Image Retrieval: Using Image Captioning and Image Matching

Akash Agarwal
Oregon State University
Corvallis, OR
agarwaak@oregonstate.edu

Anand P. Koshy
Oregon State University
Corvallis, OR
koshya@oregonstate.edu

Sonam Gyanchandani
Oregon State University
Corvallis, OR
gyanchas@oregonstate.edu

Abstract

According to Wikipedia, an Image Retrieval System is a computer system for browsing, searching and retrieving images from a large database of digital images. Search Engines such as Google, make use of image retrieval techniques for performing an image search. It should be noted that the search engines have a record of the image labels, which they can match with the query to execute the task. In the absence of the Labeled database, existing works either perform image retrieval by implementing image matching using an example image as a search query or by using Image Captioning to label all the un-labeled dataset images to match with the query. We introduce a novel approach to aid in the field of Computer Vision and Information Retrieval by combining the two methodologies.

1. Introduction

Image search problem and has been under active research in the field of Computer Vision for the past couple of years. The problem becomes more challenging than other Information Retrieval methods when there is no prior information regarding the content of the image database i.e. when the dataset is un-labelled. To overcome this problem, companies like Google and Apple, offer their own solutions. However, these methods have their own limitations. Google Images offer an Image Search feature, which returns very accurate results, but it requires user to insert an example image as a query. Apple on the other hand, in its devices provides an option to search for images using short phrases that define the category/ type of the image. The problem with this method being the lack of description for the image to be searched, unless quite distinct, the user might not be able to come-up with a keyword that identifies the image from the crowd. Also, the search can be performed only for the images present offline on the device.

To overcome all the problems discussed above we have implemented a novel method to perform image search and retrieval. The method makes use of the concepts of Image Captioning and Image Matching to retrieve images from an un-labelled dataset, enabling user to input a text-based query (label or description of content in the desired image) to get the matched results. We introduce a Bi-modal architecture, comprising of a LSTM and a CNN model to perform the above task.

The remainder of the report is arranged as follows. Section-2 gives the literature review and talks about existing research work. Section-3.1 describes the model architecture. Section-3.2 talks in detail about the datasets and modules used in the network and provides the comparison of different methods of implementation used. Lastly, Section-4 talks about the Results, the scope of improvement and Future Work.

2. Related Work

Recent advances in the field of Deep Learning and Computer Vision have led to dramatic success in solving various problems like image captioning [1] and [2], machine translation, word2vec [3], etc. These works have led to a lot of research in the field of content-based image retrieval [4]. One of the approaches makes use of a LSTM model for image matching. [5] The author provides a variety of options that can be used as a user query. In this approach, a trained LSTM model is made to run on all of the database images, such that it creates captions for every image. The captions and the user query are vectorized and the cosine distance between them serves as the measure of similarity between them. Although, if a match is found, the model is sure to return the results, but the notion of apt query selection is rather vague. Also, even if two images hold similar content, their captions generated by the model might be entirely different, in which case, only one of them would be returned as a matched result. Our model architecture discussed in the later section handles this problem. There are ample techniques that exist for Image matching as well. One of those make use of Keypoint Matching. A keypoint detector model, e.g. SIFT [6] is used to detect key-points in every dataset image, then a CNN model is used to extract feature descriptors of these points. It is a general practice to represent an image in its feature space. The motivation is to achieve an implicit alignment so as to eliminate the impact of transformations or other

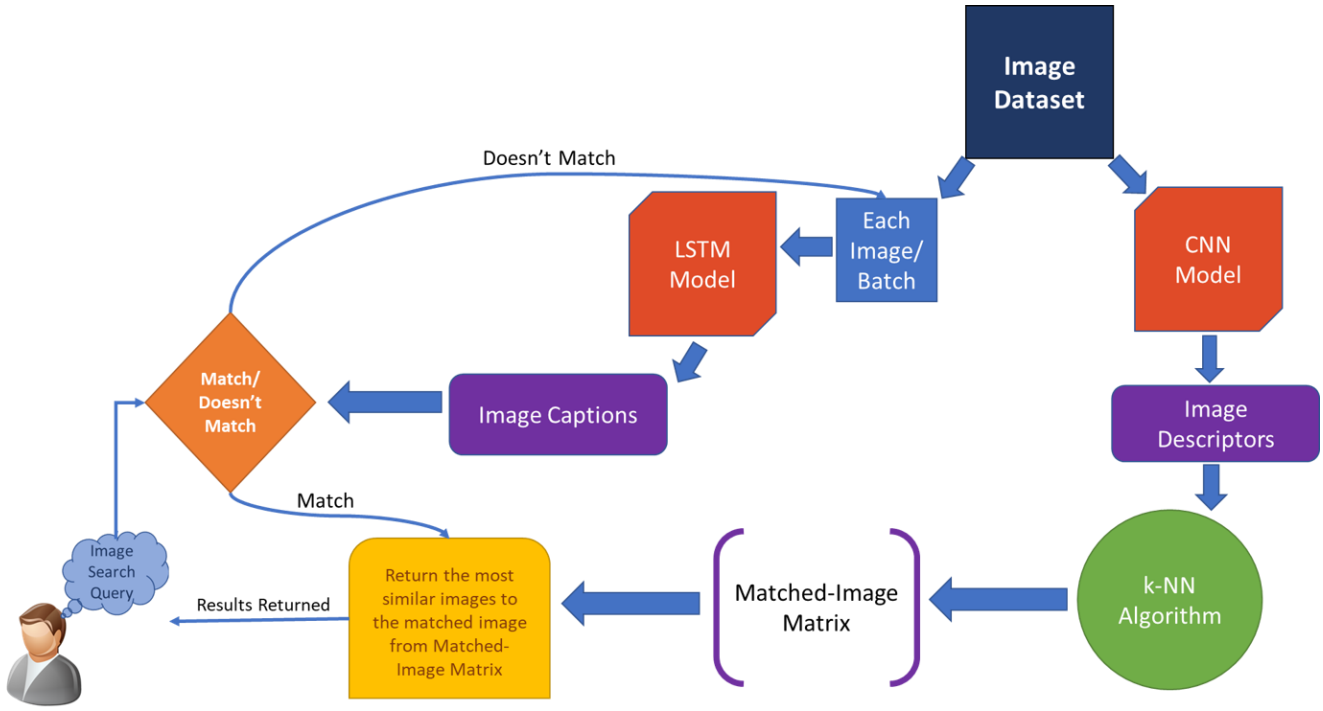

Figure 1: Model Architecture

changes and to make it invariant to difference in other extrinsic features like color, and/or location (if necessary) while keeping the intrinsic visual content distinguishable. The CNN model that produces the descriptors is trained by backpropagating on Triplet loss. The distance between the descriptors generated for every image represent the differences between them. A cost matrix is formed that is passed through some Point Matching Algorithm, e.g. Hungarian Algorithm to predict the similarity between the images. We have tried both, the above-mentioned method and another approach that uses k-NN Algorithm to predict the similarity. It was observed that the latter produces better results. Currently, there has been a trend of using the weights pre-trained on a different problem, having a larger dataset, to solve some other related problems. This concept is known as Transfer Learning[1]. The CNN model that we implemented, makes use of this concept, where the weights are taken from a model pretrained on ImageNet dataset.

It should be noted that simple image-image matching requires a query image, and Image Captioning method may not prove to be a more accurate approach. Therefore, we will discuss our novel approach of combining the two methods to overcome their shortcomings and retrieve more accurate results.

3. Methodology

In this section, we discuss the whole model architecture and explain the working of each sub-module.

3.1. Model Architecture

Figure-1 represents the Architecture of the model. The architecture consists of two sub-models: a LSTM model and a CNN model. The LSTM and the CNN, are used for the purpose of image captioning and image matching respectively. Every image in the dataset is passed through a CNN model that generates an n-Dimensional vector for each image. These vectors serve as the image descriptors. We then perform k-NN (k-Nearest Neighbor) Algorithm on a set of all the image descriptors, such that for a particular image descriptor, we find the k nearest image descriptors in the feature plan. This neighborhood represents a set of most similar images, for a particular image. The Matched-Image Matrix is a (N x k) dimensional 2-d array, where N rows represent the total number of images and k columns represent the k most similar images. Thus, for an i^{th} row, each column j contains the index of the matched image and the image represented by the j^{th} column is more similar to that of the $j+1^{th}$ column. The Matched-image matrix is then

[1] Read: "A Gentle Introduction to Transfer Learning for Deep Learning", for more information.

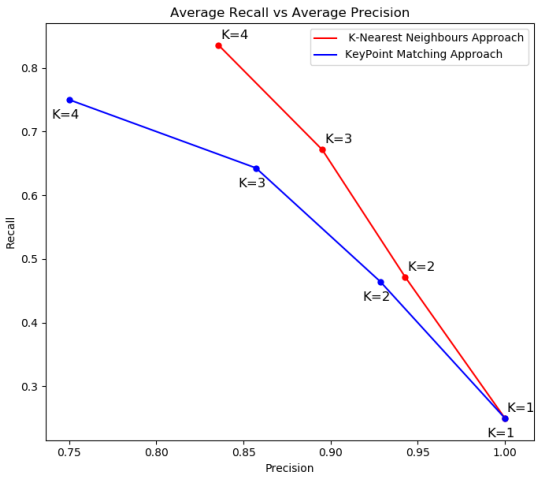

Figure 2: Recall vs Precision Curve

stored in a file, which can be directly used in real-time. Given a text-based user query, from the dataset, we extract

each image or a batch of images and pass it to the LSTM model (pre-trained), such that it generates their captions; until we find a match between the user query and the caption of that image. For matching the user query and the image captions, we vectorize the two texts and find out the cosine similarity between them. As soon as a match is found, we return the matched image and for that image, from the file containing Matched-Image matrix, we return k most similar images. Thus, performing image retrieval from an un-labelled dataset using a text-based query.

3.2. Modules

In this section we discuss the datasets, CNN model and the LSTM models used for the implementation and compare the different approaches that affected the final decision of the architecture.

3.2.1 Dataset

Since the main motive of the project was to extract images from an unlabeled dataset, we used MSCOCO validation dataset-2017 which contains 5000 images, prominently used for image captioning. The task of labeling semantic objects requires that each pixel of the image be labeled as belonging to a category. So, a dataset that combines the properties of both object detection and semantic scene labeling is necessary. We use this type of dataset for image captioning model to get good captions from everyday scene objects. The CNN module used in the project uses pre-trained weights from the model that was trained on ImageNet Dataset. Also, we prepared a very small labeled test dataset to compare the performance of the two approaches taken for Image matching.

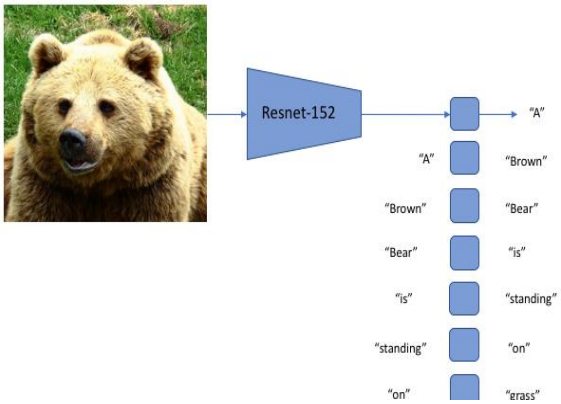

Figure 3: Captions Generation

3.2.2 CNN Model

To extract the feature descriptors, we implemented two methods of Image Matching. In the first one, we extracted 30 most prominent key-points for each image in the handmade dataset. For each keypoint we extracted image patches of size 32x32 and passed them to the CNN model for getting keypoint-descriptors. These key-point descriptors are required for Point-Point Matching, in which we calculated the L2 Norm of the distance between each pair of descriptors to represent the cost of matching. For each pair of images, we formulated a 30x30 Cost Matrix, we then used a Point-Matching algorithm, e.g. Hungarian Algorithm, to get the total cost of image matching. It is rather intuitive that the negative exponent of this total cost works as a measure of similarity between the images. Thus, we created a Similarity Matrix that consist of the similarity values between every pair of two images in the dataset. In the second approach, we removed the notion of key-points all together and passed the whole reshaped images through the CNN model, that generates one feature descriptor for the entire image. We then applied k-NN Algorithm on the set containing all the image descriptors and found out the nearest neighbor vector in the feature plane. These nearest neighbors represent the most similar images.

For each of the two approaches we made use of VGG15 and RESNET-50 architecture. RESNET-50 gave better accuracy on individual approaches, so we went with Resnet. Figure-2, shows the ROC curve comparing both the approaches.

3.2.3 LSTM Model

To get the captions out of the image, image captioning was used. Image captioning refers to the process of generating textual description of the image. The captioning for our system is done using the multi-modal learning, combining image-based model and a language-based model. The image processing was done using a ResNet-152 and text processing is done using the state-of-the-art LSTM

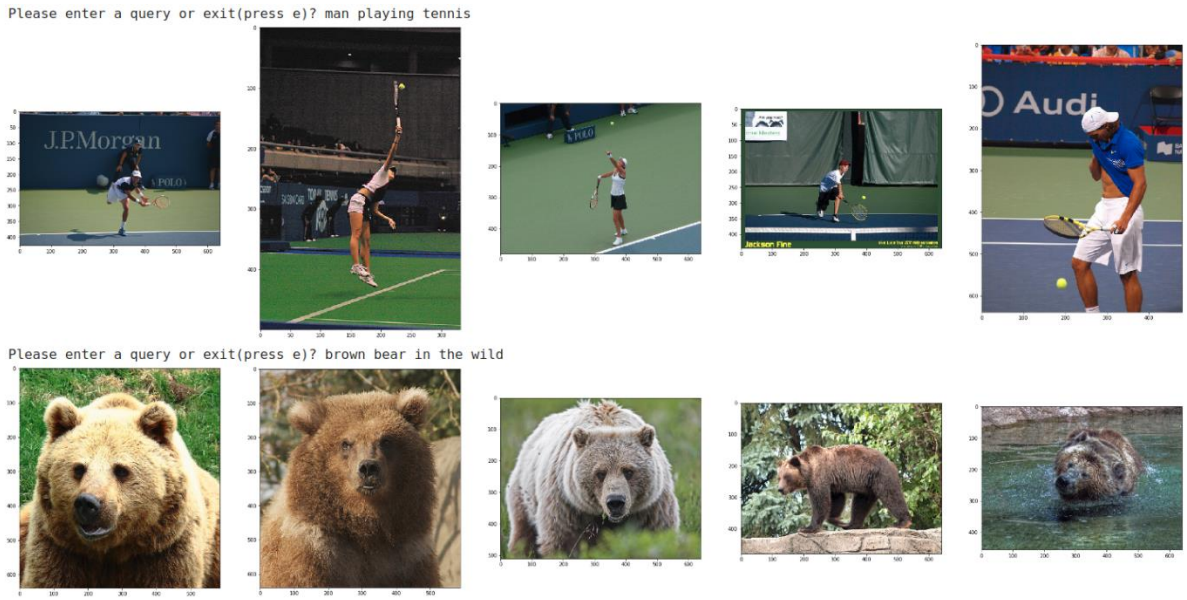

Figure 4: Results- Images Retrieved from the dataset for a text-query

model. The architecture used for image captioning is encoder-decoder architecture. The encoder being the resnet-152 gets the features and nuances out of the image and passes the features to the decoder which is LSTM. The feature vector from the resnet-152 is linearly transformed to have same output dimensions as the input dimensions of the RNN/LSTM model[2]. To use resnet-152 in as an encoder, we removed the last fully connected layer of the pretrained model and created a fully connected layer of the size which is equal to the input features of the LSTM model. For our implementation, we got 256 feature tensor and passed it to the LSTM model. We included a batch norm layer after this linear layer to stabilize our network. For the decoder, we used pytorch pretrained LSTM model which decodes the image features and generate captions.

3.2.4 Cosine Similarity

After generating the captions, in order to compare the similarity of the two texts viz user text query and the generated image caption, we used cosine similarity to measure the similarity between the two non-zero vectors of inner-product space by calculating cosine angle between the them. We converted the captions and user text query in vector form by calculating the term frequency (tf). The cosine angle between two vectors was then calculated.

4. Conclusion and Future Work

In this work, we demonstrated an Image Retrieval model that inputs a text-based user query to extract query related images from an un-labeled dataset. Our model uses LSTM and a CNN model as sub-modules; the former works in the real time, generating image captions for a single image/ batch of images at a time, for the sake of finding a match with the user-query. The latter, is used to generate a matched-image matrix for the entire un-labelled dataset, to keep track of the similar images. Figure 3, demonstrates the results generated by this model. The integrated final output was very accurate and returned acceptable results for a particular search query. As evident from the figure, the user-query could include a sentence describing the required image.

The image retrieval time was adequately fast with an average retrieval time of ~0.76 secs. Although the time that the CNN model takes to generate the matched-image matrix is roughly 2 hours for 5000 images in the MS-COCO dataset, the match-list is needed to be created only once, beforehand. This means that the CNN module is not required to work in real time and is merely a part of the model build process.

There are a number of improvements and future work that can be made to this work. Although, CNN module is a part of build system, the execution time is slow for a dynamic dataset. Currently, if a new image is being added to the dataset, the CNN model will have to be executed again to generate a new similarity matrix. As stated above, the building process takes a considerable amount of time. An approach is needed such that the new images affect only those rows that contain similar images, without having to re-calculate the similarities for other dis-similar images. Also, a lot of implementation level improvements can be

[2] https://www.analyticsvidhya.com/blog/2018/04/solving-an-image-captioning-task-using-deep-learning/

made. For example, Multi-threading can be a good approach for both, simultaneously generating captions of multiple images and for forming multiple matched-matrix rows. The CNN module that is used for image matching can also work as the encoder part of the LSTM module. Each image descriptor obtained using CNN during the build process can be stored in a file. Instead of accessing the images and passing them to the encoder at runtime, we can extract the corresponding image descriptors from the file and feed it to the decoder. This will help in reducing the space and time complexity tremendously.

5. References

[1] A. Karpathy and F.-F. Li, "Deep visual semantic alignments for generating image descriptions.," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[2] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *CoRR abs/1502.03044*, 2015.

[3] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *arXiv:1310.4546*, 2013.

[4] W. Zhou, H. Li and Q. Tian, "Recent advance in content-based image retrieval: A literature survey," in *CoRR abs/1706.06064*, 2017.

[5] T. Piplani, "DeepSeek: Content Based Image Search & Retrieval," in *arXiv:1801.03406 [cs.IR]*, 2018.

[6] D. G. Lowe., "Distinctive image features from scale-invariant keypoints.," in *Int. J. Comput. Vision*, 2004.