Automatic Image Annotation using a Hybrid Engine

Kaustubh Shivdikar
Department of Electrical Engineering
Veermata Jijabai Technological Institute
Mumbai, India
Email: kaustubh.c.shivdikar@ieee.org

Ahan Kak
Department of Electrical Engineering
Veermata Jijabai Technological Institute
Mumbai, India
Email: ahan.kak.IN@ieee.org

Kshitij Marwah
MIT Media Lab
Massachusetts Institute of Technology
Cambridge, Massachusetts
Email: ksm@mit.edu

***Abstract*—Abstract Humans can view an image and immediately determine what the image is trying to convey. While this may be an easy event for humans, it is still considerably difficult for a computer to understand of its own accord. The challenge broadly lies in developing an automatic process to complement and supplant human visual and neural systems. In this paper, we address the core issue of imparting an image the ability to caption itself automatically. We propose a hybrid engine that utilizes a combination of feature detection algorithms coupled with context-free grammar to create a model that serves to semantically and logically describe an image in its entirety. Our hybrid engine model has an F1 score of 94.33% and a unigram score of 75% when evaluated on a novel dataset trained on human-annotated images.**

I. Introduction

Humans and computers interpret images very differently. For humans an image is a representation of an act, situation, memory or more, while, for computers the same image is merely an array of numbers. For example, a set of integers arranged in a circular form might represent a boy playing with a ball for us, but are just intensity values for machines. In this paper we present a novel framework to impart the ability for an image to speak for itself.

Self-identifying an image broadly involves two steps. As a first step we attempt to discern between visually similar objects using lines and strokes inherent in the image. We can then compare it with an existing database to find the degree of coherence between the elements of the image present and the database. The second aspect of the challenge involves using key terms to form sentences that caption the image accurately. Depending on the terms obtained, we utilize natural language processing techniques to generate a large number of sentences. These sentences are then compared with an exhaustive database of books in order to shortlist and pick those sentences that are logically correct; thereby generating a caption that, along with imbibing logical accuracy also describes the image in the best possible manner.

In this paper, we present a model that is versatile enough to be adapted across the fields of photography, military and surveillance. Specifically, we develop a hybrid engine that can be used for automatically tagging and understanding images with applications to annotating goods in a warehouse, automated parsing of surveillance data, identifying enemy aircraft and vessels without human intervention and self-aware systems.

A. Benefits and Contributions

Specifically, we provide the following contributions:

- A flexible set of self-learning algorithms that learn from training databases and can be deployed onto a wide range of platforms.
- A hybrid engine for image annotation characterized by a dual feature detection algorithm and the use of context-free grammar to generate image captions.
- A detailed analysis of the parameters and performance of the feature detection and context-free grammar algorithms used.
- Finally, we show that our hybrid model achieves an F1 score of 94.33% and a unigram score of 75% which is better than the current state of art.

B. Overview of Limitations

The object detection algorithm trains on a given data-set. Thus, the quality and the diversity of the training set will have an impact on the results. While the database, which consists of a wide variety of books, performs satisfactorily in generic cases as evidenced in Fig. 1; it can become a limiting factor in the case of niche images. This can be overcome by increasing the training set to include as much variety as possible.

II. Related Work

A. Feature Detection

Along with optical character recognition, feature detection has been one of the first computer vision problems. Feature detection can be based on detecting image edges [2], corners [3], a combination of the two [4] or blobs. Chief among the blob detection methods are the Scale Invariant Feature Transform (SIFT) [5] and the Speeded up Robust Features (SURF) [6] algorithms. A comparison of these method reveals that no single method is suited for all kinds of images [7], with Mikolajczyk and Schmid even suggesting the use of a combination of different methods in order to achieve better results in terms of feature detection [8]. Accordingly, we utilize the corner detection method based on the minimum eigenvalue theorem as outlined by Shi and Tomasi [3] and the SURF feature algorithm presented by Bay, Tuytelaars and Van Gool [6].

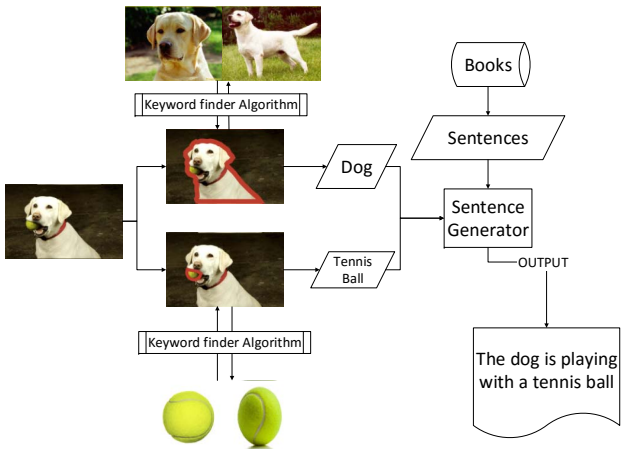


Fig. 1. Process flow. The hybrid algorithm determines the objects present in the image and the natural language processing schema generates a caption for the image.

B. Natural Language Processing and Image Annotation

A combination of feature detection and natural language processing has been long utilized for image annotation. Methods used can involve a comparison of databases of images and sentences to find the best possible match as described by Farhadi et al. [9], the use of relevance models [10], or the use simultaneous classification and annotation [11] among others. Other methods delve into using visual sentence templates [12] and utilizing probabilistic determination in order to caption images based on text surrounding them as described by Feng and Lapata [13][15][16].

Vinyals et al. [17] came up with a novel approach based on Convolutional Neural Networks (CNN) for vision and Recurrent Neural Networks (RNN) for sentence generation. Gaining on the BLEU-4 score as compared to others, when evaluating on multiple data-sets viz Flickr 8k, Flickr 30k, COCO, SBU, etc. However, they haven't yet made the algorithm robust enough to cope up with unsupervised data-sets, leaving scope for further work in the domain.We present a model that utilizes natural language processing in tandem with feature detection algorithms and k-means clustering [14] in order to find the best possible caption for a given image. The advantage presented by such a model is that it does away with the compromise resulting from an attempt to fit an image in a set database of pre-constructed sentences [9].

III. System Architecture

In this paper, we propose a hybrid engine that combines the SURF and minimum eigenvalue algorithms as outlined in Fig. 2. The motivation for a dual algorithm model stems from the fact that no single feature detection algorithm and descriptor is suited for a wide variety of images [18].

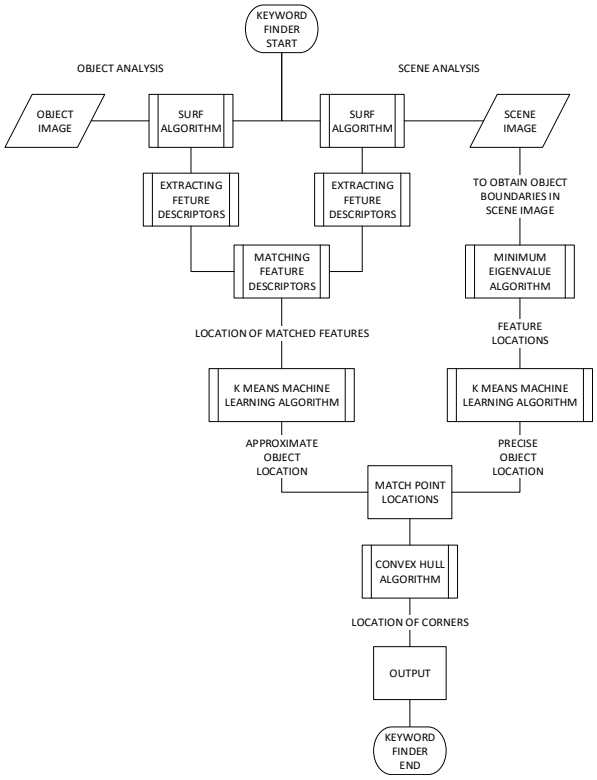


Fig. 2. Keyword Finder Algorithm process flow.

A. The SURF Algorithm: Determining the location of matched features

Preliminary feature extraction and detection is done using the SURF algorithm and forms the first stage of the proposed hybrid engine. The SURF algorithm uses a Hessian matrix approximation for feature (blob) detection. As described by Bay et al. [6], given a point x = (x, y) in an image I, the Hessian matrix H(x,) in x at scale is defined as follows:

$$H\left(x,\ \sigma\right) = \begin{bmatrix} L_{xx}\left(x,\sigma\right) & L_{xy}\left(x,\sigma\right) \\ L_{xy}\left(x,\sigma\right) & L_{yy}\left(x,\sigma\right) \end{bmatrix} \tag{1}$$

where $L_{xx}(x,\ \sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial x^2} g(x)$ with the image I in point x, and similarly for $L_{xy}(x,\ \sigma)$ and $L_{yy}(x,\ \sigma)$. The model we present here applies the SURF algorithm multiple times. The first iteration is run on the image we wish to annotate, hereby referred to as the scene image followed by subsequent iterations on the image database, so as to obtain interest point for each image in the database, hereby referred to as the object image.

The model we present here applies the SURF algorithm multiple times. The first iteration is run on the image we wish to annotate, hereby referred to as the scene image followed by subsequent iterations on the image database, so as to obtain interest point for each image in the database,

hereby referred to as the object image. Once the key points of each object and the scene image have been obtained, feature matching is applied to detect the presence of a particular object within a scene. The extracted feature points are then be vectorially characterized by feature descriptors.At the end of this stage, the presence of particular object in the scene has been identified based on matching descriptors. However, no feature matching algorithm is foolproof, and there exist certain false positives that don't represent an object and thus must be removed.

B. k-means Clustering: Making the Hybrid Engine Robust

In order to ensure that the matched points represent the position of the desired object accurately, k-means clustering [14] is used. The k-means clustering algorithm is used to group a definite number of data points into a given number of clusters based on relative positioning of the observations from each other. The purpose of this algorithm is to minimize:

$$H\left(x,\ \sigma\right) = \begin{bmatrix} L_{xx}\left(x,\sigma\right) & L_{xy}\left(x,\sigma\right) \\ L_{xy}\left(x,\sigma\right) & L_{yy}\left(x,\sigma\right) \end{bmatrix} \tag{2}$$

where x is a set of data points, k is the total number of clusters formed by the algorithm, s is the set of clusters formed, and u_i is the mean of points in S_i, such that the variance of a particular cluster is minimum. Once the matched points have been obtained, we run the k-means algorithm onto the location of these points. As the density of the matched feature SURF descriptors will naturally be higher at the true location of object we choose the cluster with highest density to extract the information of the location of the object. Stage one of the hybrid engine concludes with the location of the object being detected.

C. The minimum eigenvalue algorithm: Laying the foundation for detection of object boundaries

Now that the object has been located, stage two involves determining the object boundary. The corner features within the image are detected using the minimum eigenvalue algorithm developed by Shi and Tomasi [3] which considers a weighted auto-correlation:

$$\arg min_S \sum_{count=1}^{k} \sum_{x \in S_i} ||x - \mu_{count}\ || \tag{3}$$

where $I(x_i)$ represents an image coordinate and $w(x_i)$ represents the weight of that coordinate. As was the case with the SURF algorithm, the minimum eigenvalue is run multiple times. The first iteration is run on the scene followed by subsequent iterations on the object database, so as to obtain corners for each image in the database. The corner descriptors are obtained using the Fast Retina Key (FREAK) algorithm.

D. FREAK Descriptors and the return of k-means clustering: Detecting object boundaries

The FREAK feature descriptor, which works on the principle of Difference of Gaussian (DoG) much like the human eye, developed by Alahi, Ortiz and Vandergheynst [19] is utilized by us in the model proposed in this text. The FREAK descriptor is used to extract the corner points obtained by the minimum eigenvalue algorithm, the descriptor for each point is then subjected to the k-means algorithm, thus outputting the clusters of each individual object separately from the scene. We now have a set of clusters that represent the position of the objects present in the scene, and another set of clusters that contains information regarding the boundary of each object. When the two sets of clusters are mapped on to each other, we get a dataset that represents the objects that are present in the scene along with the boundary for each object. The final step involves using the convex hull algorithm to draw a polygon around each object present, in other words, each polygon represents a key term, which is stored in a text file. Thus at end of stage two, we have successfully detected the location of the object in the scene and have identified its boundaries, using our hybrid engine.

E. Context-Free Grammar (CFG): Generating Sentences from acquired Key Terms

A CFG is a 4-tuple, consisting of a set of non-terminals (N), a set of terminals (), a set of rules (R) and a start symbol (S). A rule set can be expressed in terms of a sentence structure. One such sentence structure is shown in Fig. 3. In CFG, a complete sentence can be obtained by beginning with the start symbol, S, and repeatedly replacing a variable X, with the ones below it. For example in the structure shown in Fig. 3, language constituents such as NP, VP, PP etc. form the non-terminals while words like A, $placed$, on form the terminals. S is divided into NP and VP, both of which are to be compulsorily used while formulating a sentence. NP is further expressed as Det and N, while VP is expressed by either $placed$ or PP or a combination of the two. PP is further compulsorily divided into P and NP. Keeping this structure as a reference, it would be safe to conclude that a sentence generated by the grammar model presented above would contain Det, N and $placed$ at least.

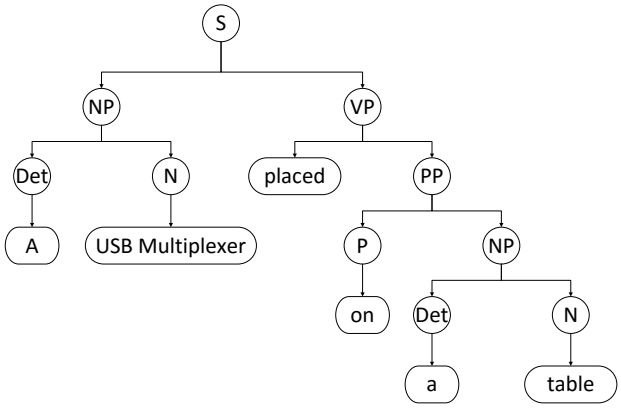


Fig. 3. Sentence structure used in the model. N is obtained from the text file input to NLTK while Det, VP and P are predefined.

The set, thus generated, consists of sentences that are grammatically correct but may lack logical accuracy; and therefore must be compared against a database of textbooks to weed out incorrect sentences. The entire process flow is described in Fig. 4, after which a sentence, or a set of sentences is obtained that describes the scene in an accurate manner.

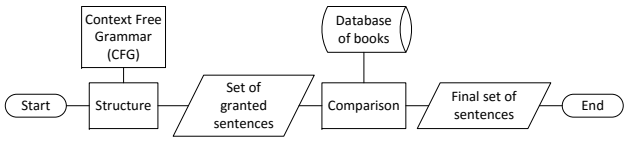


Fig. 4. Natural Language Process Flow

IV. Analysis

In this section, we compare the SURF and the minimum eigenvalue algorithm, explain why a combination of the two works best, obtain the F1 score [20] for object scene pairs at different orientations with the vertical and calculate the BLEU score [21] for the natural language processing algorithm.

A. Comparing the SURF and minimum eigenvalue feature detectors

Speaking from a purely theoretical standpoint, the SURF algorithm should always detect a far greater number of features vis--vis the minimum eigenvalue algorithm, however, as Fig. 5 illustrates, this is not always the case.

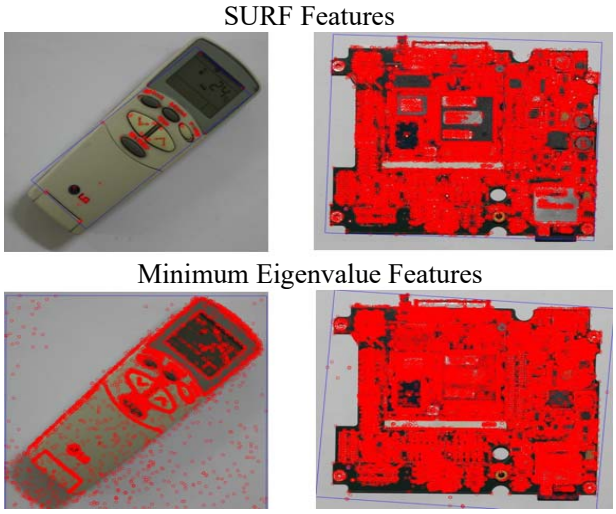


Fig. 5. SURF and Minimum Eigenvalue algorithms applied to the same set of images separately. The object on the left has fewer grayscale variations and therefore lesser SURF features as compared to the object on the right. The minimum eigenvalue algorithm works well for both objects, returning a satisfactory number of features in each case.

If the image in question exhibits less grayscale variations, the number of features as detected by SURF drastically reduce, whereas for an image containing substantially greater grayscale variations, both algorithms perform equally well. It is safe to conclude that the minimum eigenvalue algorithm does a better job of detecting object boundaries, however, this would lead us to incorrectly assume that it is the better of the two. From an image processing standpoint, the ultimate aim is to not detect object boundaries but rather detect the presence of objects within a scene. The reasons that cause such a premise to fail have been explored in the following section, a comparison of the SURF and FREAK descriptors.

B. Comparing the SURF and FREAK descriptors

It is apparent that the SURF algorithm performs exceedingly well when it comes to matching features while features matched based on FREAK descriptors are not very accurate. From our discussion earlier, the fact that the minimum eigenvalue algorithm is good at detecting boundaries has been documented, and now, we conclude that the SURF algorithm is good at matching features. Object detection within a scene requires both feature matching as well as detection of object boundaries in order to reduce the number of false positives, leading us to the hybrid model presented herein that utilizes both SURF as well as the minimum eigenvalue algorithms. Applying the same to the image pair of Fig. 5, the results obtained are presented in Fig. 6.

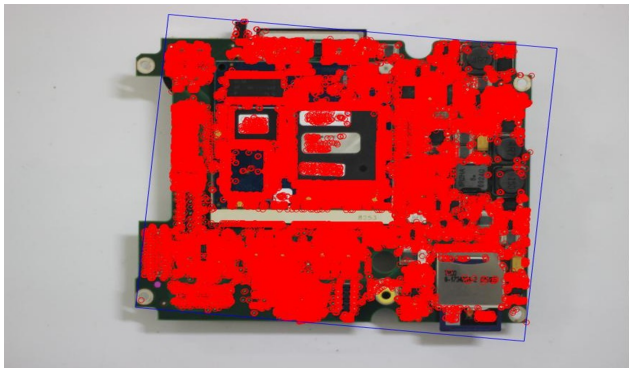


Fig. 6. A combination of the SURF and Minimum Eigenvalue algorithms applied to the object of Figure 6. The object boundary is clearly defined along with the presence of a large number of features.

The hybrid algorithm is not only able to appropriately select the object from scene but also able to detect its boundaries. The locations points obtained after running the SURF feature matching algorithm are further used to select the best clusters of points of minimum eigenvalue features. This allows the target image used for the purpose of analysis to have higher tolerance with respect to changes in reference to orientation, scale and surrounding lighting without compromising on the efficiency of detection.

C. Calculating the F1 score

The F1 score helps analyze the robustness of the object detection algorithm. Three cases corresponding to a polar angle, ϕ, of $0°$, $45°$ and $60°$ respectively have been considered and the luminous intensity and F1 score for each is calculated. The polar angle here is defined as the rotation of camera angle in-order to capture the image. The weightage values being selected are with reference to Digital Consultative Committee

for International Radio 601. From an ideal standpoint, the F1 score should have reduced gradually with increase in the angle with the vertical (polar angle), but our analysis reveals that this is not the case.

TABLE I
POLAR ANGLE AND LUMINOUS INTENSITY

Polar Angle (ϕ)	Average Luminous Intensity	F1 Score
0°	6.253280 %	94.339623 %
15°	6.233420 %	92.314827 %
30°	6.215880 %	86.401929 %
45°	6.208900 %	85.714286 %
60°	6.245270 %	87.500000 %

There exist two possible explanations for this anomaly:

- As can be observed from Table 1, post the 45 mark, the average luminous intensity has increased resulting in an increase in the number of features, which in turn could correspond to a greater F1 score.
- The object in the scene with a polar angle of 60 is much sharper than that in the image with a polar angle of 45°, probably because of a shift in the camera angle in the x-y plane.

D. Calculating the BLEU score

In order to judge the performance of the natural language processing algorithm presented herein, the sentences generated by the algorithm need to be compared with sentences a human would form. The Bilingual Evaluation Understudy (BLEU) score enables us to do this. A higher BLEU score corresponds to a greater degree of closeness between the machine generated and human formed sentences. Considering Fig. 8, which represents a number of items, including ICs, batteries and a USB multiplexer placed on a table. Some of the ideal reference sentences for this scene can be listed as:

- This is a cluttered scene consisting of ICs, batteries, USB multiplexer and a bunch of wires lying on a wooden desk.
- This is a cluttered wooden desk with ICs, batteries to the left of a bunch of wires.

For the same scene one of the many sentences generated by our algorithm was, "A USB multiplexer is placed on a table." We were able to obtain an Unigram score of 0.75 for our algorithm showcasing that database indicates that the machine translation is sufficiently adequate in terms of the information it contains.

E. Comparative Analysis

Having calculated the F1 and BLEU scores, we now proceed to compare our hybrid engine with other image annotation models. For a novel dataset consisting of human-annotated images, our hybrid engine performs fairly well as is evidenced by its BLEU and F1 scores.

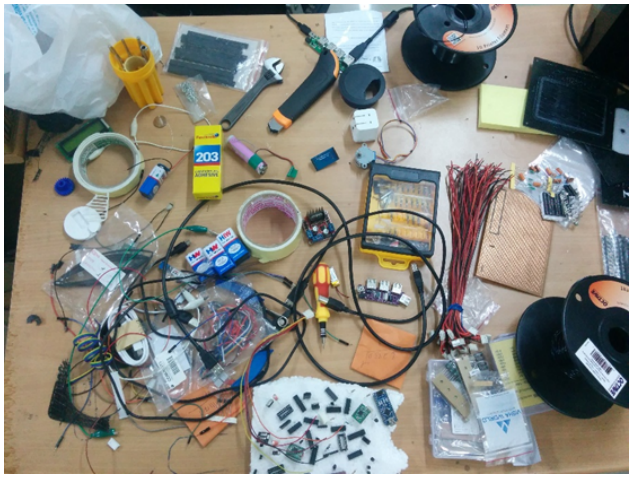
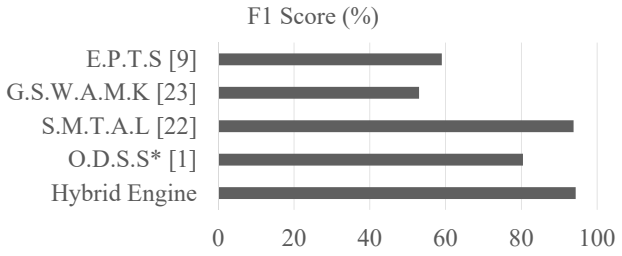
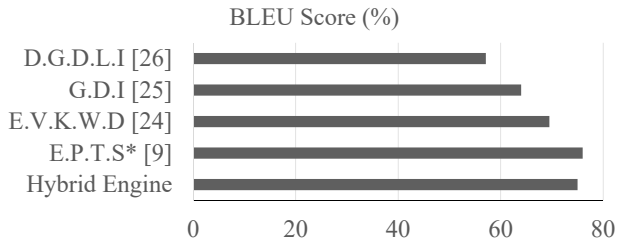


Fig. 7. A collection of items, namely ICs, batteries and a USB multiplexer placed on a table.



*For Object Detection using SURF and Superpixels, we have chosen their reading for the image with the highest variation.

Fig. 8. Comparative Analysis - F1 Score



*Every picture tells a story uses a BLUE scale instead of a BLEU utilizing triplets of words <object, action, scenes> for sentence generation, which is a probabilistic model that and serves disadvantage for their method.

Fig. 9. Comparative Analysis - BLEU Score

V. RESULTS

After evaluating the efficiency of our model on standard scales we observe some of the experimentation results derived after testing our hybrid engine annotator on a human evaluated database. Depicted in Fig. 10 are three different kinds of result sets obtained. Every image is associated with a sorted k-means cluster. We observe that there does exist a relation between the

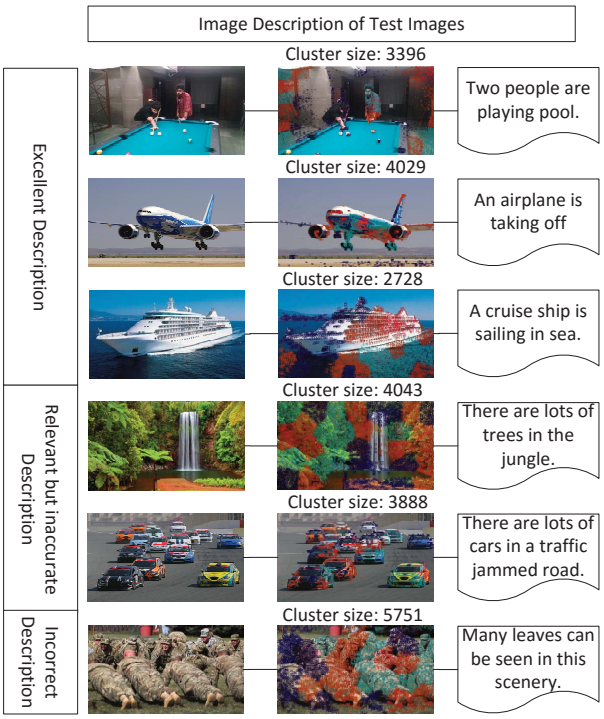


Fig. 10. Results obtained for different kinds of images. Certain types of images fare much better than others.

density of the cluster and spatial variations. The first set of images include a limited number of objects that are distinctly separated from one another as well as the environment. In the second set of images there exists a lot more environmental noise, moreover the objects are close to each other thus affecting the efficiency of the algorithm.Presence of noise notwithstanding, the algorithm performs poorly on the last image set. The misleading nature of the scenes cause the algorithm to have a higher match-metric with a completely irrelevant images thus generating incorrect results.

VI. Conclusion

Exploring the field of pattern recognition and machine learning, we in this work, not only present a hybrid method of object detection but also implement it in an algorithm of image annotation. Up to this point, work in the field of automated image annotation has focused on relying on a single feature detection method and has, consequently, suffered from the same pitfalls as the feature detection method itself [1][5][6].The hybrid model presented by us combines the strengths of two different feature detection algorithms and in doing so we have created what could well form the base for increasingly complex and accurate neural network algorithms. The CFG based sentence generator and evaluator not only verifies grammatical fluency but also checks for logical accuracy. The robustness of our hybrid model has been evaluated in terms of its good F1 score and an excellent score on the BLEU scale.

The model can be further extended to incorporate deep learning so as to do away with the changing nature of output brought by single layer learning. We would also like to test and improve our NLP algorithm to work for longer n-grams. Finally, as we enter into a new era of artificial intelligence with machine learning at its core, works like these are a step towards making computers more intelligent and thoughtful.

References

[1] M. Lopez-de-la-Calleja et al., "Object Detection Using SURF and Superpixels, Journal of Software Engineering and Applications, vol. 6 no. 9, pp. 511-518, 2013.
[2] J. Canny, A Computation Approach to Edge Detection, IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-8, no. 6, pp. 679-698, Nov. 1986.
[3] J. Shi and C.Tomasi, Good Features to Track, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, 1994, pp. 593-600.
[4] C. Harris and M. Stephens, A Combined Corner and Edge Detector, in Proceedings of the 4th Alvey Vision Conference, Manchaster, 1988, pp. 147151.
[5] D.G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, Int. J. Comput. Vision, vol. 60, issue 2, pp. 91-110, Nov. 2004.
[6] H. Bay et al., Speeded-Up Robust Features (SURF), Comput. Vis. Image Und., vol. 110, issue 3, pp. 346-359, Jun. 2008.
[7] T. Deselaers et al., Features for image retrieval: an experimental comparison, Inf. Retr., vol. 11, issue 2, pp. 77-107, Dec. 2007.
[8] K. Mikolajczyk and C. Schmid, A Performance Evaluation of Local Descriptors, IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 10, pp-1615-1630, Oct. 2005.
[9] A. Farhadi et al., Every Picture Tells a Story: Generating Sentences from Images, in The 11th European Conference on Computer Vision, Herkalion, 2010, pp. 15-29.
[10] J. Jeon et al., Automatic Image Annotation and Retrieval using Cross Media Relevance Models, in Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, Toronto, 2003, pp. 119-126.
[11] C. Wang et al., Simultaneous Image Classification and Annotation, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, 2009, pp. 1903 1910.
[12] G. Kulkarni et al., BabyTalk: Understanding and Generating Simple Image Descriptions, IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 12, pp-2981-2903, Dec. 2013.
[13] Y. Feng and M. Lapata, How Many Words is a Picture Worth? Automatic Caption Generation for News Images, in Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, 2010, pp. 1239-1249.
[14] J. MacQueen, Some methods for classification and analysis of multivariate observations, in Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkerley, CA, 1965, pp. 281-297.
[15] V. Ordonez et al., Im2Text: Describing Images Using 1 Million Captioned Photographs, in Proceedings of Neural Information Processing Systems, Granada, 2011.
[16] A. Aker and R. Gaizauskas, Generating image descriptions using dependency relational patterns, in Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, 2010, pp. 1250-1258.
[17] O. Vinyals et al,, Show and Tell: A Neural Image Caption Generator, arXiv preprint arXiv:1411.4555, 2014.
[18] J. Bauer et al., Comparing several implementations of two recently published feature detectors, in Proceedings of International Conference on Intelligent and Autonomous Systems, 2007.
[19] A. Alahi, "FREAK: Fast Retina Keypoint", in IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 510 517
[20] K. Dembczynski, An Exact Algorithm for F-Measure Maximization, in Proceedings of Neural Information Processing Systems, Granada, 2011.
[21] K. Papineni et al., BLEU: a Method for Automatic Evaluation of Machine Translation, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Philadelphia, 2002, pp. 311-318.