with our network. On the other front, we believe the addition of self-attention is a good start for our future work. Also, the network does a lot of dot product based computation hence increasing the computation time and memory. Further work is needed to fine-tune it to achieve higher accuracy with lower computation requirements.

9. CONCLUSION[*]

In this work, 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 produces fine-grained patches and the soft attention module attends to the most discriminative scale by providing more weight to important features and finally 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 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.

[*] Part of this section is adapted from the published paper, "Focus Longer to See Better: Recursively Refined Attention For Fine-Grained Image Classification" by Prateek Shroff, Tianlong Chen, Yunchao Wei, and Zhangyang Wang, Conference on Computer Vision and Pattern Recognition (CVPR) Workshops ©2020 IEEE.

REFERENCES

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

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

[3] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 91–99, 2015.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision (ECCV)*, pp. 21–37, Springer, 2016.

[7] F. Sadeghi, S. K. Kumar Divvala, and A. Farhadi, "Viske: Visual knowledge extraction and question answering by visual verification of relation phrases," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1456–1464, 2015.

[8] W. Chen, X. Gong, X. Liu, Q. Zhang, Y. Li, and Z. Wang, "Fasterseg: Searching for faster real-time semantic segmentation," *arXiv preprint arXiv:1912.10917*, 2019.

[9] D. Lin, Y. Ji, D. Lischinski, D. Cohen-Or, and H. Huang, “Multi-scale context intertwining for semantic segmentation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 603–619, 2018.

[10] J. Fu, H. Zheng, and T. 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 (CVPR)*, pp. 4438–4446, 2017.

[11] H. Zheng, J. Fu, T. Mei, and J. Luo, “Learning multi-attention convolutional neural network for fine-grained image recognition,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 5209–5217, 2017.

[12] W. Ge, X. Lin, and Y. 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 (CVPR)*, pp. 3034–3043, 2019.

[13] T.-Y. Lin, A. RoyChowdhury, and S. Maji, “Bilinear cnn models for fine-grained visual recognition,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1449–1457, 2015.

[14] Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin, and S. Belongie, “Kernel pooling for convolutional neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2930, 2017.

[15] S. Cai, W. Zuo, and L. Zhang, “Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 511–520, 2017.

[16] D. Wang, Z. Shen, J. Shao, W. Zhang, X. Xue, and Z. Zhang, “Multiple granularity descriptors for fine-grained categorization,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2399–2406, 2015.

[17] X. Liu, T. Xia, J. Wang, and Y. Lin, “Fully convolutional attention localization networks: Efficient attention localization for fine-grained recognition,” *arXiv preprint arXiv:1603.06765*, vol. 1, no. 2, p. 4, 2016.

[18] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The Caltech-UCSD Birds-200-2011 Dataset,” Tech. Rep. CNS-TR-2011-001, California Institute of Technology, 2011.

[19] A. Khosla, N. Jayadevaprakash, B. Yao, and L. 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 (CVPR)*, (Colorado Springs, CO), June 2011.

[20] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, “3d object representations for fine-grained categorization,” in *4th International IEEE Workshop on 3D Representation and Recognition (3dRR)*, (Sydney, Australia), 2013.

[21] J. Fu, T. Mei, K. Yang, H. Lu, and Y. Rui, “Tagging personal photos with transfer deep learning,” in *Proceedings of the 24th International Conference on World Wide Web*, pp. 344–354, 2015.

[22] J. Fu, Y. Wu, T. Mei, J. Wang, H. Lu, and Y. Rui, “Relaxing from vocabulary: Robust weakly-supervised deep learning for vocabulary-free image tagging,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1985–1993, 2015.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems (NIPS)*, pp. 1097–1105, 2012.

[24] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell, “Compact bilinear pooling,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 317–326, 2016.

[25] S. Kong and C. Fowlkes, “Low-rank bilinear pooling for fine-grained classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 365–374, 2017.

[26] P. Li, J. Xie, Q. Wang, and Z. 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 (CVPR)*, pp. 947–955, 2018.

[27] F. Perronnin and D. Larlus, "Fisher vectors meet neural networks: A hybrid classification architecture," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3743–3752, 2015.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, 2017.

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[32] M. Moghimi, S. J. Belongie, M. J. Saberian, J. Yang, N. Vasconcelos, and L.-J. Li, "Boosted convolutional neural networks.," in *British Machine Vision Conference (BMVC)*, pp. 24–1, 2016.

[33] P. Tang, X. Wang, A. Wang, Y. Yan, W. Liu, J. Huang, and A. Yuille, "Weakly supervised region proposal network and object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 352–368, 2018.

[34] S. Huang, Z. Xu, D. Tao, and Y. Zhang, "Part-stacked cnn for fine-grained visual categorization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1173–1182, 2016.

[35] P. Li, J. Xie, Q. Wang, and W. Zuo, “Is second-order information helpful for large-scale visual recognition?,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2070–2078, 2017.

[36] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, 2016.

[37] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. 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 (CVPR)*, pp. 842–850, 2015.

[38] M. Simon and E. Rodner, “Neural activation constellations: Unsupervised part model discovery with convolutional networks,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1143–1151, 2015.

[39] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su, “This looks like that: deep learning for interpretable image recognition,” in *Advances in Neural Information Processing Systems (NIPS)*, pp. 8928–8939, 2019.

[40] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

[41] A. Mahendran and A. Vedaldi, “Understanding deep image representations by inverting them,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5188–5196, 2015.

[42] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *International Conference on Machine Learning (ICML)*, pp. 2048–2057, 2015.

[43] V. Mnih, N. Heess, A. Graves, *et al.*, “Recurrent models of visual attention,” in *Advances in Neural Information Processing Systems (NIPS)*, pp. 2204–2212, 2014.

[44] J. C. Caicedo and S. Lazebnik, “Active object localization with deep reinforcement learning,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2488–2496, 2015.

[45] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, “Top-down neural attention by excitation backprop,” *International Journal of Computer Vision (IJCV)*, vol. 126, no. 10, pp. 1084–1102, 2018.

[46] W. Ge, S. Yang, and Y. Yu, “Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1277–1286, 2018.

[47] J. Rony, S. Belharbi, J. Dolz, I. B. Ayed, L. McCaffrey, and E. Granger, “Deep weakly-supervised learning methods for classification and localization in histology images: a survey,” *arXiv preprint arXiv:1909.03354*, 2019.

[48] P. Tang, X. Wang, A. Wang, Y. Yan, W. Liu, J. Huang, and A. Yuille, “Weakly supervised region proposal network and object detection,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 352–368, 2018.

[49] M. Lin, Q. Chen, and S. Yan, “Network in network,” *arXiv preprint arXiv:1312.4400*, 2013.

[50] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” *arXiv preprint arXiv:1805.08318*, 2018.

[51] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7794–7803, 2018.

[52] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, “Stand-alone self-attention in vision models,” *arXiv preprint arXiv:1906.05909*, 2019.

[53] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, “Stand-alone self-attention in vision models,” *arXiv preprint arXiv:1906.05909*, 2019.

[54] A. Mott, D. Zoran, M. Chrzanowski, D. Wierstra, and D. J. Rezende, “Towards interpretable reinforcement learning using attention augmented agents,” in *Advances in Neural Information Processing Systems (NIPS)*, pp. 12329–12338, 2019.