Towards Fine-grained Recognition: Joint Learning for Object Detection and Fine-grained Classification

Anonymous Authors

Anonymous Institutes

Abstract. Fine-grained classification is a challenging problem due to subtle differences between intra-class categories. In practice, fine-grained classification is often used in conjunction with object detection algorithms to locate and identify object categories. Despite recent achievements in both fine-grained classification and object detection, few works have demonstrated datasets or solutions to simultaneously handle both tasks. We make two contributions to this problem. Firstly, we construct a fine-grained classification and detection benchmark. Secondly, we show an end-to-end convolutional neural networks (CNNs) architecture to detect and classify fine-grained objects. Experimental results verify that our networks perform favorably against alternatives.

Keywords: Object Detection · Fine-grained Classification

1 Introduction

Locating and identifying objects is important for many computer vision applications. For example, one could develop automated computer vision software to process live surveillance videos and recognize brand and model of a vehicle for identifying traffic violations. For environmental monitoring, locating and recognizing wildlife could generate statistics to help protecting endangered species. However, even for car enthusiasts or bird-watching experts, it is still difficult to identify a specific car model or bird species accurately.

Recent advances in deep learning have shown promising results for image classification and detection. PASCAL VOC [3] and ImageNet [11] are widely used to evaluate classification/detection performance. Both datasets provide annotations for generic object classification and detection. In contrast to generic object classification, fine-grained classification aims at identifying objects within the same fine-grained category (or subcategory). Stanford Cars [10] and Caltech-UCSD Birds 200 (CUB-200) [24] are the two most popular benchmarks for evaluating such tasks. Comparing to the generic classification task, images in fine-grained datasets usually exhibit small inter-class and large intra-class variations in visual appearances. The small inter-class variation is due to the natural of fine-grained classification task, where all objects belonging to the same category share similar appearances. The large intra-class variation is introduced by the dataset, where objects are often presented in close-up photos with a combination of pose, viewpoint, illumination and background changes. A simple change in the camera perspective may lead to dramatic visual differences which could easily fool a neural network model trained on generic classification datasets.

While fine-grained datasets composed of close-ups are often geometrically warped comparing to photos from generic classification datasets, it enables learning of features more robust to variations in camera perspective and pose. Humans are able to rapidly identify the model of a car from key visual features such as the shape of the taillight or logo [27]. However, CNNs struggle on learning the fine details because it only learns the object appearance and lacks understanding of keypoint location, pose and geometry [7]. Recent work has shown that CNNs have remarkable capabilities to learn geometry-related information in convolutional layers [31], but final fully-connected layers weaken this ability and tend to keep only category-level information. Based on this finding, researchers propose several two-stage object detectors [20, 4] to use shared convolutional layers to find object proposals and output final predictions.

This leads to an open question: can we utilize a single network to learn both object-level localization information as well as stable fine-grained features invariant to imaging conditions and pose deformations? We address this question by creating a unified framework to perform *fine-grained object recognition*, subsuming the problems of object detection and fine-grained classification. Although many deep learning methods have been proposed for each task, we are not aware of any framework that directly train on both fine-grained classification and generic object detection datasets. By learning both spatial locations and intra-class diversities of an object, we enable the network to produce quality feature vectors with high distinctiveness. Additionally, as most convolutional layers in our network are shared for both tasks, we introduce very little computational overhead to achieve combined goals of efficiency and accuracy.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 details our overall framework based on the proposed method. We show experimental results in Section 4 and conclude in Section 5.

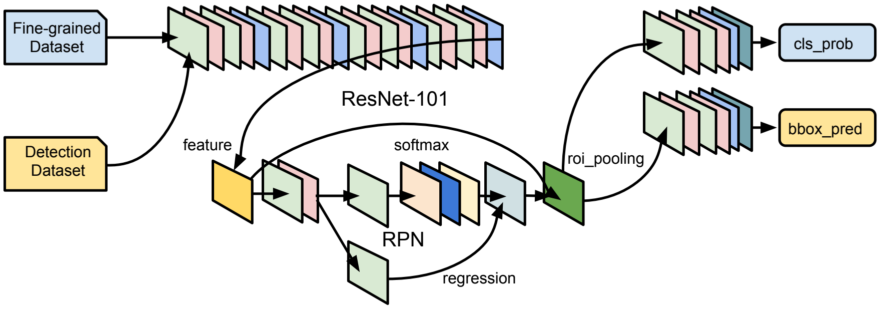


Fig. 1: Architecture of our network. Our network is based on Faster-RCNN and branches after ROI Pooling layer for fine-grained classification.

2 Related Work

Visual Object Classification: Visual object classification aims at predicting class labels from a given image. Deep convolutional neural networks (DNNs) have led to tremendous success in visual recognition [11, 5]. Deep networks naturally enforce features from different levels and can be trained easily in an end-to-end fashion to reach high accuracy. Highway Networks [22] were the first to introduce bypassing paths to tackle the gradient vanishing problem. ResNet further utilizes identity mapping as bypassing paths and achieved remarkable performance in benchmark datasets such as ImageNet and Microsoft COCO [5]. Experiments have shown that ResNet is able to converge on as many as 1000 layers. In the mean time, researchers also attempt to make the network wider. [23] proposes wider residual blocks to achieve better accuracy.

Fine-grained Classification: Fine-grained recognition is the process of identifying objects within the same category. Various fine-grained classification datasets have been proposed, including bird species classification [9] and recognition of car brand, year and model [10, 27]. [28] was among the first to employ deep object detection networks for fine-grained classification. [6] leverages a convolutional networks to locate multiple object parts and a two-stream classification network to encode object and part level information. Spatial transformer networks [7] explicitly allow spatial manipulation of training data, giving neural networks the ability to actively transform the region of interest. Other research directions include attention models and feature pooling. Attention models aim at focusing the network on only a few distinctive image parts/keypoints [30] while feature pooling collects second-order or higher-order statistics to form a more distinctive feature vector for better classification results [13].

Object Detection: Object detection aims at finding locations of object instances in a scene. Recent work shows CNN has sufficient power to learning geometric representations to predict both the class label and geometric information of an object [31, 17, 20, 4]. Modern detectors can be categorized as one-stage or two-stage frameworks. The one-stage detection framework [14, 18] is free of object proposals and can be trained in an end-to-end fashion. At test time, the entire network is only evaluated once, achieving real-time performance. The two-stage detection framework [20, 4], on the other hand, includes a class-agnostic region proposal generator and a classifier. Generally speaking, one-stage frameworks exhibit better efficiency and run at a higher frame rate, while two-stage frameworks are more accurate and are capable of locating smaller objects. Also, there is a growing interest in converting these frameworks into more compact versions for real-time/embedded systems [12].

Weakly-supervised Object Localization: The recent progress of deep learning is largely due to advances in high-power computing hardware and the availability of large-scale, high-quality annotated datasets. However, the annotation of ground truth labels is expensive and labor-intensive, and sometimes even impossible considering the scale of todays massive visual data. Therefore, it is important to develop unsupervised or weakly-supervised approaches to enable continuous learning. Researchers have investigated weakly-supervised object localization by studying maximal activations in the network layers [17, 31]. Our work attempts to learn from both detection and classification benchmarks without full annotation.

Fig. 2: Randomly selected sample images from our FGR-4K benchmark. Compared to the Stanford Cars [10] dataset, our benchmark contains more real-life images with complex background.

3 Approach

3.1 Network Architecture

In [20], Ren *et al.* designed the Faster-RCNN network which is composed of a Region Proposal Network (RPN) and a backbone CNNs. To improve efficiency, they also combined the RPN and the CNNs into a single network with a large number of shared layers. Faster-RCNN exhibits excellent tradeoff of speed and accuracy. It runs at near real-time speed (5fps) on a single GPU, while achieving state-of-the-art detection accuracy. Although Faster-RCNN gives perfect results on detection benchmarks, it can not be directly applied to the task of learning both detection and fine-grained classification. Therefore, we build upon the Faster-RCNN network architecture and made a few modifications. Firstly, we use ResNet-101 [5] as the backbone CNNs as it is deeper and gives better accuracy compared to the ZF or VGG16 network used in the original paper. Next, we made a new branch after the ROI Pooling layer for predicting fine-grained categories. Therefore, apart from the original RPN stream which predicts bounding box coordinates, the additional steam simultaneously predicts fine-grained class labels. Our network structure is illustrated in Fig. 1.

3.2 Fine-grained Recognition Benchmark

Existing fine-grained classification benchmarks such as Stanford Cars [10] only contain one object per image. Also, a large portion of the dataset is composed of stock images with clean/white background. To provide a fair technical benchmark for evaluating fine-grained classification and object detection performance, We introduce a new dataset called FGR-4K (Fine-grained Recognition 4K). We plan to open-source this dataset for research purposes. Our dataset is annotated using the same labels from the Stanford Cars dataset (see Fig. 2). This dataset is constructed for testing only, as training and validation data could be obtained from Stanford Cars or PASCAL VOC car

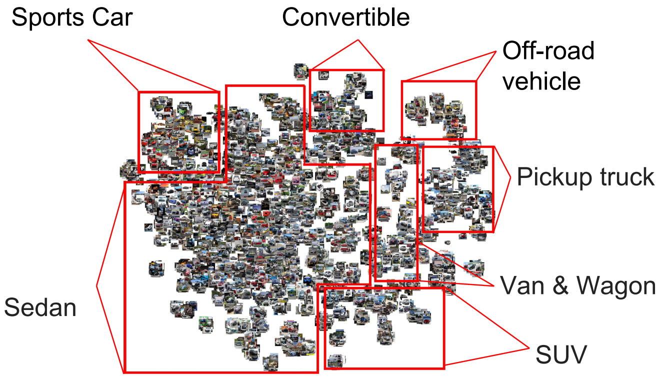


Fig. 3: t-SNE visualization [16] of features extracted from the joint model on Stanford Cars test set (Best viewed zoomed in and in color). The similarities are calculated purely based on visual feature embeddings. This illustrates that our joint model is able to preserve fine-grained semantics information after dimensionality reduction.

Algorithm 1 Training Strategy

```
Parameters: $\mathcal{M} \leftarrow$ CNNs Model
$\mathcal{D} \leftarrow$ Dataset

procedure TrainingStrategy($\mathcal{D}_{fg+det}$)
    $\mathcal{M}_{fg} \leftarrow$ LearningPolicy($\mathcal{M}_{ImageNet}, \mathcal{D}_{fg}$)
    $\mathcal{M}_{det} \leftarrow$ LearningPolicy($\mathcal{M}_{ImageNet}, \mathcal{D}_{det}$)
    $\mathcal{M}_{joint} \leftarrow$ netsurgery($\mathcal{M}_{fg}, \mathcal{M}_{det}$)
    $\mathcal{M}_{joint} \leftarrow$ LearningPolicy($\mathcal{M}_{joint}, \mathcal{D}_{fg+det}$)
    return $\mathcal{M}_{joint}$
end procedure
```

class. Our dataset provides 3818 images crawled from Google image search API. These images are filtered to include only those permitted for commercial reuse. We also run automated deduplication, white background detection and text detection algorithms to remove images not suitable for annotation. The deduplication is done by removing images with the same SHA256 checksum on RGB values. White background is identified by converting the image to a binary map using 33% above its median pixel value as the threshold. If white pixels occupy more than 50% of the image then we will remove it from the candidate set. We use the open source text detection library [2] to remove images detected with any words or letters. After the automatic filtering, we manually clean up the dataset to choose only real-life images with complex background. Next, we send the candidate set to human annotators to draw rectangles around all car objects that fall into the given 196 fine-grained categories. Compared to the Stanford Cars dataset, our benchmark contains more real-life images with complex backgrounds which are challenging for both fine-grained classification and object detection tasks.

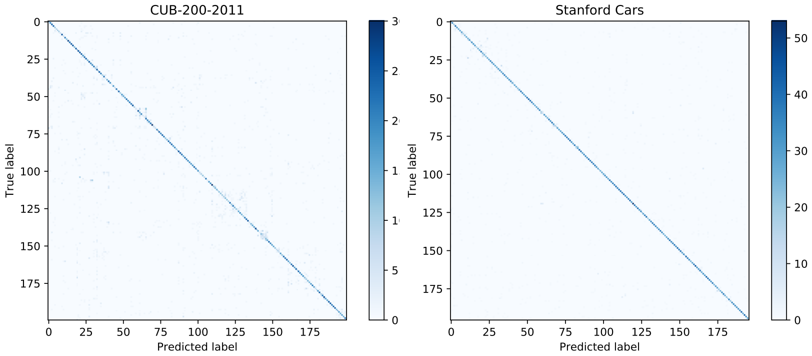


Fig. 4: Confusion matrix on the CUB-200 and Stanford Cars datasets. The vertical axis shows the groundtruth labels while the horizontal axis shows the predicted labels. Note that our model makes more false positive predictions on the CUB-200 dataset compared to the Stanford Cars dataset. This is because CUB-200 contains non-rigid transformations with larger intra-category variance on object pose and appearance.

3.3 Training Strategy

Because of the high diversity in fine-grained classes, we use on-the-fly data augmentation during training. The augmentation includes random cropping, resizing and rotation up to 30 degrees. We also included random smooth filtering and JPEG compression varying from 50% to 90% quality. The online augmentation is only applied to training while the validation/testing accuracy is still reported on the original dataset. Due to the difficulty in training initial weights for the whole network, we first train a ResNet model on the fine-grained dataset and a Faster-RCNN model on the detection dataset separately. Note that these two networks are using the same ResNet-101 backbone and the only difference between them is the RPN network inserted between Res4b and Res5a layers. Once both models converge, we freeze all layers in the detection model by setting learning rate to 0 for all bottom layers below the ROI Pooling layer. Next, we append all top layers above res4b from the fine-grained model to the detection model as a separate branch for finetuning. During training, we define the maximum number of iterations as max_iter and iterations per step as $step\_size$. The learning rate will drop by a factor of 10 (*i.e.* multiplied by lr_decay where $lr\_decay = 0.1$). Once the step size is reached, the learning rate decreases by 10. The accuracy jumps when learning rate changes. This is because the solver has been optimizing at a certain learning rate for a certain number of iterations to find the local optimum. The weight of the whole model stabilizes for the duration of a consistent learning rate. After the learning rate reduces, it is easier for the neural networks to capture fine details and increase accuracy. Assuming the initial warm-up learning rate to be lr_init, when we reach the max_iter the learning rate will be $lr\_init \times lr\_decay^{\max\_iter/step\_size}$. When the accuracy saturates, we will only fine-tune softmax layers to obtain the joint model. We illustrate the details of our training strategy in Algo. 1.

4 Experiments

4.1 Initial Training

We use Caltech-UCSD Birds-200-2011 (CUB-200-2011) [24] and Stanford Cars dataset [10] for fine-grained classification experiments. The CUB-200-2011 dataset contains 200 fine-grained bird categories with 11788 images. The Stanford Cars dataset contains 16185 images with 196 classes of cars including year, make and model. Details of the datasets are shown in Table. 1. For both datasets, We fine-tune on the ImageNet ResNet-101 model. The model is trained with the base learning rate of 0.01, gamma of 0.5, momentum of 0.9 and weight decay of 0.0001. Next, we start fine-tuning this model. Once the training accuracy is saturated, we fix all bottom layer weights and only fine-tune the softmax layer. The final accuracy is 81.0%. A comparison to related work is shown in Table 1.

For object detection, We build our approach based on the state-of-the-art Faster-RCNN [20] detection framework. We start training the modified Faster-RCNN network explained in Sect .3 with ResNet-101 backend. The weights of ResNet-101 is initialized with pre-trained ImageNet weights. We train our model with base learning rate of 0.001, gamma of 0.1, momentum of 0.9 and weight decay of 0.0005. The training is performed on the combined PASCAL VOC 07+12 dataset on a single class (car or bird). After 1000K iterations the tested mAP is 71.99% on VOC 07 and 77.85% on VOC 12 for the bird class. We repeat the same procedure for the car class on VOC 07 and 12 and the final mAP for the car class is 75.15% and 78.06%, respectively. Quantitative evaluations can be found at Table. 1. Note that our accuracy is reported on images only containing the bird or car class with all other annotations removed, while the accuracy reported by all other algorithms still consider classes other than bird or car. Qualitative results are shown in Fig. 7.

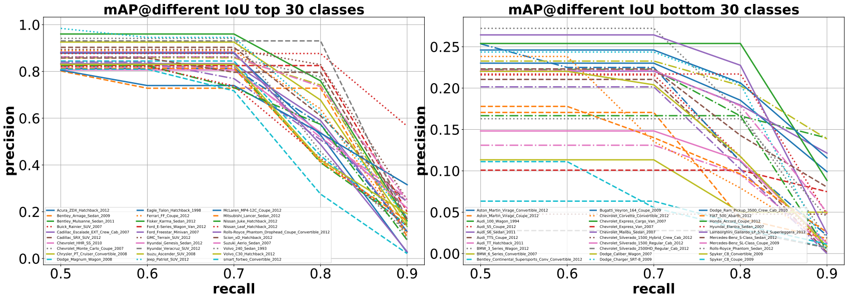


Fig. 5: Precision-recall curve of our joint model on the FGR-4K dataset (Best viewed electronically). Left: PR curve on the top-performing 30 classes. Right: PR curve on the lowest-performing 30 classes. Note that our model is robust to stricter IoU criterias and the performance starts to degrade when IoU is increased to more than 0.7.

Classification Accuracy (%)				Detection MAP (%)				
Bird (CUB-200)		Car (Stanford)		Bird (VOC)		Car (VOC)	Train	Test
Xiao *et al.* [25]	69.7	Krause *et al.* [9]	**92.8**	Faster-RCNN (VGG16)[20]	70.9	**84.7**	0712	07
Simon *et al.* [21]	81.0	Lin *et al.* [13]	91.3	Faster-RCNN (ResNet50) [20]	74.3	75.9	0712	12
Kong *et al.* [8]	84.2	Zhang *et al.* [29]	88.4	SSD300[14]	70.5	76.1	0712	07
Liu *et al.* [15]	**85.4**	Xie *et al.* [26]	86.3	YOLO[18]	57.7	55.9	0712	12
				YOLOv2 544[19]	74.8	76.5	0712	12
Ours initial	81.0		90.1		72.0	75.2	0712	07
Ours Joint	72.6		86.2		**77.9**	78.0	0712	12

Table 1: Results on Fine-grained Classification and Object Detection benchmarks. Our methods handles both tasks at the same time, while performs favorably against alternative methods designed specifically for each task.

4.2 Joint Training

Now that we obtained models for both detection and fine-grained classification, we start merging the models for joint training. During inference time, the network will produce bounding box locations in an image, as well as fine-grained class labels for each bounding box. We freeze all layers in the detection model by setting learning rate to 0 for all bottom layers below the roi_pool5 layer. Next, we append all top layers above res4b from the fine-grained classification model to roi_pool5 layer in the detection model. Because of weight discrepancies between the original bottom layers and the new bottom layers trained on the object detection dataset, the test accuracy of the joint model on CUB-200-2011 drops from 81.0% to 67.5%. After joint training for another round (1000K iterations, freezing bottom layers) the fine-grained classification accuracy goes up to 72.6%. We performed the same net-surgery and training procedures for the car class. The final classification accuracy for cars is 86.2%. The detection accuracy stays the same since we are only training the fine-grained branch with all other weights fixed. We found that this training schedule leads to the best results comparing to alternative approaches such as adjusting weights for all layers. The whole training process is illustrated in Algorithm 1. Next, we apply our joint model to the FGR-4K benchmark dataset. We show the precision-recall curve for the best and worst performing 30 categories in Fig. 5. This is done by varying the IoU threshold from 0.5 to 0.9 and recalculate mAP scores for all classes. We also compare our model with the commercial vehicle recognition service provided by Sighthound [1] on the FGR-4K dataset. Since Sighthound API only returns vehicle brand and model, we remove the year info from both the groundtruth and predicted results on the FGR-4K dataset for comparison. The final average mAP of our method is 62.59% while the average mAP of Sighthound is 56.88%. We show True /False predictions for each fine-grained class in Fig. 6. Note that the production Sighthound model is possibly trained on a enlarged dataset which includes more vehicle models than the FGR-4K dataset.

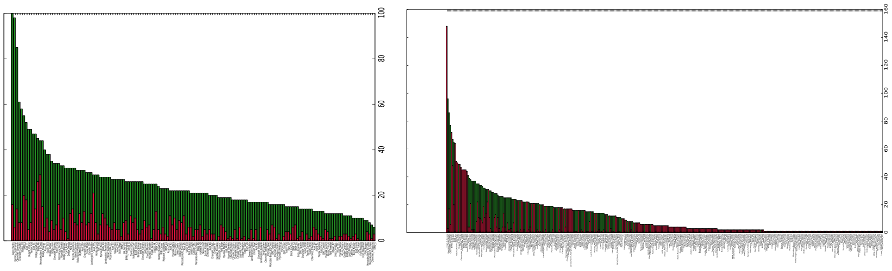


Fig. 6: mAP and TP/FP scores of our joint model on the FGR-4K dataset (Best viewed electronically). Left: True (green)/False (red) predictions using our approach. Right: True (green)/False (red) predictions obtained by Sighthound Cloud API for vehicle recognition [1]. The average mAP of our approach is 62.59% while the average mAP of the Sighthound Cloud API is 56.88 %. Note that our model predicts less false alarms comparing to the Sighthound production model.

.

4.3 Analysis

For the fine-grained classification experiments, we show the confusion matrix on the CUB-200 and Stanford Cars datasets in Fig. 4. The CUB-200 contains objects with non-rigid transformations and is more challenging compared to the Stanford Cars dataset. Also, according to Table. 1, there is a larger gap between the classification accuracy of our model and attention-based models [15, 9] on CUB-200 compared to Stanford Cars. Furthermore, our joint model suffers more accuracy degradation on the CUB-200 dataset compared to the Stanford Cars dataset. Despite these limitations, our model is able to learn from two vastly distinctive datasets and demonstrate competitive performance compared to methods developed for each task. We apply t-SNE visualization [16] to features extracted from the joint model on Stanford Cars test set and visualize the embeddings in Fig. 3. This illustrates that our joint model is able to preserve fine-grained semantics information in the high-dimensional space. The t-SNE visualization also indicates that our learnt features are able to capture visual similarities but is less sensitive to pose variations. For the FGR-4K dataset, we evaluate the detector performance against varying IoU thresholds in Fig. 5. As can be seen from the figure, the mAP per category only starts decreasing when IoU is more than 0.7. This shows that our detector is robust to stricter evaluation criterias, which is generally more desirable for real life vision applications. We also notice that the best-performing class labels are mostly composed of visually distinctive car models from different manufacturers, while the lowest-performing classes are more often from the same manufacturer with similar car-model names. This implies that the current joint network is good at detecting objects but is still having difficulties capturing small partial details within the object.

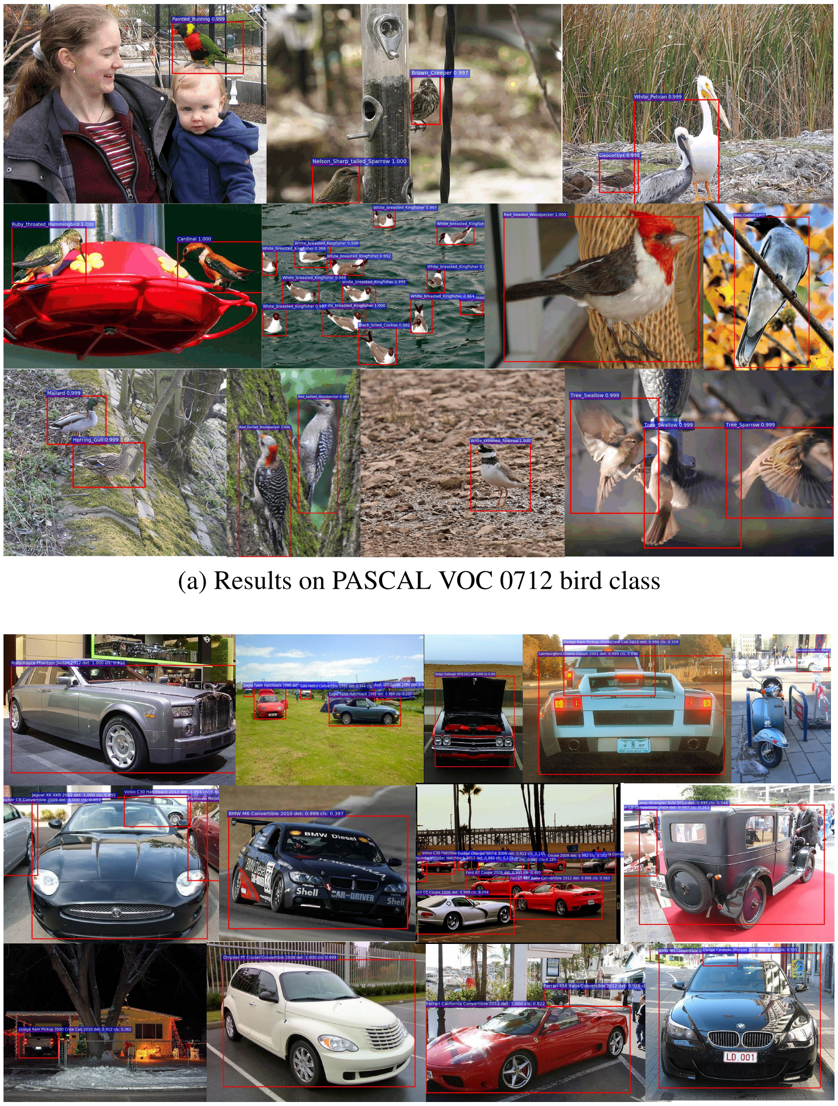
We show qualitative results in Fig. 7 (a) by running forward-inference on this joint model. Note that our model is able to predict fine-grained class labels not present in the PASCAL VOC dataset. As shown in Fig. 7 (a), our joint network predicts accurate bounding box locations for all bird objects in an image. In addition, our network is good at recognizing subtle color (e.g. Red headed woodpecker in row 2 column 3, Gray Catbird in row 2 column 4, Red bellied woodpecker in row 3 column 2 and White crowned sparrow in row 3 column 3) and shape variations (e.g. difference between Mallard and Herring Gull in row 3 column 1). In Fig. 7 (b), the model is able to recognize subtle differences between two similar-looking cars with the same color (e.g. ”Jaguar XK XKR 2012” in row 2 column 1 vs. ”BMW M6 Convertible 2010” in row 3 column 4) and partial object with occlusion (e.g. ”Mercedes Benz SL Class Sedan 2012” in row 1 column 5). For object classes not present in the Stanford Cars dataset, the network is able to assign a label with a closest visual match (e.g. ”Dodge Challenger SRT8 2011” in row 1 column 3).

5 Conclusion

In this work we have presented a framework to detect and classify fine-grained objects. To evaluate performance, we have created a new benchmark for fine-grained recognition. Experiments show that our approach performs favorably against competitive methods. In summary, our network structure provides more desirable characteristics for practical computer vision applications and reaches good balance between the model size, computational complexity and accuracy. Our system can be used to add visual intelligence to mobile devices. This feature is particularly useful for ornithologists or car enthusiasts who wish to identify or search for a particular object of interest. In the future, we plan to leverage post-training quantization techniques to compress our joint model and enable fast forward-inference on mobile apps.

References

1. Sighthound cloud api for vehicle recognition. https://www.sighthound.com/products/cloud
2. Tesseract open source ocr engine. https://github.com/tesseract-ocr/tesseract
3. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010)
4. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
6. Huang, S., Xu, Z., Tao, D., Zhang, Y.: Part-stacked cnn for fine-grained visual categorization. In: CVPR. pp. 1173–1182 (2016)
7. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: NIPS. pp. 2017–2025 (2015)
8. Kong, S., Fowlkes, C.: Low-rank bilinear pooling for fine-grained classification. In: CVPR. pp. 7025–7034. IEEE (2017)
9. Krause, J., Jin, H., Yang, J., Fei-Fei, L.: Fine-grained recognition without part annotations. In: CVPR. pp. 5546–5555 (2015)



Painted_Bunting 0.999
Brown_Creeper 0.997
Nelson_Sharp_tailed_Sparrow 1.000
White_Pelican 0.999
Geococcyx 0.998
Ruby_throated_Hummingbird 1.000
Cardinal 1.000
White_breasted_Kingfisher 0.997
White_breasted_Kingfisher 0.998
White_breasted_Kingfisher 0.966
White_breasted_Kingfisher 0.992
White_breasted_Kingfisher 0.988
White_breasted_Kingfisher 0.999
White_breasted_Kingfisher 0.997
White_breasted_Kingfisher 1.000
White_breasted_Kingfisher 0.964
Black_billed_Cuckoo 0.992
Red_headed_Woodpecker 1.000
Gray_Catbird 1.000
Mallard 0.999
Herring_Gull 0.999
Red_bellied_Woodpecker 0.999
Red_bellied_Woodpecker 1.000
White_crowned_Sparrow 1.000
Tree_Swallow 0.999
Tree_Swallow 0.999
Tree_Sparrow 0.999
Rolls-Royce Phantom Sedan 2012 det: 1.000 cls: 0.910
Eagle Talon Hatchback 1998 det: 0.999 cls: 0.237
Geo Metro Convertible 1993 det: 0.944 cls:
Audi 100 Sedan 1994 det:
Dodge Challenger SRT8 2011 det: 0.998 cls: 0.190
Dodge Ram Pickup 3500 Crew Cab 2010 det: 0.996 cls: 0.359
Lamborghini Diablo Coupe 2001 det: 0.999 cls: 0.636
Jaguar XK XKR 2012 det: 1.000 cls: 0.892
Volvo C30 Hatchback 2012 det: 0.994 cls:
Spyker C8 Convertible 2009 det: 1.000 cls: 0.851
BMW M6 Convertible 2010 det: 0.999 cls: 0.397
Dodge Charger SRT-8 2009 det: 0.911 cls: 0.145
Hyundai Veloster Hatchback 2012 det: 0.865 cls: 0.175
Coupe 2009 det: 0.982 cls: 0.352
det: 0.980 cls: 0.285
Ford GT Coupe 2006 det: 0.995 cls: 0.485
Ferrari 458 Italia Convertible 2012 det: 0.998 cls: 0.563
Ford GT Coupe 2006 det: 0.999 cls: 0.294
Jeep Wrangler SUV 2012 det: 0.997 cls: 0.548
C8 Convertible 2009 det: 0.997 cls: 0.263
Dodge Ram Pickup 3500 Crew Cab 2010 det: 0.912 cls: 0.362
Chrysler PT Cruiser Convertible 2008 det: 1.000 cls: 0.999
Ferrari 458 Italia Convertible 2012 det: 0.976
Ferrari California Convertible 2012 det: 1.000 cls: 0.822
Dodge Caravan Minivan 1997 det: 0.825 cls: 0.535

(a) Results on PASCAL VOC 0712 bird class

(b) Results on PASCAL VOC 0712 car class

Fig. 7: Qualitative results on PASCAL VOC 0712 Bird and Car classes [3]. The labels are learnt from CUB-200-2011 [24] and Stanford Cars dataset [10], respectively.

10. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: 3dRR. Sydney, Australia (2013)
11. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: NIPS. pp. 1097–1105 (2012)
12. Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., Sun, J.: Light-head r-cnn: In defense of two-stage object detector. arXiv preprint arXiv:1711.07264 (2017)
13. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear cnn models for fine-grained visual recognition. In: ICCV. pp. 1449–1457 (2015)
14. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV. pp. 21–37. Springer (2016)
15. Liu, X., Wang, J., Wen, S., Ding, E., Lin, Y.: Localizing by describing: Attribute-guided attention localization for fine-grained recognition. In: AAAI. pp. 4190–4196 (2017)
16. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. JMLR pp. 2579–2605 (2008)
17. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: CVPR. pp. 685–694 (2015)
18. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. pp. 779–788 (2016)
19. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. arXiv preprint (2017)
20. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS. pp. 91–99 (2015)
21. Simon, M., Rodner, E.: Neural activation constellations: Unsupervised part model discovery with convolutional networks. In: ICCV. pp. 1143–1151 (2015)
22. Srivastava, R.K., Greff, K., Schmidhuber, J.: Highway networks. arXiv preprint arXiv:1505.00387 (2015)
23. Targ, S., Almeida, D., Lyman, K.: Resnet in resnet: generalizing residual architectures. arXiv preprint arXiv:1603.08029 (2016)
24. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-UCSD Birds 200. Tech. Rep. CNS-TR-2010-001, California Institute of Technology (2010)
25. Xiao, T., Xu, Y., Yang, K., Zhang, J., Peng, Y., Zhang, Z.: The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In: CVPR. pp. 842–850 (2015)
26. Xie, S., Yang, T., Wang, X., Lin, Y.: Hyper-class augmented and regularized deep learning for fine-grained image classification. In: CVPR. pp. 2645–2654 (2015)
27. Yang, L., Luo, P., Change Loy, C., Tang, X.: A large-scale car dataset for fine-grained categorization and verification. In: CVPR. pp. 3973–3981 (2015)
28. Zhang, N., Donahue, J., Girshick, R., Darrell, T.: Part-based r-cnns for fine-grained category detection. In: ECCV. pp. 834–849. Springer (2014)
29. Zhang, X., Zhou, F., Lin, Y., Zhang, S.: Embedding label structures for fine-grained feature representation. In: CVPR. pp. 1114–1123 (2016)
30. Zheng, H., Fu, J., Mei, T., Luo, J.: Learning multi-attention convolutional neural network for fine-grained image recognition. In: ICCV. vol. 6 (2017)
31. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. arXiv preprint arXiv:1412.6856 (2014)