Coarse to Fine Bilinear Convolutional Neural Networks for Fine-grained Image Recognition

Gundavarapu Saaketh Koundinya (saaketh.koundinya.gundavarapu@sap.com)

***Abstract*—Fine-grained recognition problems have always been a pain point for traditional Convolutional Neural Networks (CNN's). It is because of the high degree of similarity among classes. Linear feature extractors like VGG, Resnet failed to identify the very fine differences among similar images. This issue was addressed by Bilinear CNN's. Bilinear networks showed significant improvements in the area of Fine-grained recognition. Also, using additional information such as bounding boxes for localizing features gave significant improvements but these approaches needed significant effort and domain expertise. Most importantly, all these networks ignore the very natural and significant property of fine-grained classification datasets i.e., hierarchy. This paper proposes a simple yet novel way, "Coarse to Fine Bilinear Convolutional Neural Networks", which preserves the coarse features while learning the finer features. Experiments have been performed on Hierarchical Grocery Store Image Dataset and Coarse to Fine Bilinear Convolutional Neural Networks had an accuracy of 85.33% which was better than Bilinear networks, which had accuracy of 77.82% and Linear networks, which had accuracy of 72.11%.**

***Index Terms*— Fine-grained Recognition, Bilinear Pooling, Bilinear CNN's.**

I. Introduction

IMAGE RECOGNITION is a sub-domain of computer vision, which involves tasks such as classification, object detection, action recognition etc. Image classification deals with categorizing images by assigning labels. Classification problems were initially tackled using Histogram of Oriented Gradients [1] and Support Vector Machines [2] which were later superseded by deep learning methods like Convolutional Neural Networks (CNNs) which were more accurate than traditional methods.

Fine-grained image recognition falls into the classification division of image recognition. It deals with identification and classification of images with high degree of similarity into subordinate categories. Classification of different models of cars and classification of different breeds of dogs are some of the examples of Fine-grained Image Classification. Fine-grained image classification problems are dealt with using linear CNNs with pre-trained weights on ImageNet [3] dataset and transfer learning techniques. Part-based models are also vastly used in fine-grained recognition problems. While the part-based models [4] provided better results than the linear CNNs, in such tasks, they required manual annotations during training, which required domain expertise. Bilinear Convolutional Networks (B-CNNs) [5] addressed several drawbacks of CNNs and part-based models and have proved successful in the area of fine-grained image classification, however came with very high memory overload. Most recently Weakly Supervised Bilinear Attention Network (WS-BAN) [6], have been proven to be more accurate and efficient than B-CNNs.

Hierarchical datasets are such datasets which have inherent hierarchical quality, for example Hierarchical Grocery Store Image Dataset [7], which classifies groceries into coarse categories and then into finer categories. B-CNNs, part-based models, WS-BAN despite their individual drawbacks, are being used extensively in the area of fine-grained classification. However, all the methods ignore the hierarchy of the datasets.

This paper introduces a new method for fine-grained classification of hierarchical datasets called Coarse to Fine Bilinear Convolutional Neural Networks (CF-BCNNs). This method utilizes both the inherit hierarchy of the dataset and the advantages of bilinear convolutional neural networks. CF-BCNNs use bilinear pooling layer between the coarse and fine-grained linear networks, thereby improving the accuracy over B-CNNs. Experiments have been performed on the Hierarchical Grocery Store Image Dataset using CNNs, B-CNNs and CF-BNNs. CF-BCNNs have provided better results consistently.

II. Related Work

Deep learning has revolutionized the area of computer vision. CNNs have found its way into computer vision and since, have been a very dominant part of it. AlexNet (Krizhevsky et al., 2012) [8] which was trained on ImageNet [3] has produced exceptional results which marked a shift in paradigm in image classification. Many of the techniques used in AlexNet such as usage of ReLU, data augmentation and dropout layers have become standard techniques in modern CNN's. Later on, with the advancement of computing power, several new deep learning architectures have been introduced, which exceeded AlexNet's performance on ImageNet dataset. Among them Very Deep Convolutional Networks for Large-Scale Image Recognition (VGG networks) [9], from the Oxford's visual geometry group, Going Deeper with Convolutions called Inception networks [10] from Google Inc, and Deep Residual Learning for Image Recognition (Resnet) [11] by Microsoft research have been used extensively for image classification problems. For fine-grained recognition B-CNNs, part based CNNs [4] are being used extensively.

A. VGG

Very deep convolutional Networks for large scale imaging (VGG) by Simonyan et al., 2015 have been extensively used as feature extractors and have been proven effective in image classification. VGG16 is a 16-layer feature extractor, having approximately 138 million parameters. Klasson et al., 2019 [7] showed that VGG16 with pre-trained weights on ImageNet in combination with fine-tuning layers, when trained on grocery Dataset, produced an accuracy of 73.4%, which is consistent with our experiments, which 72.11% after training for 40 epochs.

B. Resnet

Deep residual learning for image recognition (Resnet) (He et al., 2016) [11] introduced ResNet which solved the vanishing gradient problem of the VGG network. They occupied less disk-space and were quicker to train. Using 18-layer architecture of Resnet (Resnet18), with approximately 11 million parameters and pre-trained weights from ImageNet for feature extraction, along with fine tuning layers we were able to achieve an accuracy of 75.63% after training for 40 epochs on grocery dataset.

C. InceptionV3

Going deeper with convolutions (Inception) (Szegedy et al., 2015) [10] introduced inception networks with inception blocks. Inception blocks compute multiple different transformations in parallel, connecting results into single output. Rethinking inception architecture for computer vision (Szegedy et al., 2015) [12] introduced Inception-v2 and Inception-v3 which used batch normalization and factorization. We used InceptionV3, with pre-trained weights on ImageNet, having approximately 24 million parameters, along with fine-tuning layers for training on grocery dataset and achieved an accuracy of 78.26%.

D. Bilinear CNN

Bilinear CNNs for Fine-grained visual recognition (B-CNN) (Lin et al., 2015) [5] had introduced bilinear convolutional networks which are specially designed for fine-grained image classification. Bilinear CNNs have two convolutional streams which are combined using outer product and then normalized using signed square root method. Bilinear networks can be implemented using two fully shared linear networks, two partially shared networks or two non-sharing networks. We have implemented non-sharing B-CNN with two VGG16 linear CNNs and a non-sharing B-CNN with two Resnet linear CNNs for feature extraction. Bilinear networks have been used as a foundation to our new approach CF-BCNN.

E. Dataset

The dataset which we selected for performing our experiments is Hierarchical Grocery store dataset (Klasson et al., 2019) [7]. Dataset consists of 5125 images of fruits, vegetables, dairy, and juice products divided into 46 course categories and 81 fine-grained categories. Each fine-grained category consists anywhere between 30 to 138 images. We have divided 70% of images for training and validation and 30% for testing. Dataset was augmented by different image transformations to overcome class imbalance.

III. CF-BCNN FOR FINE-GRAINED CLASSIFICATION

Coarse to Fine Bilinear Convolutional Neural Networks introduced in this paper are bilinear convolutions networks with a coarse-stream and a fine-grained stream which are combined using matrix outer product which is followed by normalization and regularization.

A. B-CNN architecture

The architecture proposed by Lin et al., 2015 [5] has been implemented in our CF-BNN.

Reshaping. For a given input image, using linear convolutional networks $\mathbb{A}$ and $\mathbb{B}$ features are extracted. The outputs are feature descriptors $\mathbb{p}$ of shape $c1 \times h1 \times w1$ and $\mathbb{q}$ of shape $c2 \times h2 \times w2$ denoting the number of channels, the height, and the width of the output respectively.

$$\mathbb{A}(\mathbb{I}) = \mathbb{p}(c1 \times h1 \times w1)\,. \tag{1}$$

$$\mathbb{B}(\mathbb{I}) = \mathbb{q}(c2 \times h2 \times w2)\,. \tag{2}$$

The number channels in $\mathbb{p}$ should be equal to the number of channels in $\mathbb{q}$ $(c1 = c2)$. The dimensions $c1\ and\ c2$ depend on the type of model, c = 1 represents a bag of words model. The feature vectors are then reshaped by coalescing the height and width. Reshaping will result in feature vectors having shape of $c \times m$ for $\mathbb{p}$ and $c \times n$ for $\mathbb{q}$.

$$\mathbb{p}(c \times m) \leftarrow \mathbb{p}(c1 \times h1 \times w1). \tag{3}$$

$$\mathbb{q}(c \times n) \leftarrow \mathbb{q}(c2 \times h2 \times w2). \tag{4}$$

Outer product of matrix $\mathbb{u}$ and matrix $\mathbb{v}$ is defined as $\mathbb{u} \times \mathbb{v}$, which is equal to $\mathbb{u}^{\mathbb{T}}(\mathbb{v})$. Outer product $(\mathbb{B})$ is defined as outer product of feature vector $\mathbb{p}$ and feature vector $\mathbb{q}$. The outer product is the pairwise product of each feature and the output feature vector $(\mathbb{x})$ will be C vectors of dimension $m \times n$.

$$\mathbb{B}(\mathbb{p}, \mathbb{q}) = \mathbb{p}^{\mathbb{T}}\mathbb{q}. \tag{5}$$

$$\mathbb{x}(m \times n) = \mathbb{B}(\mathbb{p}, \mathbb{q}). \tag{6}$$

Bilinear Pooling. The c vectors from outer product are pooled using bilinear pooling (Ψ), which denotes the sum pooling of c vectors obtained from outer product. During the process of pooling the location of features is ignored. The bilinear pooling will reduce all the c vectors to a single feature vector ($\mathbb{y}$) of size $m \times n$ by adding corresponding elements.

$$\Psi(\mathbb{x}) = \sum_{c=1}^{c=C} \mathbb{x}. \tag{7}$$

$$\mathbb{y}(m \times n) = \Psi(\mathbb{x}). \tag{8}$$

Signed square-root normalization. The output from the bilinear pooling is normalized using the signed square root method. Signed square root is defined as

$$\mathbb{z} \leftarrow sign(\mathbb{y})\sqrt{\mathbb{y}}. \tag{9}$$

The normalization will not change the dimensionality of the input.

L2 normalization. The output from the signed square root normalization is further normalized using L2 normalization. L2 normalization is defined as,

$$\mathbb{l} \leftarrow \frac{z}{||z||}. \tag{10}$$

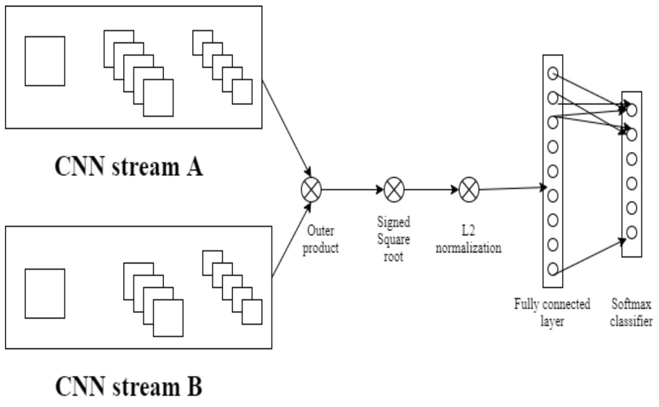

Fig. 1. BCNN architecture.

B. CF-BCNN architecture

CF-BCNN consists of

1. Coarse-grained linear CNN based classifier.
2. Fine-grained linear CNN based classifier.
3. Bilinear coarse and fine-grained classifier

Coarse-grained linear CNN based classifier. Coarse grained classifiers are trained using linear CNNs such as VGG and Resnet for feature extraction. The output of this feature extraction (a) is passed as input to a fully connected fine-tuning layer which will classify output into coarse categories.

Fine-grained linear CNN based classifier. Fine-grained classifiers are trained using linear CNNs such as VGG and Resnet for feature extraction. Like coarse-grained classifier, the output of the fine-grained feature extractor (b) is passed as input to fully connected fine-tuning layers for classification into fine categories,

Bilinear coarse and fine-grained classifier. CF-BCNN is trained using the bilinear combination of coarse-grained classifier and fine-grained classifier. The fine-tuning layers are stripped from both the previously trained coarse-grained and fine-grained classifier then they are combined using bilinear pooling followed by normalization, then the fine-tuning layers are added at the end for fine-grained classification. The entire process is described in Algorithm 1.

```
Algorithm 1: BF-CNN Algorithm.
    Input: Hierarchical data.
    Output: BF-CNN feature extractor.
    Coarse-grained_CNN ← Train dataset
    for coarse-grained classification.
    Fine-grained_CNN ← Train dataset for
    fine-grained classification.
    R
    Outer_product ← Calculate outer
    product of Coarse-grained_CNN and Fine-
    grained_CNN.
    Signed_Square_Root ← Calculate
    signed square root of the outer product.
    L2_Normalization ← Calculate L2
    normalization for the
    Signed_Square_Root
    BF-CNN_feature_extractor ← Retrain
    the entire model for fine-grained
    classification.
```

Training the CF-BCNN. The architecture of CF-BNN is a directed acyclic graph (DAG), which implies no presence of loops and back-propagation can be used effectively. The output of the coarse-grained feature extractor (a) with shape $k \times m$ and the output of the fine-grained feature extractor (b) with shape $k \times n$ when combined using outer product will result in bilinear feature vector $a^{\mathbb{T}}b$ with shape $m \times n$.

$$c(m \times n) = \Psi\big(\mathbb{B}(a, b)\big). \tag{11}$$

If l is the loss function the gradient is given by $\frac{dl}{dx}$.The gradient back propagated to C is given by chain rule. The gradient back-propagated to F is given by chain rule.

$$\frac{dl}{dA} = b\left(\frac{dl}{dx}\right)^{\mathbb{T}}, \qquad \frac{dl}{dB} = a\left(\frac{dl}{dx}\right) \tag{12}$$

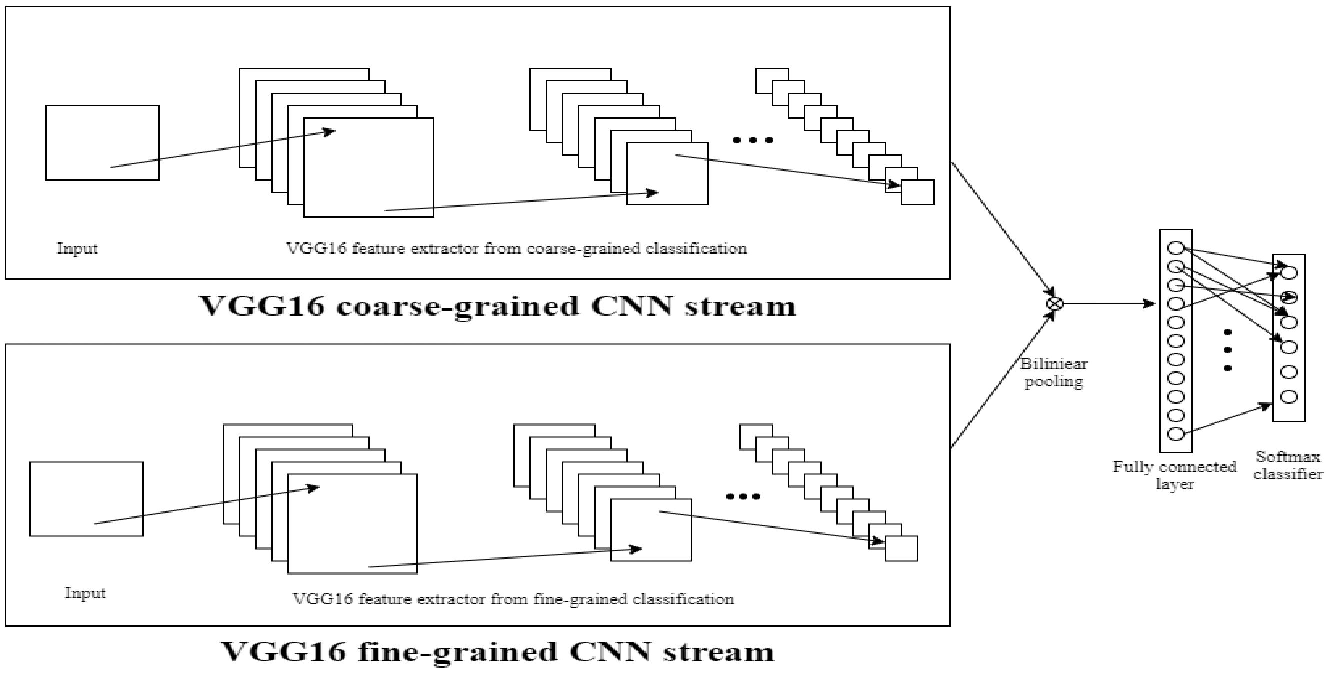
Input
VGG16 feature extractor from coarse-grained classification
VGG16 coarse-grained CNN stream
Input
VGG16 feature extractor from fine-grained classification
VGG16 fine-grained CNN stream
Biliniear pooling
Fully connected layer
Softmax classifier

Fig. 2. Architecture of CF-BCNN

IV. EXPERIMETNS

We have chosen the grocery dataset for its hierarchical quality and have trained our CF-BCNN models and other standard linear models as well as bilinear models on it.

The following criterion was maintained for training

1. The input layer for all the models is same and is having dimension of 150*150*3
2. All our models were trained for the same number of epochs for both coarse and fine-grained classification.
3. All our models were initialized with pre-trained ImageNet weights.
4. First 40 epochs for all the models, the feature extractors weights were frozen and only the fully-connected layers were trained.
5. Last 20 epochs, all the weights were un-frozen, and the complete model was trained.
6. All the fully connected layers have been made similar. Two dropout layers one after feature extraction with 50% dropout and one after first fully connected layer with 20% dropout were used.
7. Output layer for all fine-grained classifiers had 81 categories and the output layer for all coarse-grained models had 46 categories.

The fully connected layer in all the models had approximately 0.5 million parameters. For bilinear networks the total number of parameters is approximately equal to the sum of coarse-grained CNN parameters and fine-grained CNN parameters, because the outer product and the normalization layers do not have any trainable parameters.

Various models used in our experiment and their configurations.

Linear VGG16 has 16 layers for feature extraction and approximately 138 million parameters.

Linear Resnet50 has 50 layers for feature extraction with approximately 26 million parameters.

Linear InceptionV3 has 48 layers for feature extraction with approximately 25 million parameters.

Resnet B-CNN is a bilinear combination of two resnet50 linear CNNs for feature extraction and has approximately 52 million parameters with two 32-layer CNNs.

VGG16 B-CNN is a bilinear combination of two VGG16 CNNs for feature extraction and has approximately 266 million parameters with two 16-layer CNNs.

VGG16 CF-BCNN is a bilinear combination of coarse-grained VGG16 and fine-grained VGG16 for feature extraction and has approximately 266 million parameters with two 16-layer CNNs.

Resnet CF-BCNN is a bilinear combination of coarse-grained Resnet50 and fine-grained Resnet50 for feature extraction and has approximately 52 million parameters with two 50-layer CNNs.

V. Analysis and Results

The fine-grained accuracy of all the models used in the experiments have been outlined in Table-1. All the bilinear CNNs have outperformed linear CNNs significantly. CF-BCNN have outperformed their counterparts' bilinear networks and linear CNNs.

TABLE I
FINE-GRAINED CLASSIFICATION ACCURACY OF DIFFERENT MODELS ON GROCERY DATASET

Model	TYPE	Accuracy
VGG16	Linear	72.11%
Resnet50	Linear	75.63%
Inception-v3	Linear	78.26%
VGG16	Bilinear	77.82%
Resnet50	Bilinear	80.15%
VGG16	Coarse to Fine bilinear	80.21%
Resnet50	Coarse to fine bilinear	85.33%

All the accuracies are for calculated after training for 60 epochs on the test dataset.

.

VI. Conclusion

In this paper we introduced a novel architecture for convolutional neural networks for hierarchical datasets, Coarse to Fine Bilinear Convolutional networks (CF-BCNN). CF-BCNNs have been found to more accurate than both linear and bilinear CNNs in the task if fine-grained classification of hierarchical datasets CF-BCNNs exploit the simplicity of bilinear convolutional networks as they are built on top of B-CNN. CF-BCNN architecture is a directed acyclic graph and hence training from scratch as well as with pre-trained weights is possible.CF-BCNN is also highly modular and can be extended to multiple parallel networks easily. CF-BCNN also utilizes the hierarchy of dataset effectively which is found inherently in most fine-grained datasets. Thus, CF-BCNN can be extended to most kinds of fine-grained classification tasks effectively as feature extractors.

VII. References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),* vol. 1, no. 10.1109/CVPR.2005.177, pp. 886-893, 2005.

[2] B. Schölkopf and A. J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, 2001.

[3] J. Deng, W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," *IEEE Conference on Computer Vision and Pattern Recognition,* no. 10.1109/CVPR.2009.5206848, pp. 248-255, 2009.

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, no. doi: 10.1109/TPAMI.2009.167, pp. 1627-1645, 2010.

[5] T. Lin, A. RoyChowdhury and S. Maji, "Bilinear CNN Models for Fine-Grained Visual Recognition," *IEEE International Conference on Computer Vision (ICCV),* no. 10.1109/ICCV.2015.170, pp. 1449-1457, 2015.

[6] Tao Hu , H. Qi, Yan Lu, Cong huang, Qingming Huang and Jizheng Xu, "Weakly Supervised Bilinear Attention Network for Fine-Grained Visual Classification," *Computer Vision and Pattern Recognition,* 2019.

[7] M. Klasson, C. Zhang and H. Kjellström, "A Hierarchical Grocery Store Image Dataset with Visual and Semantic Labels," *IEEE Winter Conference on Applications of Computer Vision (WACV),* pp. 491-500, 2019.

[8] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional," *NIPS'12: Proceedings of the 25th International Conference on Neural Information Processing Systems,* vol. 1, p. 1097–1105, 2012.

[9] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *International Conference on Learning Representations,* 2015.

[10] C. Szegedy , W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going Deeper with Convolutions," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* no. 10.1109/CVPR.2015.7298594, pp. 1-9, 2015.

[11] K. He, X. Zhang , S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* no. 10.1109/CVPR.2016.90, pp. 770-778, 2016.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* no. 10.1109/CVPR.2016.308, pp. 2818-2826, 2016.

Saaketh Koundinya. Gundavarapu was born in Hyderabad, Telangana, India, in 1998. He received bachelor's degree in civil engineering from National Institute of Technology, Warangal, India, in 2019.

He is working as a Software Research and Development Engineer with SAP Labs, Bengaluru, India since 2019. His research interests include machine learning, deep learning, computer vision, and fine-grained image classification. He was awarded the SAP Research Fellowship in 2021.