A NOVEL IMAGE LEARNING SYSTEM TO RECOGNIZE OBJECT USING GOOGLE IMAGE SEARCH AND ITS ANALYSIS

Anubhav Singh and Sumit Pathak
Department of Information Technology, College of Technology, Govind Ballabh Pant University of Agriculture & Technology , Pantnagar City, Uttarakhand, India

ABSTRACT
One very important challenge in the field of Computer Vision and Machine Learning is the implementation of a system that can learn in the same way as a human can learn. In particular, in the current state-of-the art object recognition systems, it is highly desirable to detect as much object as possible with as much as possible good accuracy along with the ability to learn new objects and it add to their understanding. Although a significant number of Object Recognition systems have been developed and implemented, triggering very accurate results, the vast majority of them cannot add new object to their understanding; this is mainly due to the fact that more emphasis is given to the accuracy of recognition than to the fact that even humans are not hundred per cent accurate, rather the focus should be on increasing the number of objects a system can recognise with a satisfactory accuracy. In this paper, we present a novel object recognition system using SIFT descriptors, SVM for classification a Google Images Search. Our main focus was to recognize new images that the user downloads along with maintaining satisfactory accuracy and speed of. Object recognition. Our Object Recognition system download new images according to user learn them and save them for future use and recognition. Our novel system also maintains a cache of the feature tables, thus increasing the speed of recognition after first processing by manifolds. This is, to our knowledge, the first system that dynamically learns new objects using internet and thus adding it to its memory.

KEYWORDS: *Computer Vision, Machine Learning, Object Recognition, SIFT, SVM, Google Image Search, feature table, cache.*

I. INTRODUCTION

We want to build a vision system that can download new images and learn those images, thus recognizing that same object from a dataset. This problem is very difficult and largely unsolved. The current paradigm is to collect a large training set of images of a desirable image category; training a classifier, on them and then evaluating it on novel images, possibly of more challenging nature. The assumption is that training is a hard task that only needs to be performed once; hence the allocation of human resources to collecting a training set is justifiable. However, a constraint to current progress is the effort in obtaining large enough training sets of all the objects we wish to recognize. This effort varies with the size of the training set required, and the level of supervision required for each image. The paper first goes through the basic approach for learning any object, explaining the required background and the algorithms that are used in our learning system. Then it explains the implementation detail, telling how we have implemented our system in a outline. Latter we have shown our experiments and some results that are approximately the worst and the bases cases. In the end we have we have discussed the final result of our experiment and the potential future of our work.

II. APPROACH

Our approach is to download the images of a particular object using Google Image Search, using SIFT[10] feature detector to obtain features of those images, training a classifier for images using SVM classifier and classify the test images and assess the performance using observation data set containing random images[1][2][3]. There is a plenty supply of images available at the typing of single word using Internet image search engines such as Google. With increased quality of search results given by Google, images obtained using Google search engine are almost like pure training images: as many as 90% of the returned images may be visually related to the intended category. However the number of objects in the image is unknown and variable, and the pose (visual aspect) and the scale are uncontrolled. However, if one can succeed in learning using these images, the reward is tremendous: it enable us to automatically learn a classifier for whatever visual category we wish. Using this we can download more images with different increasing possible categories thus making this learning system as close as human in recognizing different objects

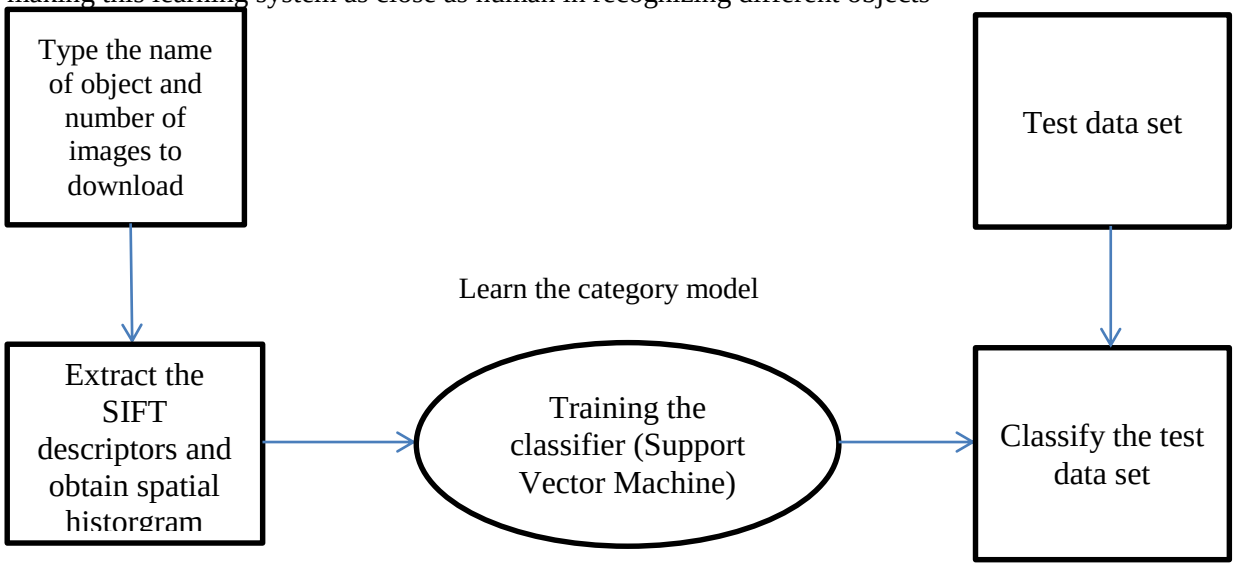

Figure 1: A summary of our approach.

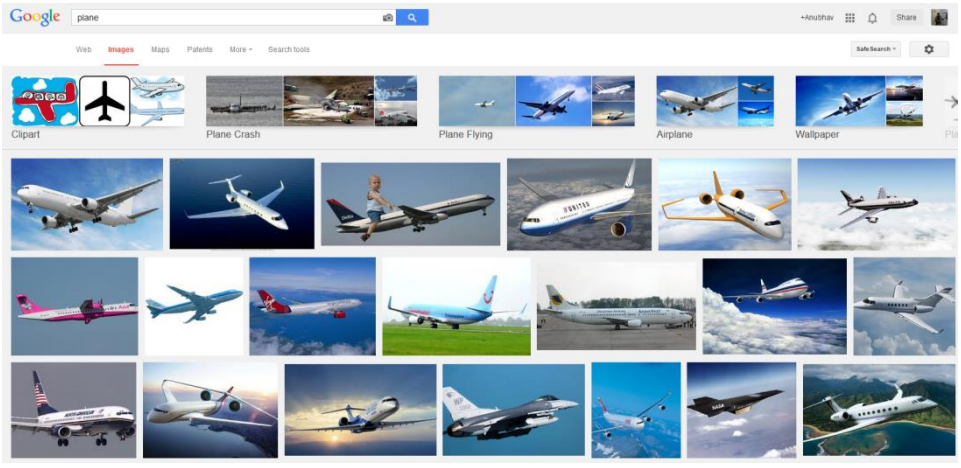

Figure 2: Images returned from Google's image search using the keywork "plane" . This is a representative sample for our training data. Note the large proportion of visually related images and the wide pose variation.

To further understand our approach we should review the SIFT for feature detection and SVM as a classifier.

2.1 SIFT (Scale Invariant Feature Transform)

A SIFT[10] feature is a selected image region (also called key point) with an associated descriptor. Key points are extracted by the SIFT detector and their descriptors are computed by the SIFT descriptor. It is also common to use independently the SIFT detector (i.e. computing the key points without descriptors) or the SIFT descriptor (i.e. computing descriptors of custom key points).

2.1.1 SIFT detector

A SIFT *key point* is a circular image region with an orientation. It is described by a geometric *frame* of four parameters: the key point centre coordinates *x* and *y*, its *scale* (the radius of the region), and its *orientation* (an angle expressed in radians). The SIFT detector uses as key points image structures which resemble "blobs". By searching for blobs at multiple scales and positions, the SIFT detector is invariant (or, more accurately, covariant) to translation, rotations, and re scaling of the image.

The key point orientation is also determined from the local image appearance and is covariant to image rotations. Depending on the symmetry of the key point appearance, determining the orientation can be ambiguous. In this case, the SIFT detectors returns a list of up to four possible orientations, constructing up to four frames (differing only by their orientation) for each detected image blob.

There are several parameters that influence the detection of SIFT key points. First, searching key points at multiple scales is obtained by constructing a so-called "Gaussian scale space". The scale space is just a collection of images obtained by progressively smoothing the input image, which is analogous to gradually reducing the image resolution. Conventionally, the smoothing level is called scale of the image.

2.1.2 SIFT Descriptor

A SIFT descriptor[9][11] is a 3-D spatial histogram of the image gradients in characterizing the appearance of a key point. The gradient at each pixel is regarded as a sample of a three-dimensional elementary feature vector, formed by the pixel location and the gradient orientation. Samples are weighed by the gradient norm and accumulated in a 3-D histogram *h*, which (up to normalization and clamping) forms the SIFT descriptor of the region. An additional Gaussian weighting function is applied to give less importance to gradients farther away from the key point centre. Orientations are quantized into eight bins and the spatial coordinates into four each, as follows:

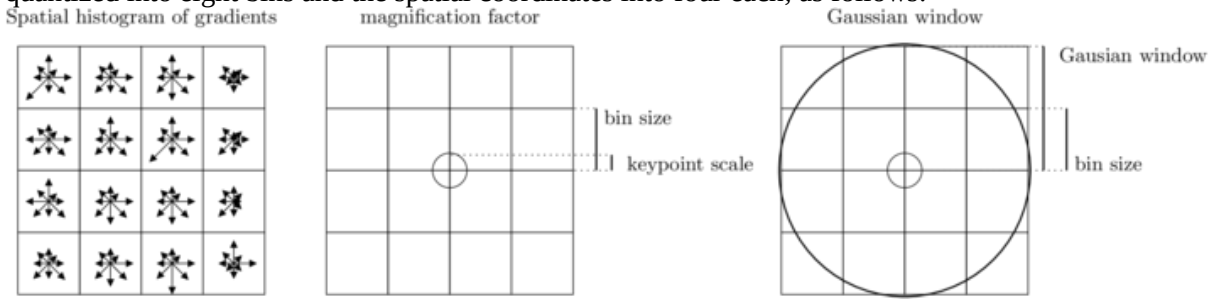

Figure 3: The SIFT descriptor is a spatial histogram of the image gradient.

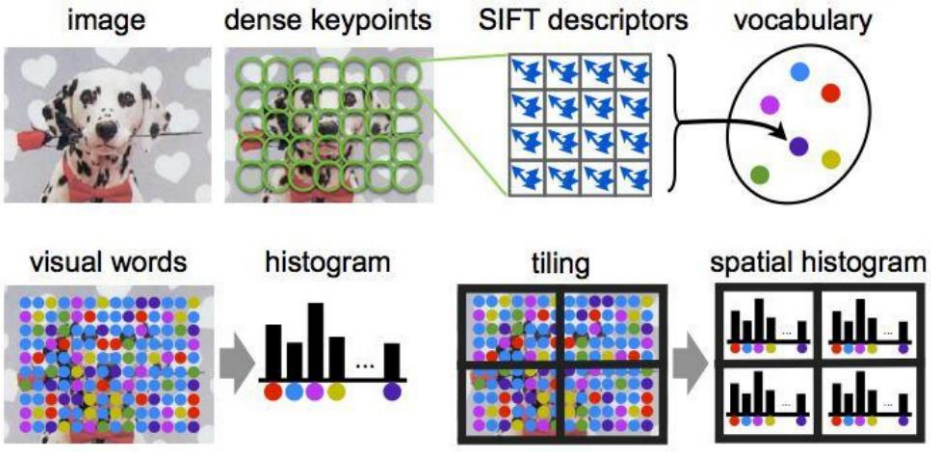

Figure 4: showing the SIFT descriptors to extract visual vocabulary and converting it into spatial histograms.

2.2 SVM

In our system we have used linear SVM[4]. Let $\mathbf{x} \in R^d$ be a vector representing, for example, an image, an audio track, or a fragment of text. Our goal is to design a classifier, i.e. a function that associates to each vector **x** a positive or negative label based on a desired criterion, for example the fact that the image contains or not a cat, that the audio track contains or not English speech, or that the text is or not a scientific paper.

The vector **x** is classified by looking at the sign of a linear scoring function $\langle \mathbf{x},\mathbf{w} \rangle$. The goal of learning is to estimate the parameter $\mathbf{w} \in R^d$ in such a way that the score is positive if the vector x belongs to the positive class and negative otherwise. In fact, in the standard SVM formulation the the goal is to have a score of at least 1 in the first case, and of at most -1 in the second one, imposing a margin.

The parameter **w** is estimated or learned by fitting the scoring function to a training set of n example pairs (x_i, y_i),**i=1,…,n**. Here $y_i \in \{-1,1\}$ are the ground truth labels of the corresponding example vectors. The fit quality is measured by a loss function which, in standard SVMs, is the hinge loss:

$$\ell i(\langle w,x \rangle) = \max\{0, 1 - y_i \langle w,x \rangle\} \quad (1)$$

Note that the hinge loss is zero only if the score $\langle \mathbf{w},\mathbf{x} \rangle$ is at least 1 or at most -1, depending on the label y_i.

Fitting the training data is usually insufficient. In order for the scoring function generalize to future data as well, it is usually preferable to trade off the fitting accuracy with the regularity of the learned scoring function $\langle \mathbf{x},\mathbf{w} \rangle$. Regularity in the standard formulation is measured by the norm of the parameter vector$\| \boldsymbol{w} \|^2$. Averaging the loss on all training samples and adding to it the regularized weighed by a parameter λ yields the regularized loss objective.

$$E(\mathbf{w}) = \frac{\lambda}{2} \| \boldsymbol{w} \|^2 + \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{max}\{0, 1 - yi\langle w, x \rangle\} \quad (2)$$

Note that this objective function is *convex*, so that there exists a single global optimum.

The scoring function $\langle \mathbf{x},\mathbf{w} \rangle$ considered so far has been linear and unbiased. Adding a bias discusses how a bias term can be added to the SVM and Non-linear SVMs and feature maps shows how non-linear SVMs can be reduced to the linear case by computing suitable feature maps.

III. IMPLEMENTATION DETAILS

3.1 Dataset

The experiments used 101 different object categories, from Caltech datasets[8] and we divided these into two datasets. The datasets were as follows:

3.1.1 Prepared training set

Manually gathered images from Caltech 101[8] image set were used.

3.1.2 Prepared test set

Manual gathered dataset of random images was used. To test the different between the training images and downloaded images.

3.1.3 Raw images downloaded using Google Image Search:

A set of definite number of images that is downloaded automatically by program using Google image search API. It returned the top images by the Google image search result.

3.2 Image Preparation

Images in form of spatial histogram were extracted using SIFT descriptor. A cache memory is maintained after processing of each training images, testing images and automatically downloaded Google images. This cache memory helps when the images are altered: If a new image is added its histogram file is made in the cache. This provide very fast information retrieval when the system is again started or object recognition is repeated as for the first time it takes a lot of time to process each image using SIFT.

3.3 Google Search downloader

Using Google Image Search API, images are downloaded. The user gives the program the name of the image he wants to download and also the number of images. The program then using Google Image Search downloads top images that appear in the Google Image Search. Thus storing them in system memory for processing.

3.4 Classifier

Linear SVM is used as a final classifier. Which is used to learn the images that are downloaded using Google Search downloader? SVM classify these images, thus adding these to the memory of application. The histograms are treated as representing each image as vectors normalized to unit Euclidean norm. VLfeat [5] library is used to implement linear SVM classifier.

3.5. Assessing the performance

We will measure the retrieval performance quantitatively by computing a Precision-Recall curve.

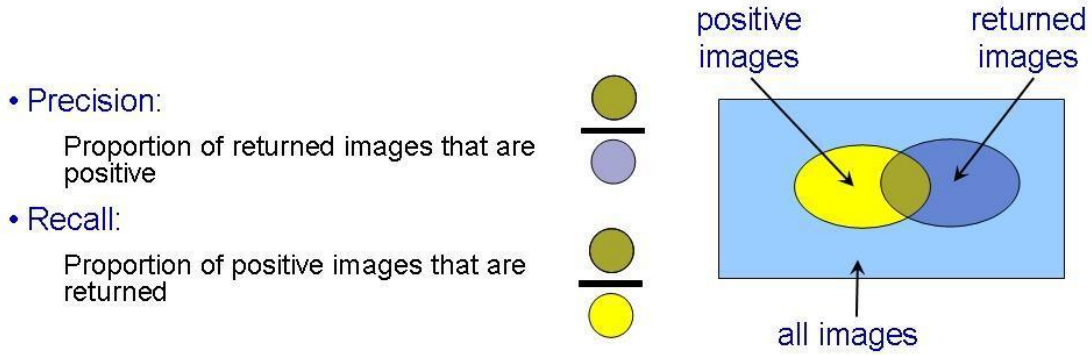

Figure 5: Demonstration of Precision Curve

The Precision-Recall curve is computed by varying the threshold on the classifier (from high to low) and plotting the values of precision against recall for each threshold value. In order to assess the retrieval performance by a single number (rather than a curve), the Average Precision (AP, the area under the curve) is often computed.

IV. EXPERIMENTS

Experiments were performed on several images of different category. First results using Google image search was done on simple and distinguishable images set like car, brain, airplane, and bike. Second result was using a little difficult to guess images like horse, buffalo, dog. The accuracy in the first case was really good as the structure of these objects is totally different. But, in the second case the accuracy was a little low because of animals with four legs like dog, buffalo, goat, have almost the same structure. These images were also added to the system by making a folder representing the name of the object. Thus it helped us to increase the number of objects the system can recognize by adding new objects to its memory. Due to usage of cache for storing spatial histograms of each images after the first run, the object recognition was done is no time when we again ran the system thus making it to recognize more objects along with good speed of processing for the future usage of images.

V. RESULTS

5.1 Objects with unique shape

Objects with unique shape like brain can easily be distinguished from other objects. The result of the object recognition using training images and automatically downloaded images were almost the same. The image below shows the output of the test images using Google Image Search downloaded images of brain.

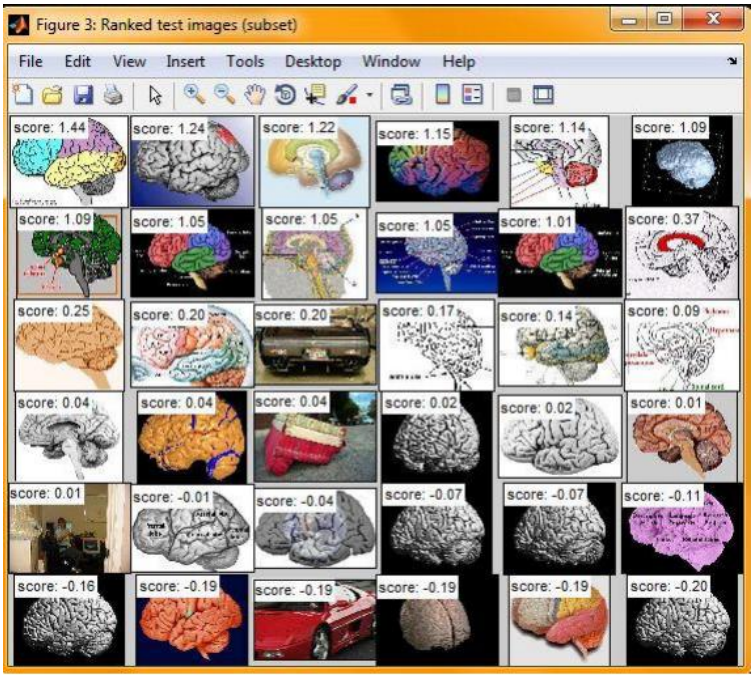

Figure 6: the output images using Google Image Search downloaded images as training data set. The number shows the correlation between the training images and tested images.

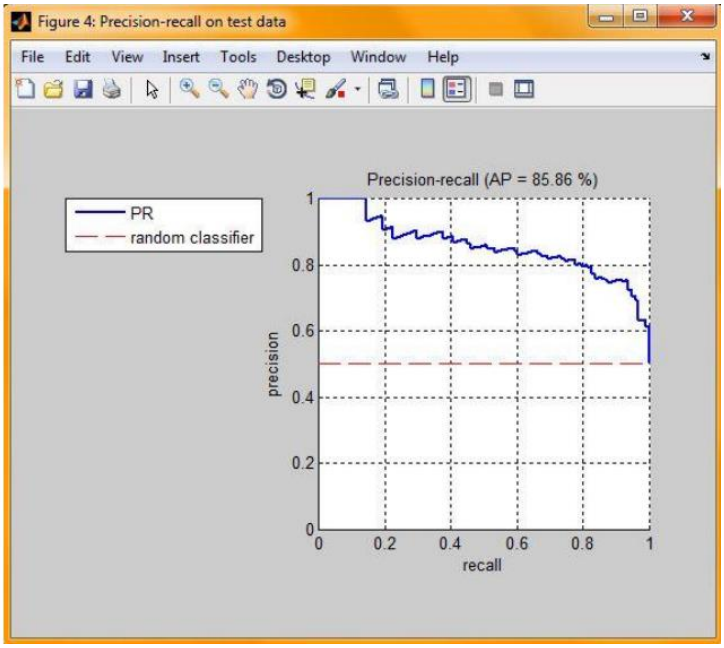

Figure 7: the curve of the output images. This shows that the accuracy of brain as an object recognition is 85.86%, which is pretty good.

5.2 Objects with common type shape

Objects with common shape like horses are not easily are distinguished from other objects. As a result the images that are returned by Google Image Search are not that close to the original object. Thus the accuracy of the object recognition decreases, but that is the case with us humans also: we fail to distinguish sometimes from a horse to a buffalo. Below, first is the result of using a data set from Cal. Tech. to train images and another is images data set obtained using Google Image Search. The accuracy in using the figures 8, 9, 10, 11, clearly shows the difference between two ways.

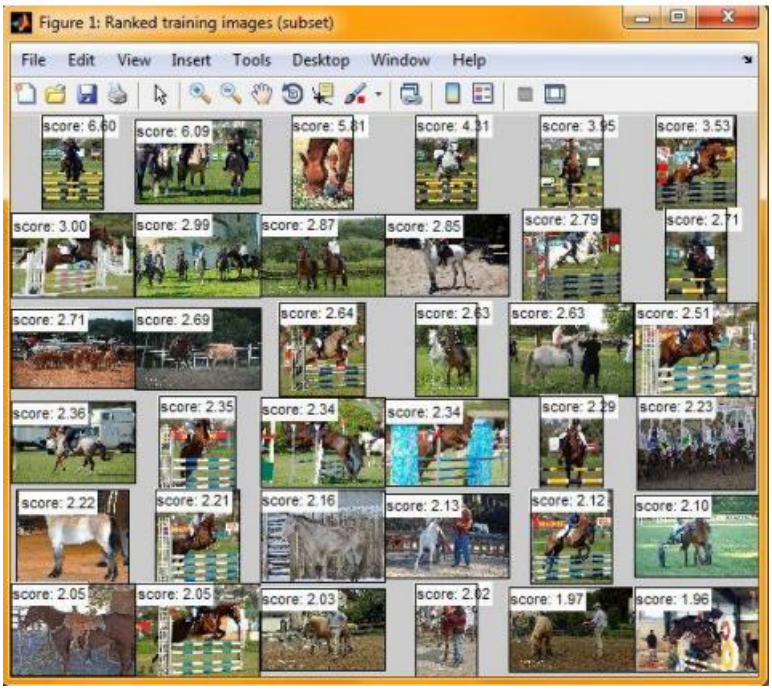

Figure 8: The output images using already existing horses' dataset from Caltech as a training dataset.

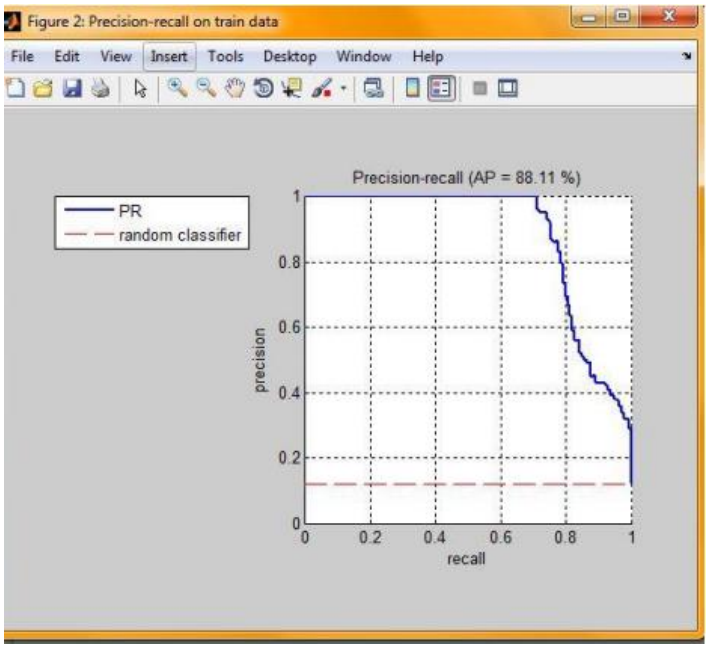

Figure 9: the curve of the output images. This shows that the accuracy of horse as an object recognition is 88.11%, which is expected for predefined training set of images.

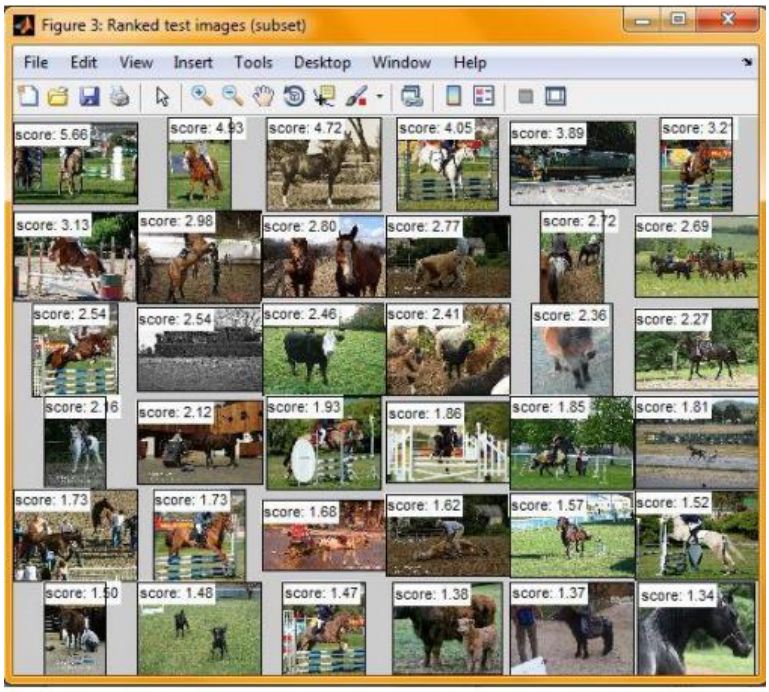

Figure 10: the output images using Google Image Search downloaded images as training data set. This clearly shows the difference in recognition using already existing training set and recognition using Google Image Search. The system recognized buffalo and even dog as horse.

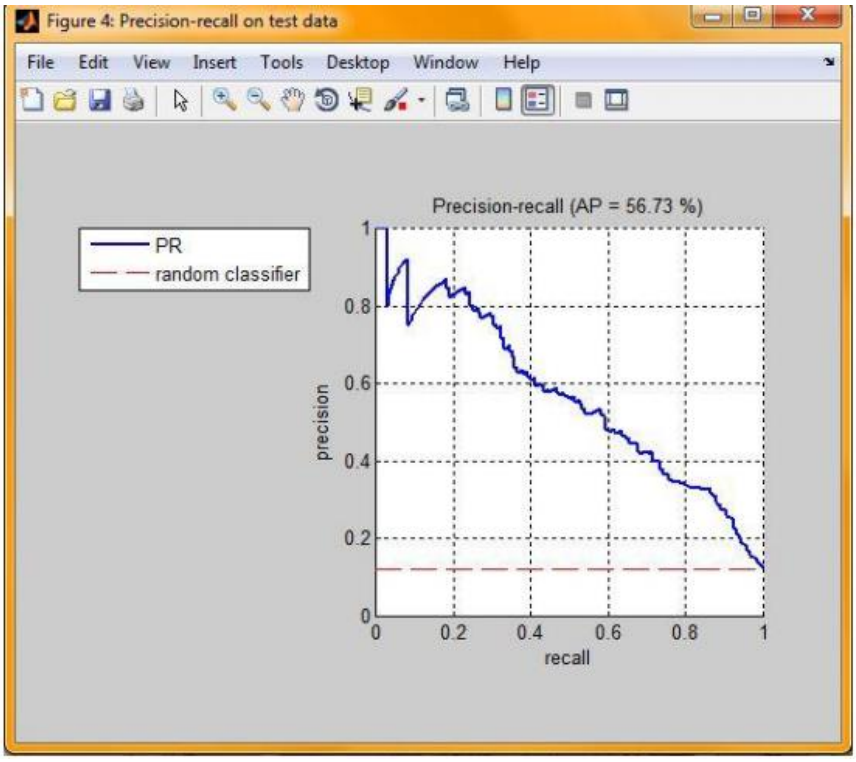

Figure 11: the curve of the output images. This shows that the accuracy of brain as an object recognition is 56.73%, which is not good enough, but resembles the nature of human in error.

VI. CONCLUSIONS

We have proposed a method, of training using the object name and number of images that we require to download with an image search engine like Google. We used Vlfeat[5] library for using their highly optimized SIFT and SVM algorithms. We ran our test on approximately 10000 images and we

used our system for recognition of 100 different objects using its own downloaded images. Our system was able to recognize successfully all of these 100 objects. The accuracy of object recognition depends on the particular shape of the object and also the number of object having the same shape as that object. For the objects having unique shape like aeroplane, helicopter, brain, etc. the accuracy of the recognition was ranging from 80%-92%, and for the objects not having unique shape like horse, dog, buffalo, etc. the accuracy of the recognition was ranging from 45%-70%. The training quality depends on the Image Search Engine. As Google is continuously working on increasing the quality of result in image search, the accuracy of system to recognize and learn new objects will increase. However there are many open issues: the choice the choice of features, the use of fixed background densities to assist learning leads to the reduction of noise in many images.

VII. FUTURE WORK

As we can see that our object recognition system is like a new child that tries to learn new objects by understanding the shape and features of many images of the same type, the accuracy is only better than average when we make the system to recognize an image from other images having same shape and features. We are currently working on increasing this accuracy when similar types of objects are difficult to differentiate. Our future work will be to make this system more like a grown up human being, which will be able to recognize objects with similar features and shapes using images that are available on internet. Moreover, this future system will be able to learn new objects without even taking input from user. It will be able to keep learning new objects and increase the number of objects it can recognize without any bound just like us, humans. We are working on deep learning for our future project[6][7].

ACKNOWLEDGEMENTS

We wish to express our profound sense of gratitude and indebtedness to Dr. H.L. Mandoria, Professor and Head, and Ratnesh Srivastava Professor of Department of Information Technology for imparting valuable guidance, help rendered to us during fabrication work and unstained encouragement rendered during the various phase of work. We would like to thank him for having confidence is us and trusting us in what we did. We would also like to thank Naveen Rature for helping us in fabricating the test model and making all the things needed to be available to us at great times of need. Here's big "Thank You" to all those who we involved in this project directly or indirectly and making this project a great success.

REFERENCES

[1]. S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part based representation. IEEE PAMI, 20(11):1475–1490, 2004

[2]. K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. JMLR , 3:1107–1135, Feb 2003

[3]. G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In Workshop on Statistical Learning in Computer Vision, ECCV, pages 1–22, 2004.

[4]. Support vector machine active learning for image retrieval. MULTIMEDIA '01 Proceedings of the ninth ACM international conference on Multimedia. ACM ISBN:1-58113-394-4

[5]. Vlfeat: an open and portable library of computer vision algorithms. MM '10 Proceedings of the international conference on Multimedia. ACM. ISBN: 978-1-60558-933-6.

[6]. Learning Deep Architectures for AI. Journal. Foundations and Trends® in Machine Learning archive Volume 2 Issue 1, January 2009.

[7]. What is the best multi-stage architecture for object recognition? Computer Vision, 2009 IEEE 12th International Conference. E-ISBN : 978-1-4244-4419-9

[8]. CALTECH 101 image data set. http://www.vision.caltech.edu/Image_Datasets/Caltech101/

[9]. K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "Fast inference in sparse coding algorithms with applications to object recognition," Technical Report, Computational and Biological Learning Lab, Courant Institute, NYU, Technical Report CBLL-TR-2008-12-01, 2008.

[10]. Real-time SIFT-based object recognition system. Mechatronics and Automation (ICMA), 2013 IEEE International Conference.

[11]. M. Ranzato, F. Huang, Y. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR'07), IEEE Press, 2007.

AUTHORS

Anubhav Singh received his B. Tech degree from GBPUAT, Pantnagar, Uttarakhand, and his key areas of research are Machine Learning and Computer Vision.

Sumit Pathak received his B. Tech degree from GBPUAT, Pantnagar, Uttarakhand, and his key areas of research are Networking and Semantic Web.