TRANSFER LEARNING BASED IMAGE VISUALIZATION USING CNN

Santosh Giri[1] and Basanta Joshi[2]

[1]Department of Computer & Electronics Engineering, Kathford Int'l College of Engineering and Management, IOE, TU, Nepal
[2]Department of Electronics & Computer Engineering, Pulchowk Campus, IOE, TU, Nepal

ABSTRACT

Image classification is a popular machine learning based applications of deep learning. Deep learning techniques are very popular because they can be effectively used in performing operations on image data in large-scale. In this paper CNN model was designed to better classify images. We make use of feature extraction part of inception v3 model for feature vector calculation and retrained the classification layer with these feature vector. By using the transfer learning mechanism the classification layer of the CNN model was trained with 20 classes of Caltech101 image dataset and 17 classes of Oxford 17 flower image dataset. After training, network was evaluated with testing dataset images from Oxford 17 flower dataset and Caltech101 image dataset. The mean testing precision of the neural network architecture with Caltech101 dataset was 98 % and with Oxford 17 Flower image dataset was 92.27 %.

KEYWORDS

Image Classification, CNN, Deep Learning, Transfer Learning.

1. INTRODUCTION

In this modern era, Computers are being powerful day by day. They have become perfect companion with high speed computing capabilities over the time. Few decades ago it was believed that machines are only for arithmetic operations but not for complex tasks like speech recognition, object detection, image classification, language modeling etc. But these days, situation is inverted. Machines are capable of doing these things more easily with very much high accuracy. Usual algorithm consisting of finite arithmetic operations cannot provide capacity to do such complex tasks for machine. For this, Artificial Intelligence provides lots of techniques. Learning Algorithms are used for such purpose. Huge dataset is required for training the model with appropriate architecture. Testing is required to evaluate whether the model is working properly or not. Neural Network is one of AI techniques emerged long ago in 1940s but technology at that time was not so advanced. It was up at time in 1980s with the development of back-propagation [1]. Later it was again discarded due to slow learning and expensive computation. Initially It was believed that only 2 to 3 hidden layers are sufficient for Neural Network to work properly but later on it is observed that even more layers can represent high dimensional features of the input signals [2]. Image classification has received extensive attention since the early years of computer vision research. Classification remains main problems in image analysis. CNNs are applicable to fields like Speech Recognition[3], text prediction[4], handwriting generation[5]and so on.

DOI: 10.5121/ijaia.2019.10404

In These days, RAM on a machine is cheap and is available in plenty. we need lots of labeled data, time and GPU speed to train a CNN model from scratch[6]. With this limitation, it is not feasible for training the CNN network from scratch, because it is a computationally intensive task and it takes several days or even weeks with high speed GPU computer[7], which is not possible with limited resource we have. Defining appropriate model for image classifications which will produce good result in small training time and minimum CPU speed is the main task that this paper is intended to do.

2. RELATED WORKS

In [8], Noval general K nearest neighbor classifier GKMNC[9] was used for visual classification. Sparse representation based method[10] for learning and deriving the weights coefficients and FISTA[11] was used for optimization. CNN-M[12], a pretrained CNN was used for image features extractions then marginal PCA[13] is applied to reduce the dimension of the extracted features. In[14], Alex net[15] model, a deep neural network is used to learn scene image features. During the training phase, series of transformation such as convolution, max pooling, etc are performed to obtained image features. Then two classifier SVM[16] classifier and Softmax[17] classifier are trained using extracted features from the AlexNet model.

In[18], Spatial pyramid pooling was used in CNN to eliminate the fixed size input requirements. for this new network structure SPP-net was used, which can generate a fixed length representation regardless of image size. Standard back propagation algorithm[1] was used for training, regardless of the input image. In[19], Kernalized version of Nave bayes Neabour[20] was used for image classification and SVM[16] classifier was trained on Bag-Of-Features[21] for visual classification. In[22], Extension of the HMAX[23], a four level NN has been used for image classifications. The local filters at first level are integrated into last level complex filters to provide a exible description of object regions. In[24], Nearest neighbor classifier[20] was used for visual classifications. SIFT[25] descriptor to describe shape, HSV[26] values to describe colors and MR filters to describe texture were used.

3. METHODOLOGY

3.1. Image Preprocessing

The learning method used in this experiment is supervised learning[27].In supervised learning[27], we have to label the data for training & evaluation of the model. For training and testing the model, Caltech101[28] image dataset and Oxford 17 flower[29] image dataset were used. In preprocessing, all images from Caltech101[28] and Oxford 17 flower[29] are resized to 299x299X3 because to train a CNN using transfer learning[30], image input size to CNN must be same as the input size given to original model. Then images from two standard image dataset[28][29] were divided into training set, validation set and testing set. From caltech101[28] dataset, amongst 70 images per class, 60 images were used for training & validation and remaining 10 images were used for testing. For Oxford 17 flower[29] image dataset, amongt 80 images per class, 64 images were used for training & validation and remaining 16 images were used for testing.

3.2. Cnn Model Design

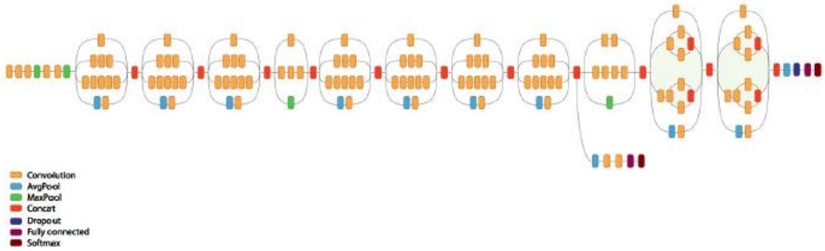

Fig. 1. Architecture Of Inception V3 Model [6].

The CNN model designed here is based on inception v3[6]. Fig. 1 represent the detail architecture of Inception model. It is a 42 layer Deep, pretrained CNN model trained on ImageNet[31], which was 1st runner up in ImageNet Large Scale Visual Recognition Competition, with lower error rate i.e. top-1 error: 17.2% & top-5 error: 3.58 %. The inception model has two parts; feature extraction and classification. we make use of the feature extraction part of inception model and retrained the classification layer with Oxford 17 flower image dataset[29] and Caltech101 image dataset[28]. To retrain the classification layer we implement the transfer learning[30]

3.3. Transfer Learning

Inception v3[6] is a convolutional neural network model and by using GPU configured computer it takes weeks to train from scratch[6], tensorflow[32] a machine learning framework which provides plateform to train the classification layer with images from Caltech101[28] and Oxford 17 flower[29] using transfer learning mechanism[30]. transfer learning mechanism[30], which keeps the weights and bias values of the feature extraction layer and removes parameters on classification layer of inception v3[6]. First the input image of size of 299x299x3 are feded to feature extraction layer of CNN. After that feature extraction layer calculate the feature values for each images, feature vector are 2048 float values for each images. then classification layer of the CNN is trained with these feature vector. The output labels in the classification layer is equal to the number of image classes on the dataset.

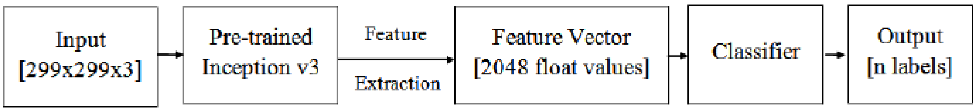

Fig. 2. Transfer Learning Implementation Pipelines.

4. Evaluation

The pretrained inception model[6] is used here for experimental purpose and the platform used here is tensorflow[32] and the hardware plateform used here is dell latitude e6410: 2.4 GHZ processor intel i5, 4 GB RAM. For experiment Oxford 17 flower dataset[29] and Caltech101 dataset[28] were used.

4.1. Dataset

Oxford 17 flower dataset[29]: This dataset consist of 17 category flowers images with 80 images for each class. The flowers chosen are some common flowers in Britain. The images has been collected by Maria-Elena Nilsback and Andrew Zisserman of University of Oxford, UK. Caltech101 image dataset[28]: This dataset has 101 classes of images. It has 40 to 800 images per class and Collected by Fei-Fei Li, Marco Andreetto, and Marc Aurelio Ranzato in September 2003.

4.2. Evaluation Procedure

To test the model on Caltech dataset[28], 20 classes of images form Caltech101 image dataset[28] are used. Each category consist total of 70 images. The training and validation set consist 60 images and testing set consists of 10 images per class. And to test the model on Oxford 17 flower dataset[29] 17 classes of flower images were used, each class consist of 80 images: 64 images per class for training & validation and 16 images per class for evaluation were used.

To train the model transfer learning mechanism[30] is implemented on pretrained inception model[6]. In transfer learning, weights and bias values of the feature extraction layer are kept same as original CNN model and removes parameters on classification layer of a CNN. To train the classification layer, training set images from two dataset[29][28] were used. The Back propagation algorithm[1] was used to train the classification layer of the CNN model and weight parameters of classification layer is adjusted by using cross entropy cost function by calculating the cross entropy error between classification layer output and input feature value from feature extraction layer.

5. RESULTS

5.1. Training

The training accuracy, validation accuracy and cross entropy graph on flower dataset[29] and Clatech101 dataset[28] are given in Fig. 3, Fig. 4, Fig. 5 and Fig. 6 respectively. The parameters used for retraining the model are, training steps: 4500, training interval: 1 and learning rate: 0.045. The training and validation accuracy graph in Fig. 3 and Fig. 5 represents accuracy the CNN model get from training and validation images. And the cross entropy graph in Fig. 4 and Fig. 6 represents the difference between the actual output and input feature value while training the Classification layer of CNN model. In Fig. 3, 4, 5, 6 the Blue line represents accuracy & cross entropy error variation curve on training images and Orange line represents accuracy and cross entropy error variation curve on validation images from two dataset[29][28]. Fig. 3 represents the training & validation accuracy variation on flower image dataset[29]. Training accuracy was 91.2% at the beginning of the training process and starts to increase, after completion of (3/4) training steps it reached and remains to 100%. Validation accuracy was 72.13% during initiation of training and validation process and final validation accuracy was 92.7%. Fig. 4 represents the cross entropy error for training and validation respectively on flower image[29]. The cross entropy error were 0.58 and 0.69, during initiation of training and validation process. As the training steps increases the cross entropy error starts to decrease. At final step of training and validation the training and validation cross entropy error were 0.02 and 0.09.

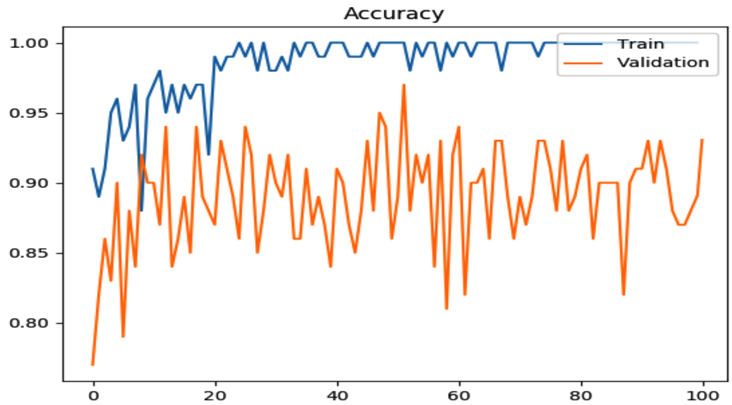

Fig. 3. Accuracy graph on Oxford 17 flower dataset[29].

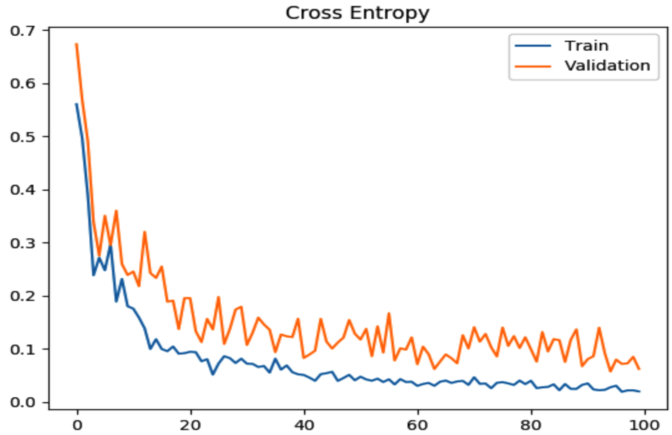

Fig. 4. Cross entropy on Oxford 17 flower dataset[29].

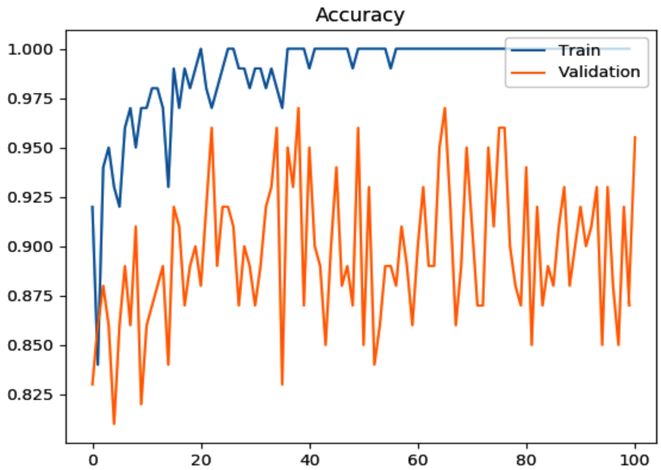

Fig. 5. Accuracy graph on Caltech101 dataset[28].

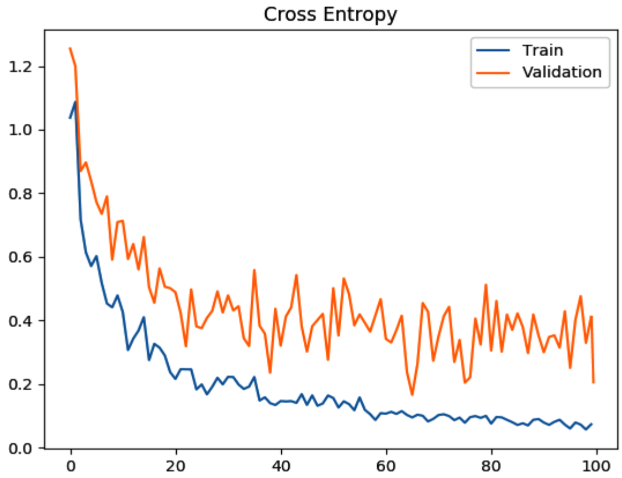

Fig. 6. Cross entropy graph of training on Caltech101 dataset[28].

Fig. 5 represents the training & validation accuracy variation respectively on Caltech101 image dataset[28]. Training accuracy was 83.6% at the beginning of the training process and starts to increase, after completion of (3/5) training steps it reached and remains to 100%. Validation accuracy was 82.7% during initiation of training and validation process and final validation accuracy was 94.9%. Fig. 6 represents the cross entropy error for training and validation respectively on Caltech images[28].

The cross entropy were 1.0 and 1.3, during initiation of training and validation process. As the training steps increases the cross entropy error starts to decrease. At final step of training and validation process the training and validation cross entropy error were 0.28 and 0.1.

5.2 Testing and Evaluation

The classification result of the system on testing images from flower dataset[29] and Caltech dataset[28] are represented in evaluation chart given in Fig. 7 and Fig. 8 and the results comparisons of our system with other experiments are represented in Table 1 and Table 2. respectively. The 'TP' in the chart is 'True Positive', which represents correctly classified images by the system.

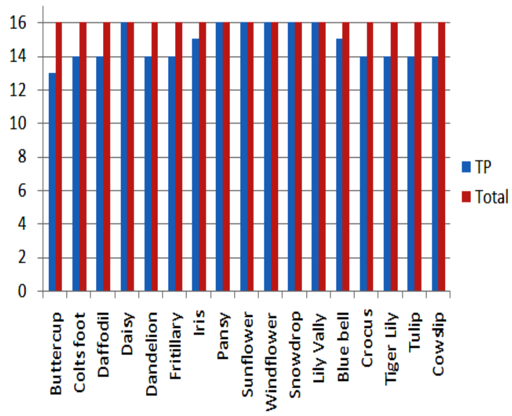

Fig. 7. Performance evaluation parameters on flower dataset[29].

For testing the model on flower image[29], 16 images were used from each class, i.e. 272 images from 17 classes. Amongst 272 testing images, correctly classified images by the system were 251 as shown in Fig. 7. For testing the model on caltech image dataset[28], 10 images were used from

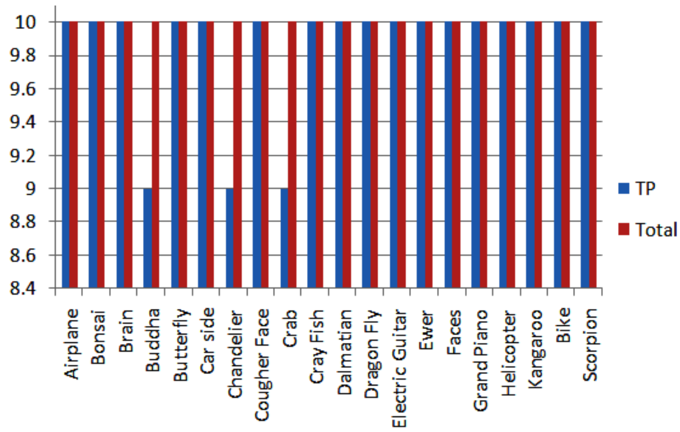

Fig. 8. Performance evaluation metrics on Caltech image dataset [28].

each class, i.e. 200 images from 20 classes. Amongst 200 testing images, correctly classified images by the system were 196 as shown in Fig. 8.

Table 1. Performance evaluation of our system on flower dataset [29] with other method

S.N.	Method	Mean Precision
1	[24]	81.3 %
2	[33]	90.4 %
3	**Our system**	**92.27 %**

Table 2. Performance evaluation of our system on Caltech101 dataset [28] with other method

S.N.	Method	Mean Precision
1	[8]	59.12 %
2	[19]	75.2 %
3	[22]	76.32 %
4	[18]	80.4 %
5	**Our system**	**98.0 %**

6. CONCLUSIONS

In this paper, the classification layer of pre-trained inception v3 model was re trained successfully by implementing transfer learning mechanism. The model yields precision of 92.27 % on 17 classes of Oxford 17 flower image dataset and 98.0 % on 20 classes of Caltech101 image dataset.

REFERENCES

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, Backpropagation applied to handwritten zip code recognition," Neural computation, vol. 1, no. 4, pp. 541-551, 1989.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097-1105.

[3] D. R. Reddy, Speech recognition by machine: A review," Proceedings of the IEEE, vol. 64, no. 4, pp. 501-531, 1976.

[4 A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng, Text detection and character recognition in scene images with unsupervised feature learning,"in Document Analysis and Recognition (ICDAR), 2011 International Conference on. IEEE, 2011, pp. 440-445.

[5] T. Varga, D. Kilchhofer, and H. Bunke, Template-based synthetic handwriting generation for the training of recognition systems," in Proceedings of the 12th Conference of the International Graphonomics Society, 2005, pp. 206-211.

[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, Rethinking the inception architecture for computer vision," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818-2826.

[7] 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, 2015, pp. 1-9.

[8] Q. Liu, A. Puthenputhussery, and C. Liu, Novel general knn classifier and general nearest mean classifier for visual classification," in Image Processing (ICIP), 2015 IEEE International Conference on. IEEE, 2015, pp. 1810-1814.

[9] J. M. Keller, M. R. Gray, and J. A. Givens, A fuzzy k-nearest neighbor algorithm," IEEE transactions on systems, man, and cybernetics, no. 4, pp. 580-585, 1985.

[10] J. A. Tropp, Greed is good: Algorithmic results for sparse approximation," IEEE Transactions on Information theory, vol. 50, no. 10, pp. 2231-2242, 2004.

[11] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems," SIAM journal on imaging sciences, vol. 2, no. 1, pp. 183-202, 2009.

[12] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, Return of the devil in the details: Delving deep into convolutional nets," arXiv preprint arXiv:1405.3531, 2014.

[13] I. Jolliffe, Principal component analysis," in International encyclopedia of statistical science. Springer, 2011, pp. 1094-1096.

[14] J. Sun, X. Cai, F. Sun, and J. Zhang, Scene image classification method based on alex-net model," in Informative and Cybernetics for Computational Social Systems (ICCSS), 2016 3rd International Conference on. IEEE, 2016, pp. 363-367.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097-1105.

[16] C. Cortes and V. Vapnik, Support-vector networks," Machine learning, vol. 20, no. 3, pp. 273-297, 1995.

[17] N. M. Nasrabadi, Pattern recognition and machine learning," Journal of electronic imaging, vol. 16, no. 4, p. 049901, 2007.

[18] K. He, X. Zhang, S. Ren, and J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition," in European conference on computer vision.Springer, 2014, pp. 346-361.

[19] T. Tuytelaars, M. Fritz, K. Saenko, and T. Darrell, The nbnn kernel," in Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011, pp. 1824-1831.

[20] K. P. Murphy et al., Naive bayes classifiers," University of British Columbia, vol. 18, 2006.

[21] Z. S. Harris, Distributional structure," Word, vol. 10, no. 2-3, pp. 146-162, 1954.

[22] C. Theriault, N. Thome, and M. Cord, Extended coding and pooling in the hmax model," IEEE Transactions on Image Processing, vol. 22, no. 2, pp. 764-777, 2013.

[23] M. Riesenhuber and T. Poggio, Hierarchical models of object recognition in cortex," Nature neuroscience, vol. 2, no. 11, p. 1019, 1999.

[24] M.-E. Nilsback and A. Zisserman, A visual vocabulary for flower classification," in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 2. IEEE, 2006, pp. 1447-1454.

[25] D. G. Lowe, Object recognition from local scale-invariant features," in Computer vision, 1999. The proceedings of the seventh IEEE international conference on, vol. 2. Ieee, 1999, pp. 1150-1157.

[26] A. R. Smith, Color gamut transform pairs," ACM Siggraph Computer Graphics, vol. 12, no. 3, pp. 12-19, 1978.

[27] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, Supervised machine learning: A review of classification techniques," Emerging artificial intelligence applications in computer engineering, vol. 160, pp. 3-24, 2007.

[28] L. Fei-Fei, R. Fergus, and P. Perona, Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," Computer vision and Image understanding, vol. 106, no. 1, pp. 59-70, 2007.

[29] M.-E. Nilsback and A. Zisserman, A visual vocabulary for flower classification," in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 2. IEEE, 2006, pp. 1447-1454.

[30] S. J. Pan, Q. Yang et al., A survey on transfer learning," IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1345-1359, 2010.

[31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database," in CVPR09, 2009.

[32] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard et al., Tensor flow: a system for large-scale machine learning." in OSDI, vol. 16, 2016, pp. 265-283.

[33] Y. Chai, V. Lempitsky, and A. Zisserman, Bicos: A bi-level co-segmentation method for image classification," 2011.