Image Classification- Caltech 101

Vamsi Krishna Kovuru, Sai Kalyan Yeturu, Gowtham Kommineni, Sai Amrit Bulusu

Abstract:

Image classification is one of the core problems of Computer Vision, which is the task of assigning an input image one label from a fixed set of categories. It is particularly complex to perform image classification due to multiple differences between the images of the same class label such as viewpoint variation, scale variation, deformation, occlusion, background clutter and many more factors.

The goal of the project is to first understand what kind of features can be extracted from the features, perform preprocessing of the images, perform feature selection to reduce the dimensionality of the space - which is one of the most important factors while performing image classification - and study different classification methods of supervised and unsupervised learning approaches to figure out which performs a better role of classification. Introduction section deals with what we propose to do so in a step by step procedure, Image Preprocessing section deals with all the techniques involved to preprocess the data and their observations, Feature Engineering deals with multiple efforts that have been made to extract and select key underlying features of each image, Classification section contains information about algorithms used and their observations, Experimental setup section provides information on the experiments that were done and finally Results and Analysis section deals with the findings and analysis on the experiments.

1. Introduction:

Image classification is the process of labelling the images into one of several predefined categories. The steps of classification include Image preprocessing, feature extraction, feature selection and object classification. Classification process consists of the following steps:

A. **Pre-processing:** Enhances the quality of the input image such as noise removal, image Equalization, find the key point descriptors, cropping and locate the edges and image of interest.

B. **Feature Extraction:** It contains the extraction of features for the images to train and classify the data.

C. **Feature Selection:** It contains the selection of important features for any given image which best performs for any training and test data.

D. **Classification:** Object classification step classifies detected objects into predefined classes by using proper method that matches the image patterns with the target patterns.

The Caltech101 dataset consists of 101 class of images. Dataset consists of 9,146 images, with 101 categories with more than 40 images per category. The size of each image is roughly 300 x 200 pixels. It can be downloaded from [1]. This paper presents the classification of image using different classifiers and compare which model performs better. The results of experiments are presented in the paper, and conclusions are drawn.

Fig-1: Sample Caltech101 images in grayscale.

2. Image Preprocessing:

For Caltech101 image classification we have performed image preprocessing techniques and extract features. In this paper, we present the preprocessing techniques we have applied. Initially the images present in the faces category of the caltech101 dataset are grayscale images, we have opted to apply classification techniques for the grayscale images. We have used OpenCV library(python) for image preprocessing. To balance the contrast of the grayscale images, histogram equalization technique is used. the noise in the images is reduced by applying Gaussian filter. As we know the best way for images is to obtain the edges which are scale invariant. For this we have use canny edge detectors. From the fig-1 we can notice the object in the image is small compared to the

background. For reducing these inconsistencies, we cropped the image to obtain only the object present in the images by using the contours and key point descriptors. Fig-2 provides all the image preprocessing operations performed on images. We have also detected Keypoint descriptors for images which are scale invariant. These scale invariant descriptors can be used for feature selection process.

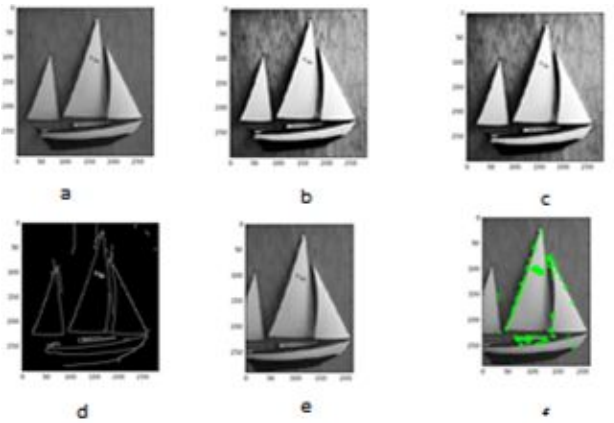


Fig-2: Image Preprocessing a: Original, b: Equalized image, c: Gaussian filter, d: Edges, e: Cropped image, and f: Key point descriptors

3. Feature Engineering:

3.1 Feature Extraction:

In this section, we discuss the feature extraction and feature selection methods we have performed on caltech101 dataset. For feature extraction, we have resized each image in the dataset from roughly 300X200 to 50X50. Initially the images are in RGG format so the dimensionality of each image is around 3X300x200 which gives 180,000 attributes (or pixels) for better understanding of the models and time considerations grayscale images would reduce the dimensions to 1x300x200 and after resizing the images the number of attributes are reduced to 2500 by flattening the images into an numpy array. We have selected different features based on the processing tools we used and added histogram of Oriented Gradients of images features to improve the performance.

3.2 Feature Selection:

The most important step after feature extraction is feature selection. It plays an important role, especially in classification problems. A well extracted feature must have the value of robustness, discriminative, and easy to compute an efficient algorithm. After performing the classification models and observing the results for the 2500 features, the features selection methods are implemented to increase the performance of our model. In this paper, we are discussing Random Forest feature selection and PCA analysis. The irrelevant input features may lead to overfitting. Feature selection focuses on the outstanding attributes over the dataset, which offers higher accuracy. There are lots of potential benefits of feature selection such as facilitating data understanding, reducing utilization times and techniques, reducing measurement, and defying the curse of dimensionality to improve prediction performance. Reduce the number of features to be used by using techniques for Feature Selection such as PCA, Info Gain Evaluation.

Information Gain Attribute Evaluation in WEKA: Evaluates the worth of an attribute by measuring the information gain with respect to the class.

InfoGain(Class,Attribute) = H(Class) - H(Class | Attribute).

Principal Component Analysis in WEKA: Performs a principal components analysis and transformation of the data. Used in conjunction with a Ranker search. Dimensionality reduction is accomplished by choosing enough eigenvectors to account for some percentage of the variance in the original data---default 0.95 (95%). Attribute noise was filtered by transforming to the PC space, eliminating some of the worst eigenvectors, and then transforming back to the original space. We have selected a minimum threshold of 2e-03 which reduced the number of attributes from 2500 to 1500.

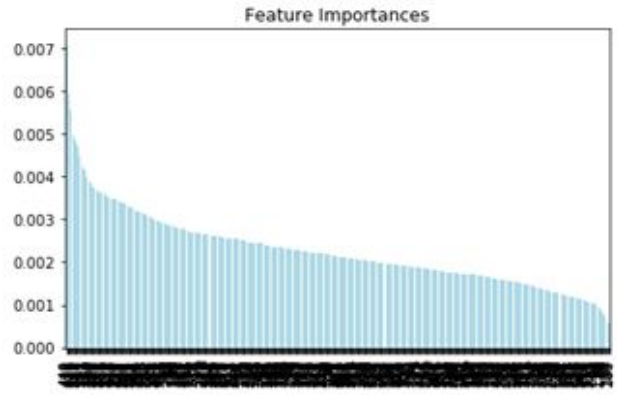


Fig-4: Feature importance vs category graph using Random Forest Feature Selection.

4. Classification:

Discretization: Discretization refers to the process of converting or partitioning continuous attributes, features or variables to discrete or nominal attributes/ features/variables/intervals. We have converted the class value into nominal values and then discretized the entire selected attributes to create important intervals for each

attribute to make classification easier. This process is also known as Data Binning.

In this paper, we are discussing multiple supervised learning models like K-NN, Artificial Neural Network, Support Vector Machine and Deep learning model Convolutional Neural Networks are implemented on Caltech101 dataset.

4.1 K- Nearest Neighbors:

In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. If k = 1, then the object is simply assigned to the class of that single nearest neighbor. For Caltech101 dataset we compared the results for different scenarios where n ranges from 1 to 30 and Classes ranging 10,50,101 with 2500 attributes and selected 1500 attributes over 5,10,20,30,40,50,60 images per class. Results for this model are represented in fig-

4.2 Artificial Neural Network:

Neural network is to simulate lots of densely interconnected brain cells inside a computer so we can get it to learn things, recognize patterns, and make decisions in a humanlike way. The amazing thing about a neural network is that we don't have to program it to learn explicitly: it learns all by itself, just like a brain.

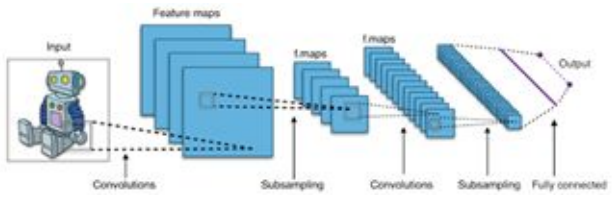
Trained Neural network model with alpha=1e-05, learning rate =0.01 with 12 neurons and 2 hidden layers with **relu activation function.**

4.3 Support Vector Machines:

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. For our project we have used radial basis 'RBF' kernel with cost values c= 10.

4.4 Convolution Neural Network:

Convolutional Neural Networks (CNN's) are multi-layer neural networks (sometimes up to 17 or more layers) that assume the input data to be images. For our project we have implemented basic CNN with 2 convolutional layers, 2 pooling layers, 2 dropout layers and 2 relu layers with softmax activation. Here is the basic working of 4 layer CNN with resized images of 50X50 dimensions.



- INPUT [50x50x1] will hold the raw pixel values of the image, in this case an image of width 32, height 32, and with single grayscale channel
- CONV layer will compute the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume. This may result in volume such as [50x50x12] if we decided to use 12 filters.
- RELU layer will apply an elementwise activation function, such as the max (0,x) max (0,x) thresholding at zero. This leaves the size of the volume unchanged ([50x50x12]).
- POOL layer will perform a down sampling operation along the spatial dimensions (width, height), resulting in volume such as [25x25x12].
- FC (i.e. fully-connected) layer will compute the class scores, resulting in volume of size [1x1x101], where each of the 101 numbers correspond to a class score, such as among the 101 categories of Caltech101. As with ordinary Neural Networks and as the name implies, each neuron in this layer will be connected to all the numbers in the previous volume.

5. Experimental Setup:

5.1. Dataset:

The experiment used Caltech101 dataset as mentioned earlier. Fig.4 shows examples of the 40 image categories used in this paper a sample of 101 categories. The categories with mostly 30-40 images per category with few exceptions of 800 per Airplanes category which leads to class unbalancing when applying classification algorithms.

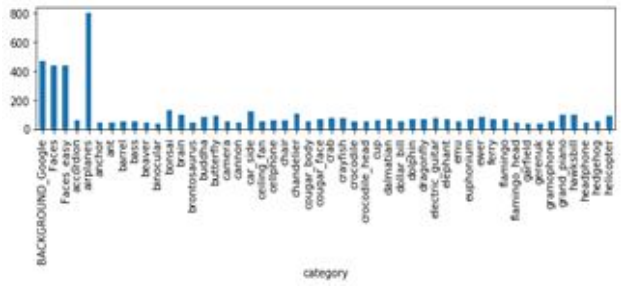


Fig-4: Sample Caltech101 dataset for 40 classes ranging from 40-800 images per class.

5.2 Experiments:

We have performed Supervised Learning Classification models mentioned earlier on the dataset with 5,10,20,30,40,60 and total images per class with features

extracted from Histogram of Oriented Gradients (HOG) of images descriptors with 490 attributes and Normalized HOG descriptors.. Features extracted from original images(Gray scale images), Equalized images, By reducing the noise using the Gaussian filter, Edge images, Cropped images and key point descriptors extracted from these images with 2500 attributes as the resized images are of dimensions 50X50. Additionally we have combined HOG features with different preprocessing techniques and applied machine learning models for classification of the images to improve the performance.

5.2.1 Image Augmentation:

Image augmentation can increase the size of your training set 10-fold or more by applying different transformation techniques to the images. For our project we have decided to take 10 fold images with transformations including flipping, rotate by 45 degrees, stretch the images and zoom. Adding this data to the previous dataset and applied for 10 classes with 40 images per class for which we obtained 4400 images to classify for 10 classes which were previously 400.

6. Experimental Results and Analysis:

The Performance measure of the KNN is better for 10 classes with number of nearest neighbours=22.Fig-6 represents the performance for HOG descriptors, Cropped images and HOG+edges Images(2940 attributes). From this Fig-6 we got the highest accuracy for HOG descriptors with 490 features only. For further implementation we have used HOG descriptors and combined with different features.

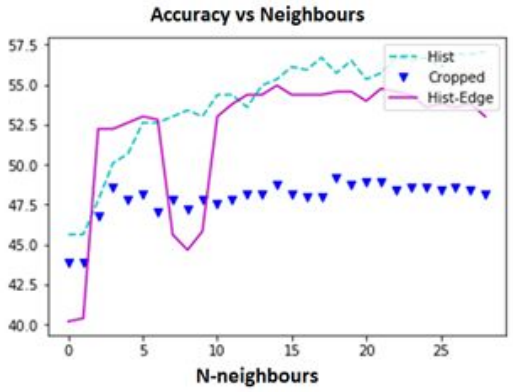


Fig-6: Accuracy vs Number of neighbors for HOG descriptors, Cropped images and HOG +Edge images.

From Fig-7 the accuracy vs number of neighbours graph we obtain the highest accuracy for Gaussian filter images with around 69% accuracy when compared to Edge images and Orignal grayscale images. For further analysis we have used Gaussian filter which reduces the noise in the images.

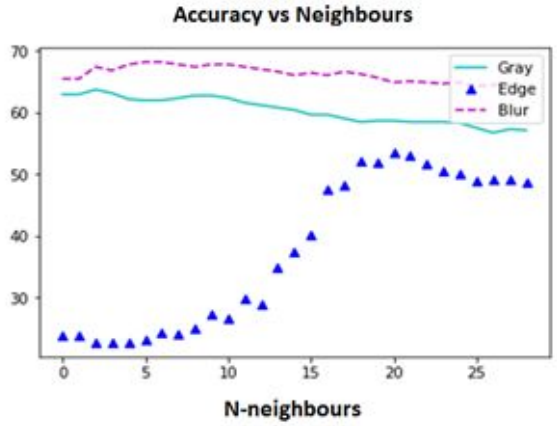


Fig-7: Accuracy vs Number of neighbors for Gaussian filtered images, Edge images and Original images.

From Fig-8 the accuracy vs number of neighbours graph we obtain the highest accuracy for HOG by normalizing the dataset with around 68% accuracy when compared to Cropped images and Normalized dataset.

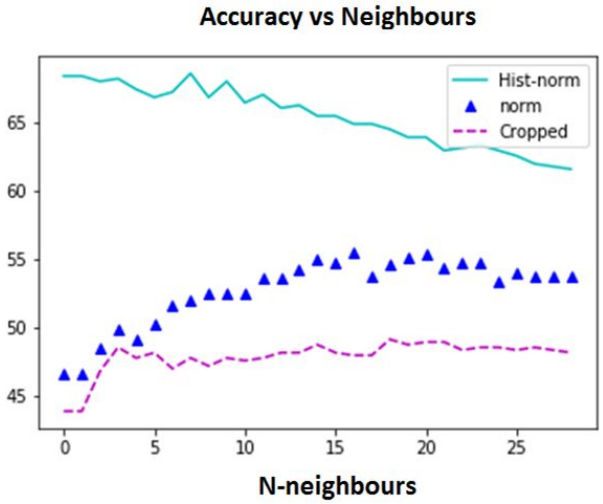


Fig-8: Accuracy vs Number of neighbors for Normalized HOG descriptors, cropped images and Normalized gray images.

For the SVM model radial basis we obtained had high accuracy all above 77 percent for classifying 10 classes. For the first two classes we obtained an accuracy of 100% accuracy and for the last three classes we got an accuracy of 50% . Our model could not classify the other classes properly. So we got an overall accuracy of around 45%. This is because of the presence of some bias classes in the dataset. From figure 4 we can see the plot describing the number of images per each class. For the first 3

classes there are 400,400,800 images respectively. Confusion matrix from figure 9 describes that some of the images in the other classes are classified among the first 3 classes. This is because of the bias the first 3 classes has.

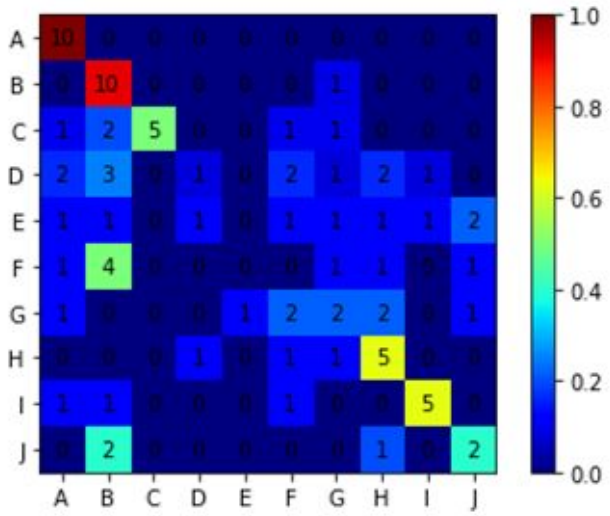


Fig-9: Confusion Matrix for SVM with 10 images per category for 10 classes.

From Fig-10 we can obtain the Cluster for 10 classes with 1701 images.

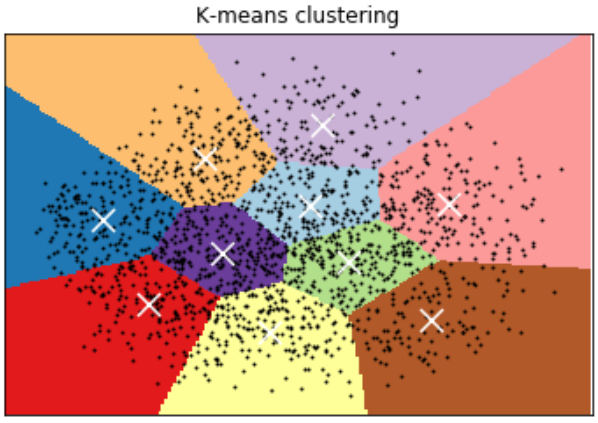


Fig-10 K-means Clustering for 10 classes with 1701 images

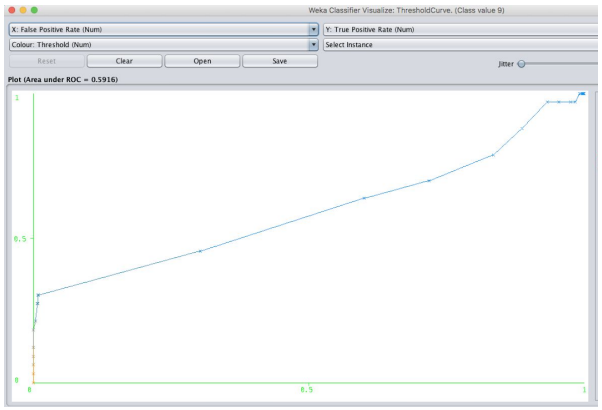


Fig-11 ROC curve average performing class using SMO on WEKA for complexity parameter c=10

Along with SVM, we have also tried to apply softmax to train the model. The prediction from softmax had 55.3% accuracy with transfer learning features. This is lower than 78.51% result we got from SVM. Thus, we decided to use SVM in the final model.

From Fig-12 we can identify the accuracy of 74.54 for the KNN for K=20 with mean error =0.1632.

```
Time taken to build model: 16.14 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 2.75 seconds

=== Summary ===

Correctly Classified Instances         383               74.5136 %
Incorrectly Classified Instances       131               25.4864 %
Kappa statistic                          0.6181
Mean absolute error                      0.1632
Root mean squared error                  0.2778
Relative absolute error                116.9307 %
Root relative squared error            105.5104 %
Total Number of Instances              514

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               1.000    0.018    0.654      1.000    0.791      0.801    0.993     0.708     1
               0.947    0.041    0.955      0.947    0.951      0.906    0.979     0.960     2
               0.000    0.002    0.000      0.000    0.000      -0.007   0.473     0.032     3
               0.077    0.006    0.250      0.077    0.118      0.127    0.687     0.126     4
               0.841    0.215    0.589      0.841    0.693      0.570    0.821     0.550     5
               0.077    0.010    0.167      0.077    0.105      0.098    0.538     0.039     6
               0.150    0.002    0.750      0.150    0.250      0.326    0.796     0.218     7
               0.071    0.012    0.143      0.071    0.095      0.083    0.724     0.072     8
               0.222    0.000    1.000      0.222    0.364      0.468    0.872     0.205     9
               0.314    0.029    0.440      0.314    0.367      0.334    0.825     0.278     10
Weighted Avg.  0.745    0.081    0.724      0.745    0.713      0.661    0.881     0.665

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
  17   0   0   0   0   0   0   0   0   0 |   a = 1
   0 231   1   0  11   0   0   1   0   0 |   b = 2
   1   0   0   0   8   0   0   0   0   2 |   c = 3
   3   1   0   1   8   0   0   0   0   0 |   d = 4
   2   5   0   2 116   5   1   0   0   7 |   e = 5
   1   1   0   0   6   1   0   1   0   3 |   f = 6
   0   1   0   1  14   0   3   1   0   0 |   g = 7
   0   1   0   0  11   0   0   1   0   1 |   h = 8
   2   0   0   0   3   0   0   1   2   1 |   i = 9
   0   2   0   0  20   0   0   2   0  11 |   j = 10
```

Fig-12 Confusion Matrix for using Discretizated attributes on KNN for 10 classes with K=20

Initially when the classifiers were directly applied to the entire dataset, the accuracy and precision scores were around 8-10%. Using multiple feature selection procedures and discretization of the data the accuracy was increased to around 35% - 40% or considering only for 10 classes, the accuracy was 78% using an SVM algorithm using WEKA for complexity parameter C=10.

The challenge for this dataset is to understand what features we can remove after the preprocessing stage. Multiple methods like PCA, information gain and gain ratio evaluation were conducted on the attributes to select

only 1500 from 2500. This increased the accuracy by at least 5%.

Applied image augmentation technique over 10 classes with 40 images per each class we got 4400 images. This increased the accuracy by 7% overall for a KNN implementation and further implementing this models for total dataset can improve the accuracy.

The accuracy for the 101 classes using Convolutional Neural Network for the basic implementation we have obtained 45.8% and more detailed feature engineering and extraction methods can increase the accuracy. But the performance of the CNN is decreased due to overfitting of the data.

One of the main reasons we observe why there was more than 50% misclassification is due to biased class balancing. If we look at the confusion matrix, we can see a couple of classes having more than 700 samples to it. One future scope might be to produce equal number of instances per class or to create more classes to diversify the data in a better manner.

7.Conclusion

Several techniques such as histogram of gradients, blurring the images and converting the images to grayscale were effective techniques to generate features among the ones that were used for image preprocessing.

The classifiers worked better when feature selection methods such as PCA and information gain ranker methods were used. Discretization also improved the classification accuracy by a relatively huge margin.

Traditional machine learning methods such as KNN, SVM and Ensemble Methods like Random forest worked fairly better when tuned the model to fit the data best.

The convolutional neural network features performed the best out of the methods that were attempted and one can conclude that tuning the model and arranging the layers in the order to fit the model best for the data improved the accuracy.

8. Future Scope:

Optimization: The challenge for this dataset is to understand what features we can remove after the preprocessing stage. We can further increase the performance of the model using the feature selection techniques and preprocessing techniques like SIFT scale invariant methods.

Class balancing: One of the main issues that was observed while performing supervised or unsupervised learning is the very bad balancing of samples for every class. As an effect, the classification was mostly biased to the classes having more samples. One solution that one can inculcate in the future is to provide more samples making the distribution fairly equal.

Image Augmentation: We have implemented image augmentation for only 10 classes with 40 images we can further implement this technique for the entire dataset with 101 features and 40 images per class.

Acknowledgment:

We are thankful to Professor Nadia Najjar for providing us with the academic support and for being reachable anytime we had a concern regarding the project. We thank the CCI Technical support for their contribution with this poster. We thank the vast repositories of information out there on the internet which provides us with everlasting knowledge and the inspiration to learn more.

References:

[1] http://www.vision.caltech.edu/Image_Datasets/Caltech_101/Caltech101/

[2] L. & Q. Weng, “A survey of image classification methods and techniques for improving classification performance,” Int. J. Remote Sens., 2007.

[3] 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,” Conf. Comput. Vis. Pattern Recognit. Work., pp. 178–178, 2004.

[4] S. Lazebnik, C. Achmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In IEEE Conf. on Computer Vision and Pattern Recognition, volume 2, pages 2169–2178, New York City, June17 - 22 2006.

[5] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, CalTech, 2007.

[6] L. Fei-Fei, R. Fergus and P. Perona, “A Bayesian approach to unsupervised learning of object categories’, Proc. ICCV, 2003.