# CS482/682 Final Project Report Group 21

Fine-Grained Image Classification (FGIC) Using a Feather Dataset

Logan Donaldson (ldonald3), Hashem AlSabi (halsabi1),
Samer Aslan (saslan1), Ryunosuke Saito (rsaito1)

May 4, 2022

## 1 Introduction

**Background:** The FGIC task requires distinguishing classes with subtle discriminatory features. Its applications span numerous domains including conservation [1], healthcare [2], and retail stores [3]. Our application is feather to species mapping which is useful for identifying birds prone to aircraft collisons [4].

**Related Work:** Approaches not explored in this paper include Recursive Attention CNNs (RA-CNNs) and part based image representation. The former recursively make predictions and pass forward smaller croppings of the input to attend to specific regions [5]. The latter seeks to segment high level features shared amongst classes to generate a representation for classification [6]. Specific loss functions have also been developed to discourage prediction over-confidence and enhance generalizability [7].

## 2 Methods

**Dataset:** The dataset consists of 9562 train, 2391 validation, and 2988 test images of feathers [4]. Associated with each image are two hierarchical labels, denoting order (16 classes) and species (100 classes). The latter is the classification goal. Images are reshaped to size 64x64 to reduce the required compute.

**Setup, Training and Evaluation:** The data pipeline was adapted from the dataset's creator's work [8]. FGIC data augmentation is difficult as even subtle changes may not be label preserving due to low inter-class variability. Thus, simple image flips and small color jitters to mimic different lighting conditions are used. Class sizes are heavily skewed, ranging from 383 to 43 images in the training set. To promote generalizability class-weighted cross entropy loss is used to more heavily penalize misclassifications of images belonging to underrepresented classes.

A VGG-16 model is used as a baseline. Modified VGG-16 models which used self-attention modules to either replace or be included in addition to the $4^{\text{th}}$, $6^{\text{th}}$, and $8^{\text{th}}$ convolutional layers were implemented using the following sources as a guide [9], [10]. The self-attention module was originally designed for GANs, though we show it provides marginal improvements in FGIC. The module allows the CNN to consider pairwise interactions across the entire image through a covariance matrix $\beta \in R^{N \times N}$, where $N$ is the product of the input feature map's width and height. In this way self-attention increases receptive field. Note the use of a skip-ahead connection to improve gradient flow.

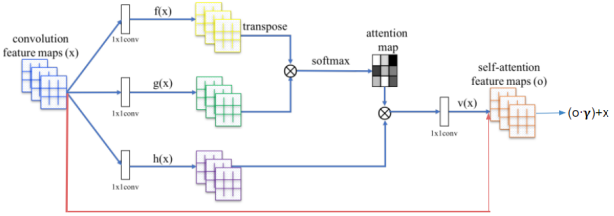


Figure 1: Attention Module; Fig. Adapted From [9]

A Bilinear CNN (BCNN) model which combines feature maps of two parallel VGG-16 models truncated at the $7^{\text{th}}$ convolutional layer via outer prod-

uct was implemented using the following sources as a guide [11],[12],[13]. Notably, the outer product results in a matrix $M \in R^{C \times C}$ where $C$ is the number of channels. $M$ is a channel-wise covariance matrix that considers second-order feature interactions but lacks any spatial information, making it orderless. A $1 \times 1$ convolutional reduction layer is used to reduce the classifier's width.

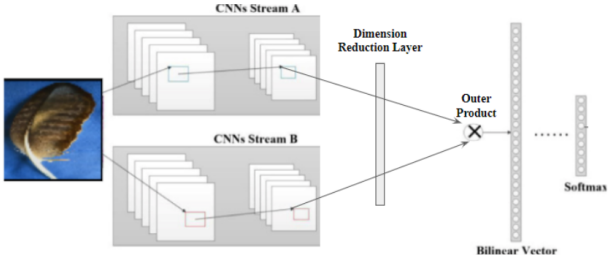


Figure 2: BCNN; Fig. Adapted From [11]

We designed a custom amendment to the BCNN to leverage hierarchical labeling. By appending classifiers, Stream A is trained to predict **orders** and Stream B to predict **species**. The streams' final feature maps are combined via outer product and used to make the final species prediction. Thus, we have three loss functions whose average is used as the computational graph's head for backpropagation. The intuition is to contextualize the feature maps by considering the species' order in the predictive input. Note the streams are now full VGG16 feature extractors.

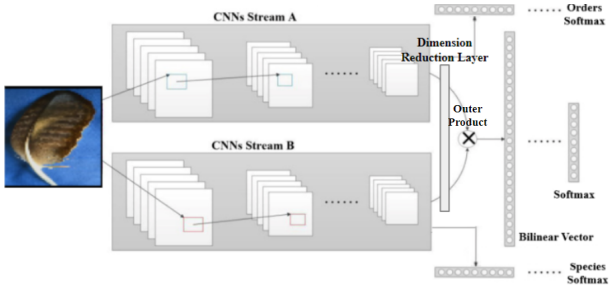


Figure 3: Custom BCNN; Fig. Adapted From [11]

Models are trained with 32 batch size and Adam optimizer. Early stopping is applied for regularization and to lessen the required compute. When the validation loss fails to decrease in 12 consecutive epochs the model yielding the lowest validation loss is saved. Kaiming initialization and $10^{-5}$ learning rate is used for all models except #4 which uses PyTorch default initialization and $2.5 \cdot 10^{-5}$ learning rate.

# 3 Results

| Model | Top-1 Accuracy | Top-5 Accuracy |
|---|---|---|
| #1 VGG16 | 86.08 | 97.76 |
| #2 VGG16, AS | 87.58 | 98.26 |
| #3 VGG16, AA | 87.42 | 98.05 |
| #4 BCNN | 91.27 | 98.93 |
| #5 Custom BCNN | 87.62 | 96.45 |

AS: Attention Substituted, AA: Attention Added

# 4 Discussion

VGG16 performs well, likely due to the bias of CNNs towards the local texture regularities rather than shape [14] which are more useful in FGIC.

Both the self-attention module and BCNN consider second-order pairwise interactions. However, the BCNN's interactions are between channels and thus orderless. Both the BCNN and attention modules best the VGG16, demonstrating that both forms of second-order information are useful for FGIC. However, the BCNN's superior performance suggests space-aware interactions are of lesser utility. The intuition is similar to that employed in part-based learning[6]; feature content rather than feature location is more useful when relative position is uninformative, as is often the case in FGIC. The caveat being that BCNN requires significantly more parameters.

#5's underperformance relative to #4 is potentially due to the lack of an informative 3-way relationship between order, species, and feather appearance. Moreover, the large number of parameters (over double VGG16) and 3 losses likely produce a complicated loss surface with many saddle points which requires techniques such as learning rate decay to navigate. For the same reason 12 consecutive epochs of non-decreasing val loss may be insufficient to determine convergence causing training to halt prematurely.

A more difficult dataset may have made the models' relative performance more pronounced. More compute would have allowed longer training times, additional configuration testing, and more robust results. Being careful to only evaluate on the test set once per model would have reduced the potential for test set overfitting.

# 5 References


[1] M. Bain, A. Nagrani, D. Schofield, and A. Zisserman, "Count, crop and recognise: Fine-grained recognition in the wild," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[2] W. Liu, M. Juhas, and Y. Zhang, "Fine-grained breast cancer classification with bilinear convolutional neural networks (bcnns)," *Frontiers in Genetics*, vol. 11, 2020.

[3] Y. Wei, S. Tran, S. Xu, B. Kang, and M. Springer, "Deep learning for retail product recognition: Challenges and techniques," *Computational Intelligence and Neuroscience*, vol. 2020, pp. 1–23, 11 2020.

[4] A. Belko, K. Dobratulin, and A. V. Kuznetsov, "Feathers dataset for fine-grained visual categorization," *CoRR*, vol. abs/2004.08606, 2020.

[5] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4476–4484, 2017.

[6] Y. Zhang, X.-S. Wei, J. Wu, J. Cai, J. Lu, V.-A. Nguyen, and M. N. Do, "Weakly supervised fine-grained categorization with part-based image representation," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1713–1725, 2016.

[7] A. Dubey, O. Gupta, R. Raskar, and N. Naik, "Maximum-entropy fine grained classification," in *Advances in Neural Information Processing Systems* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31, Curran Associates, Inc., 2018.

[8] A. Belko, K. Dobratulin, and A. V. Kuznetsov, "Feathers dataset for fine-grained visual categorization [dataset]," 2020. Available at https://github.com/feathers-dataset/feathersv1-dataset.

[9] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2018.

[10] Ramin, "Self attention in convolutional neural networks," 2021. Available at https://medium.com/mlearning-ai/self-attention-in-convolutional-neural-networks-172d947afc00.

[11] T. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," *CoRR*, vol. abs/1504.07889, 2015.

[12] K. Kutzkov, "Bilinear pooling for fine-grained visual recognition and multi-modal deep learning," 2021. Available at https://towardsdatascience.com/bilinear-pooling-for-fine-grained-visual-recognition-and-multi-modal-deep-learning-20051c1f0e7e.

[13] H. Mood, "Bilinear-cnn pytorch repository," 2019. Used suggestion of dividing after outer-product to scale data. Available at https://github.com/HaoMood/bilinear-cnn.

[14] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," *CoRR*, vol. abs/1811.12231, 2018.

[15] X.-S. Wei, Y.-Z. Song, O. Mac Aodha, J. Wu, Y. Peng, J. Tang, J. Yang, and S. Belongie, "Fine-grained image analysis with deep learning: A survey," 2021. Used as a general summary of FGIC methodologies.