A SCALABLE CONTENT BASED VISUAL MEDIA RETRIEVAL SYSTEM

Ambareesh Ravi, Amith Nandakumar, Aishwarya Krishna Allada, Sandhya Manasa Gunda
Department of ECE, University of Waterloo, Waterloo, ON, Canada

ABSTRACT

Our research proposes a novel, efficient and scalable content based visual media retrieval system. We focus on creating an comprehensive architecture by leveraging the power of Deep Learning to enable modularization and efficiency for various retrieval applications that require scalability as their key requirement. The proposed novel architecture is flexible to work for both images and videos conjointly. We also put forward the results of two important experiments - the impact of colors on retrieval tasks and fast comparison techniques that may have a considerable impact in improving retrieval tasks further.

Index Terms— Convolutional Neural network (CNN, ConvNets), 3D Convolutions (C3D), Long Short Term Memory networks (LSTMs), Content Based Image Retrieval (CBIR).

1. INTRODUCTION

The recent developments in the field of digital media has led to abundance of visual media around us. Organizing and managing the visual data given the volume has become difficult. Hence visual search has become a crucial component in the field of computer vision research. The need for an automated approach to index, categorize and organize visual media without any human intervention is prevalent in various fields ranging from medicine, satellite imagery, digital libraries etc. The contemporary research in the field of retrieval tasks are due to the fact that metadata based retrieval tasks require a lot of human effort in terms of annotations, tag generation and they work satisfactorily only when the input query match with the tags available in the database. Whereas in content based techniques, the gist in the input is directly correlated with the images in the database. The crux of the solution is to generate features which is the visual vocabulary or embedding for every image in the large database and compare it with the features of the test image.

Deep Learning has improved over the years and have proven to be better at complex tasks over traditional approaches [4] and has the potential to connect the semantic gap between low level features such as shape, edges, color, texture, orientation etc. and high level features like context through abstraction and understanding the context in the content of the image as perceived by the humans.

We propose a light-weight, efficient and end-to-end deep learning architecture for content based retrieval tasks which can be used for both images and videos. The architecture utilizes a convolutional network for feature extraction and a recurrent network to operate on the sequence of frames. We have also provided substantiating evidence on the concordant performance of our architecture in comparison with the counterpart which is a 3D convolutional network. We perform an experiment to determine the effect of color on the performance of convolutional network on image retrieval tasks. We also give insights on techniques to improve the retrieval process.

2. LITERATURE REVIEW

2.1 Retrieval works

We review all the related works to visual retrieval tasks in this section. The major areas of research for content based retrieval tasks are global learning, feature representation, comparison metrics, faster comparison architectures etc.

2.1.1 Traditional approaches

Ladahke et al. [1] provide a review of different retrieval tasks with their applications and early approaches which led to the development in the field as of now. The authors also illustrate the basic technique for CBIR tasks. In [2], the authors Csurka et al., compare classifiers for visual categorization which is one of the earliest works in the field of CBIR after which improved approaches started emerging.

2.1.2 Deep Learning approaches

In his thesis [4], the author A.V. Singh elaborates on the differences between traditional and learning based approaches and proves that deep learning based approaches perform well over the traditional ones with their ability to bridge the semantic gap considerably. [5] provides a comprehensive study on deep learning for image retrieval tasks by putting forward some compelling arguments on the sub-par performance of traditional methods over learning based methods. In [6], A. Gordo et al., propose an approach to learn global representation for instance level retrieval using deep learning. There are several papers [7 - 11] which provide the best practices for content based retrieval tasks which are very helpful and insightful.

2.2 Optimization

A few works [12-16] offer intuitive techniques for optimization and scalability of retrieval tasks such applying PCA for dimensionality reduction of feature representation or

faster comparison time, faster indexing in retrieval tasks, applying different clustering methods for easier matching and retrieval, approximation of user input queries etc.

3. DATASETS

We use three datasets for our proposed system. We use Oxford5k [18], a buildings dataset and CIFAR10 [29] a dataset containing 10 classes of 60,000 tiny images for the experiment to determine the effect of color on retrieval tasks. We train the models on CIFAR10 – both color and grayscale models and test them for retrieval on Oxford5k. For the video retrieval approach we use the KTH Recognition of Human actions (RHA) [17] which is a video dataset containing 6 classes and about 100 videos per class.

4. METHODOLOGY

This section elaborates our proposed approaches, the intuition behind them and their significance. We divide this section into three parts – proposed novel architecture, experiments and finally faster retrieval. In general, a retrieval task involves comparison of input's feature representation with that of the data available in the database.

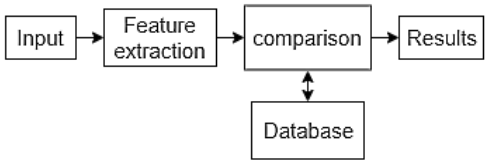

Figure 4.1 General overview of retrieval systems

4.1 Proposed novel architecture

We introduce a *novel* deep learning architecture- ***MobileNet V1 + LSTM*** that can work both on images and videos. The key features of this architecture are modularization, scalability, versatility for any applications etc. This architecture consists of co-existing plug-and-play models that the user can readily select depending on the application and data. The architecture is shown in Figure 4.2 below:

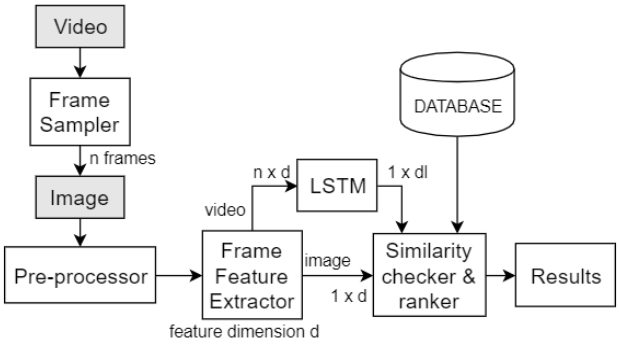

Figure 4.2 Proposed novel architecture

4.1.1 Working

A video is a collection of strongly correlated frames or images. In this method, the input video is segmented into chunks. Each chunk is sampled to *'n'* number of frames which decides the time-steps of the LSTM network [24]. Based on heuristics [22], we have selected *16 frames.* The pre-processed sampled frames are sent to the *feature extractor that learns the spatial features.* If the input is an image, it is pre-processed directly and sent to the feature extractor. In pre-processing, the frame is resized, color space, intensity, saturation etc. are modified as required by the feature extractor. The feature extractor can be of any type – traditional extractor like SIFT, SURF or a CNN model trained on similar data. We use MobileNet V1 [25] architecture trained on ImageNet [27] dataset as it is light-weight and the data is diverse for the network to learn in a generic manner as we target generalization. We extract the features before the fully connect layer i.e. after the *global average pooling* layer from the MobileNet model which gives features of dimension 1x1024. For an image, we use the feature representation to compare with the feature vectors of the images in the database. For a video, the features of all sampled frames are aggregated to a multi-dimensional array of size 16x1024. This array is sent to an LSTM network with one frame's feature at a time step of unrolling that can *learn the temporal correlation (features)*. This model is inspired from LRCCN [22] which was used for video human activity classification. We have modified it with smaller, fewer, efficient components for the case of content based retrieval. The LSTM's cell state preserves the learnt information and hidden state gives the required output [24]. We extract the features from the last time-step of the LSTM. This is the final feature representation of the video and is compared with the rest of the videos in the database. The comparator module checks the similarity and ranks the results based on the scores. For scoring the results, we first rank them in the order of their Euclidean distance and then among the top *n*, we check their cosine similarity (which gives the orientation information) and give the final results.

$$Euclidean\ distance = \sqrt{\sum_{i=1}^{nd}(ft_i - fi_i)^2} \quad (1)$$

$$Cosine\ similarity = \frac{\sum_{i=1}^{nd} ft_i . fi_i}{\sqrt{\sum_{i=1}^{nd}(ft_i)^2}\sqrt{\sum_{i=1}^{nd}(fi_i)^2}} \quad (2)$$

Where *nd* is the number of images/videos in the database.

4.1.2 Training and using the models

For the feature extractor, though we have used an ImageNet [27] pre-trained MobileNet V1 model [25], it is always possible to train an application specific CNN as a classifier or an object detector. The thumb rule is to use at least 2000 images per class for decent performance. It is advisable not to train the model completely yielding very high accuracy as it would induce bias towards the data and restrict generalization which is important. Later we experiment and

pick an intermediate layer that can give a compact yet efficient feature representation as the output layer and discard the rest.

For the LSTM network, we train the network as a classifier with three classes KTH Human Action Recognition data [17] which are *Walking, Boxing and Waving*. The input to LSTM cells are the frames features and the output as a category of video. The data necessity for this network is very low i.e. about 1 hour of video per class is enough to train the model. Later after training, we remove the classification head and use the hidden features from the last time step of the LSTM network but this can be done in multiple ways. One can aggregate the outputs of hidden states at all the time steps or one can combine hidden state and cell state outputs. We found through experimentation that the last hidden output of dimension *1x dl* (where *dl* is the number of LSTM hidden units) was sufficient to produce compelling results. We test the results on a combination of manually collected videos and the three other classes available in the KTH-RHA which are *Jogging, Running and Clapping* to verify if the architecture is able to generalize and learn the content of the videos. The classes of KTH-HRA are distinct from each other in terms of human posture and the micro-actions involved. The results of the model are given in the sections to follow.

4.2 Experiments

In this section, we discuss the experiments we performed for this research. Though, it is believed that ConvNets benefit from learning colors resulting in performance, the exact dependence hasn't been studied. We were curious on how much color in images can make a difference in terms of performance of retrieval in comparison to grayscale images. For this experiment, we trained two instances of the same MobileNetV2 [30] with exactly the same parameters except for only the number of channels – 3 for color and 1 for grayscale on the same dataset. After the model was trained on the data, we tested the model on the same data to compare the results.

We also conducted several small experiments on picking the right layer for feature representation that gives the maximum performance for image features, tuning hidden units in the LSTM network for the best video representation. We found out that the layer before the first fully connected layer gives the best performance almost always. For the LSTM features, the hidden output from the last time step with size 1x200 gave compelling results in our case. Both these are with respect to the models we have selected or trained for our application and there is a great chance that this will vary drastically in other cases.

4.3 Faster retrieval techniques

As our main emphasis is on scalability, we focus on factors like processing time (computation time and retrieval time), feature dimension, comparison metric, reliability of the algorithm etc. In this section, we discuss the methods for efficient retrieval.

4.3.1 Feature dimensionality reduction

The comparison time is directly proportional to dimension of the features. We propose to reduce the feature dimension using Principal Component Analysis [13] which has the capability of retaining only the key features and discard the rest. We found that the retrieval accuracy was almost similar after using PCA according to [6] whose results are in section 5. We also suggest to employ Linear Discriminant Analysis, using AutoEncoders which is known for feature compression, Siamese network that can directly give a comparison score between two images. The AutoEncoder and Siamese network may prove to be heavy on computation.

4.3.2 Fast comparison

We identify techniques for faster comparison. One obvious option is to use clustering on the media in database and use only the centroid for comparison. But if the application requires precision, clustering may not be a good option and one has to update the centroid, every time a data point is added.

4.3.3 Model optimization

There are several options to optimize the deep learning models like filter pruning in which we remove the redundant filters and reduce the computation time, replacing fully-connected layers with 1D convolution layers have been popular in the domain, distillation [23] to train a large model with the best accuracy and using it to create sub-models as required by the nature of application and reducing the size of the model.

5. RESULTS

5.1 Results of video retrieval model

3D Convolution Neural Networks [19] are prevalent for video tasks such as activity recognition, video classification etc. They are known for their heavy data and computational requirement owing to their huge size. Our novel architecture is a good alternative to C3D as it has several attractive features such as modularity, flexibility, light-weight etc.

We tested our *MobileNet V1 + LSTM model* on few manually downloaded videos along with videos from 3 classes (Jogging, Running, and Clapping) of KTH-RHA which were excluded for training. We trained and benchmarked all the results in this section on a Core i5 – 6300, 12GB RAM machine without any graphics card to show the light-weight nature of our model. The advantages of our novel architecture are mentioned in the table 5.1.1. Also, as this is not a classification problem, ***we recommend using***

precision and accuracy as benchmarking metrics *since recall, F1-score, mean Average Precision are not relevant to retrieval tasks; meaning, the retrieval is either right or wrong and there is no true negative cases.*

$$\boldsymbol{Precision} = \frac{TP}{TP+FP} \quad (3)$$

$$\boldsymbol{Retrieval\ Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (4)$$

Field	Conv3D	MobileNet V1 + LSTM
Data requirement	Minimum 30 hours of video per class for ~70% accuracy	Achieved 70% retrieval accuracy with 1 hour of video
Training time	In terms of several hours	< 10 minutes for LSTM with a pre-trained model
Model size	305.17MB with ~80M parameters	16.8MB + 5.72MB with 4.9M parameters in total
Flexibility	Retrain the whole network to add a class	Retrain only the LSTM part due to the modular nature
Processing time for 16x112x112x3	2.43 seconds	1.51 seconds
Performance metrics on KTH-RHA (*unseen classes - 300 clips)*	<TBD>*	*Average Precision 72.93%* *Accuracy 71.78%* *F1 Score 61.32%*

Table 5.1.1 Advantages of our novel architecture

* Due to the time constraint and the heavy nature of the C3D model requiring huge computational resources, we were not able to bench mark the results of C3D on KTH-RHA

5.2 Results of experiments

The results of our experiment on the effect of color is given in the table 5.2.1 below. The MobileNetV2 [30] color and grayscale classification models were trained* on CIFAR10 dataset [29] and were tested on Oxford5k [18] images resized to (32x32) after discarding the classification head. It can be seen from the table 5.2.1 that the variation in accuracy between the grayscale model and the color model is very meagre on trained and untrained data. This permits the grayscale model to be used in retrieval tasks *which ensures faster computation and performance almost similar to color images*.

Dataset	Retrieval accuracy – color model (%)	Retrieval accuracy – grayscale model (%)
CIFAR10 test set	81.62	80.03
Oxford5k - unseen 250 images	21.3	19.17

Table 5.2.1 Results of experiments on color

* Models trained for 50 epochs on 50,000 32x32 images with validation accuracy of ~74%.

The results of another experiment, incorporating PCA for dimensionality reduction is shown in the Figure 5.2.2 below. PCA was applied on the LSTM features (video representation) and the accuracy of retrieval for videos are provided. ***The retrieval accuracy with LSTM final state hidden features without PCA is 65.52%.*** PCA reduces dimension of the feature representation while preserving the important components which can help making the comparison efficient as comparing two arrays of size 10 is faster than comparing 2 arrays of size 200.

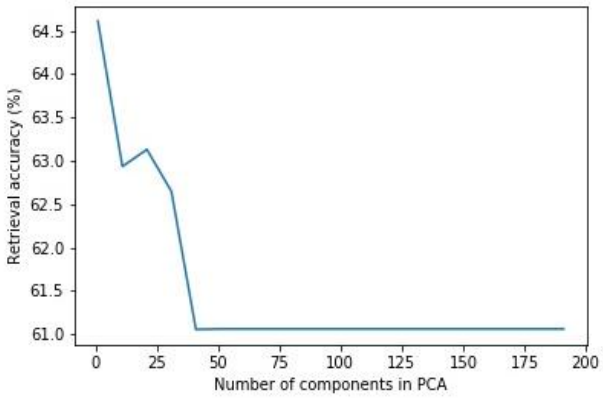

Figure 5.2.2 Effect of PCA on retrieval tasks.

6. FUTURE SCOPE

In this section, we identify the future scope for improving this proposed system in terms of optimization and better implementation. First we focus on better implementation, for better accuracy of retrieval the changes that can be imparted into a system. Due to the modular nature of our proposed architecture, an inclusion of an application specific feature extractor, increasing the length of the LSTM network [24] , experimenting with the optimum number units for the best results, implementing dynamic, bidirectional, multilayered LSTMs can be done. Also, implementing attention in LSTMs have been proven to produce better results in such complex tasks [26] which can be done at a later stage. Due to the modular nature of our proposed architecture, we think it is possible. We also think that exploring fast comparison libraries like FAIR's FAISS [21], Spotify's modification of approximate nearest neighbors ANNOY index [20] will help in improving the processing time when it comes to large databases.

7. CONCLUSION

In this paper, we present a novel modular approach for joint image and video retrieval. This proposed architecture is flexible for wide range of applications. We have also mentioned our experimental results on the effect of color on retrieval tasks that gray scale model fall behind color counterpart by a minuscular level but will be capable of performing better in terms of computation time with almost similar accuracy. We also discuss various learning metrics and approaches that can speed up the comparison process in retrieval tasks. The three parts together have the potential to make a significant impact on scalability of retrieval tasks for real world applications.

8. REFERENCES

[1] S. A. Ladhake and A. A. Solio, "A review of query image in Content Based Image Retrieval", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, Volume 2, Issue 4, April 2013.

[2] G. Csurka, C. Dance et al. "Visual categorization with bags of keypoints". *Work Stat Learn Comput Vision, ECCV*. Vol. 1 (2004)

[3] M. Rezaei, A. Ahmadi and N. Naderi, "Content-based image retrieval using Mix histogram" in *2nd National Conference on new researches in Electrical and Computer Engineering, Tehran-2017*.

[4] A. V. Singh, "Content-Based Image Retrieval using Deep Learning". *Thesis - Rochester Institute of Technology, June-2015.*

[5] J. Wan, D. Wang, S. C. Hoi, P.Wu, J. Zhu,Y. Zhang and J Li, "Deep learning for content-based image retrieval: A comprehensive study" *MM '14: Proceedings of the 22nd ACM International Conference on Multimedia*, November 3-7, 2014, Orlando. 157-166.

[6] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. "Deep image retrieval: Learning global representations for image search". *In European conference on computer vision*, pages 241–257. Springer, 2016.

[7] A. Potapov, I. Zhdanov, O. Scherbakov, N. Skorobogatko, H. Latapie and E. Fenoglio, "Semantic Image Retrieval by Uniting Deep Neural Networks and Cognitive Architectures" *Computer Vision and Pattern Matching, arXiv*, July 2018.

[8] R. Arandjelović and A. Zisserman. "Three things everyone should know to improve object retrieval." *In 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2911–2918. IEEE, 2012.

[9] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. "End-to-end learning of deep visual representations for image retrieval." *International Journal of Computer Vision*, 124(2):237–254, Sept. 2017.

[10] A. Babenko and V. Lempitsky. "Aggregating local deep features for image retrieval". In *Proceedings of the IEEE international conference on computer vision*, pages 1269–1277, 2015.

[11] H. Jun, B. Ko, Y. Kim, I. Kim, and J. Kim. "Combination of multiple global descriptors for image retrieval." *arXiv*, 2019.

[12] F. Radenović, G. Tolias and O. Chum, "Fine-tuning CNN Image Retrieval with No Human Annotation": in *arXiv*, Submitted on 3 Nov 2017.

[13] H. Jégou and O. Chum. "Negative evidences and cooccurences in image retrieval: The benefit of pca and whitening." In *European conference on computer vision*, pages 774–787. Springer, 2012.

[14] A. Suprem, "Approximate Query Matching for Image Retrieval": in *arXiv*, 15, March 2018.

[15] P. Sadeghi-Tehran, P. Angelov, N. Virlet and M. J. Hawkesford, "Scalable Database Indexing and Fast Image Retrieval Based on Deep Learning and Hierarchically Nested Structure Applied to Remote Sensing and Plant Biology" in *Journal of Imaging*, Published: 1 March 2019.

[16] S. Ray and R. H. Turi. "Determination of number of clusters in k-means clustering and application in colour image segmentation". In Proceedings of *the 4th international conference on advances in pattern recognition and digital techniques*, pages 137–143. Calcutta, India, 1999.

[17] C. Schüldt, I. Laptev, B. Caputo, "Recognizing Human Actions: A Local SVM Approach", Proceedings of the *17th IEEE International Conference on Pattern Recognition* (ICPR 2004)

[18] J. Philbin, O. Chum, M. Isard, J. Sivic and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching", Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition (2007)*

[19] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks", *ICCV* 2015.

[20] W. Li, Y. Zhang, Y. Sun, W. Wang,W. Zhang,X. Lin, "Approximate Nearest Neighbor Search on HighDimensional Data - Experiments, Analyses, and Improvement", *arXiv*, Oct, 2016

[21] J. Johnson, M. Douze, H. Jégou, "Billion-scale similarity search with GPUs", *arXiv* 2017

[22] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, T. Darrell, "Long-term Recurrent Convolutional Networks forVisual Recognition and Description", *arXiv*, May 2016.

[23] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, "Distilling the Knowledge in a Neural Network", *arXiv*, March 2015.

[24] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term memory." *Neural computation*, 9(8):1735–1780, 1997.

[25] A. G. Howard, M. Zhu, B. Chen, D, Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", *cs.CV, arXiv*, 2017.

[26] A Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, "Attention Is All You Need", *NIPS*, 2017.

[27] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database", *CVPR* 2017

[28] J. Yosinski, J. Clune, Y. Bengio and H. Lipson , "How transferable are features in deep neural networks?", *NIPS* 2014.

[29] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images", 2009

[30] M. Sandler, A. Howards, M. Zhu, A. Zhmoginov, L Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", *arXiv cs.CV*, March 2019

[31] http://www.nada.kth.se/cvap/actions/

[32] https://www.robots.ox.ac.uk/~vgg/data/parisbuildings/

[33] https://www.robots.ox.ac.uk/~vgg/data/oxbuildings/

[34] https://www.cs.toronto.edu/~kriz/cifar.html