Sketch Based Image Retrieval
Project Course On Media Retrieval

Kazi Injamamul Haque - 204873

University of Trento, TN, Italy
kaziinjamamul.haque@studenti.unitn.it

Abstract. From the early 1990's, researchers started exploring sketch-based image retrieval yet it gained substantial popularity in 2014 after the advent of deep learning techniques. In the process of carrying out this project, we explore deep learning, more specifically convolutional neural network for training a classification model in addition to exploring Bing's image search RESTful API using python for the retrieval process. We trained a classifier model that classifies a user input, a free hand sketch in this case, and predicts the object in the input image. The prediction is transformed in a text and passed on to the retrieval module that retrieves real images of the predicted class using python and BING image search API and shows to the user in a gallery format. Our system was trained with 75481 sketch images from 150 different classes. Although in recent years SBIR relies on fine-grained classification, in this project we did not incorporate this trend in our training. The objective of this project was to briefly explore convolutional neural network and different image retrieval techniques.

Keywords: SBIR · Deep learning · Image retrieval · Convolutional neural network · Image search API

1 Introduction

Sketch-based image retrieval(SBIR) is an active research topic under the content based image retrieval for quite a long time. SBIR allows users to query images from the database using free hand coarse sketches. In a complex SBIR system, it allows user to search for images where text based search is not convenient as sketches can express human mind better than texts. The visual similarity is then used to search and retrieve similar real images from the database. Unfortunately, this works very well well the image database is small scale or there are few number of classes to retrieve from. With the booming usage of internet we now have access to large scale datasets and with the help of ground-breaking performance in recent years of deep learning in image analysis, it is now possible to train a classifier model that can understand free hand sketches with holistic shapes of objects and classifies in its category with acceptable error rate. In most recent years of SBIR research, a novel term "fine-grained" classification is introduced where the trained classifier distinguishes among intra-category objects. For instance, in a fine-grained sketch classifier will classify an apple and an orange as

the same category of "round fruit". In this project, we did not train our model as a fine-grained classifier rather we explored how robust can a convolutional neural network be in distinguishing exact sketches. In the next section we briefly discuss about the related work on SBIR starting from the first era to the most recent accomplishments. In section 3 we discuss our proposed framework and the tools that we used in detail and explain the system architecture. Furthermore the following section after that presents the results of the retrieval system, and in section 5 we discuss about the limitations and the future work before summarizing the whole report in the conclusion section.

2 Related Work

Research in sketch based image retrieval can date back to the early 1990's and until now, according to [1], SBIR can be divided into four different eras depending on the methodologies on how the problem was tackled by the researchers. In the first era in 1992, SBIR can be summarized into two phases, Population phase where the researches extracted the edges from the sketch images and in the retrieval phase the extracted edge was compared for similarities and displaying the most similar contents in a gallery. In the second era, between 1994 and 2005, in addition to extracting edges, different features for sketches and real images were introduced and these features[2][3][4][5] were used to carry out similarity matching for retrieval as well. Between 2009 and 2014, in the third era of SBIR, the specific research area gained popularity because of the explosion of the internet usage and availability of large-scale dataset. [6][7][8][9][10] are some notable SBIR researches in the third era those handle the random distortion in the sketch image data and robustly handle different stroke variants in the data.

The fourth era of SBIR which started around 2014 and still prevailing is the most important era for us to explore because of the explosion of deep learning techniques and what deep learning can achieve when it is put to appropriate use. Furthermore, the increase in performance of hardware in recent years, especially GPUs has enabled researchers to exploit the computing power for deep learning as these GPUs are also now in affordable price range. In addition, the cuda cores in the modern day GPUs can be exploited in order to accelerate the learning phase drastically.

In most recent SBIR researches, the novel idea of "fine-grained" SBIR was introduced. Li et al.[12] first raised the concept of this novel trend in 2014 to differentiate intra-category variations. Li et al. adopted the deformable part-based (DPM) model[13] in grid division scheme to be served as object detector and a representation. In addition to that, the authors also employed histogram of gradient features (HOG) to cope with the random distortions. Fine-grained concept was also encouraged by two more notable works [14][15] where both of the works uses deep learning techniques to train a triplet network model with three branches for fine-grained classification. The triplet network is shared among three tasks - triplet ranking, attribute prediction and attribute ranking in order to perform the final triplet ranking. The network architectures in Sketch-

a-Net[16] and the deep neural network in [17] are good fine-grained SBIR model architectures derived from AlexNet architecture and GoogLeNet architecture respectively. Although the latter is much deeper network than the former. In [18], Bui et al. propose a multi-stage regression training strategy for partial sharing networks. The authors integrate the two most widely used regression functions in deep convnet, the contrastive loss and triplet loss, in their training procedure.

3 Proposed Framework

Our proposed framework for sketch based image retrieval is divided into two modules where the first module is "Model Training" and the second module is "Retrieval". Model training exploits deep learning technique, more specifically convolutional neural network to train a classification model using publicly available sketch image dataset. The classifier predicts the user input (i.e. a free hand sketch image) and classifies as one of the classes among all the classes the model is trained with. The prediction is transformed into text from a softmax vector and passed on to the retrieval module in order to retrieve real images to show to the user. For training procedure we used Keras[22] deep learning framework with Tensorflow[21] in the backend.

The Retrieval module makes use of Bing's image search RESTful API to query the predicted result and retrieve real images of the predicted class by searching the predicted term. This module loads the saved weights of the trained model and predicts by comparing the user input that is a free hand sketch on a python canvas container by the user and shows retrieval results as an image gallery grid.

Dataset The dataset used for training the model was acquired from The Sketchy Database[19]. The dataset consists of 75471 sketches belonging to 125 categories. All the sketches in the dataset are human drawn inspired by 12000 real images. The sketches were combined with TU Berlin sketch dataset[20] which consists of sketches of 250 different categories each containing 80 sketch images. We conbined the two datasets on the common categories that both datasets share and split the resulting dataset into training and validation datasets. As a result for final training we end up with 75481 training sketches and 11189 validation images all belonging to 125 different categories.

3.1 Model Training

Our model training was carried out following keras deep learning framework with tensorflow on the backend. In convolutional neural network[11], the convolutional layers together with the max-pooling layers can cope up with the random distortion and translation. Additionally, the max-pooling layers can simplify the process by offering abstraction for the objects in the pooling. The input layer

was set of sketch images all resized and rendered in 256x256 uniform dimensions with 3 color channels RGB. The input layer is then processed with three consecutive pairs of convolution and max pooling layer where the convolution filters are of size 3x3 and each max pooling filter is of size 2x2. Between each convolution and max pooling, we incorporate Rectified Linear Unit(ReLU) as activation fuction. After the third max-pool layer we introduced a layer in the network to flatten the output of the last max-pool layer then pass it onto the first fully connected dense layer. Further, another (second) fully connected layer is then in work with activation function as ReLU before getting the output layer with activation function as Softmax. During the network training we used Adam optimizer, categorical loss and took Accuracy as our cross entropy metric to train the final network. Fig.1 illustrates the abstraction of our final trained network.

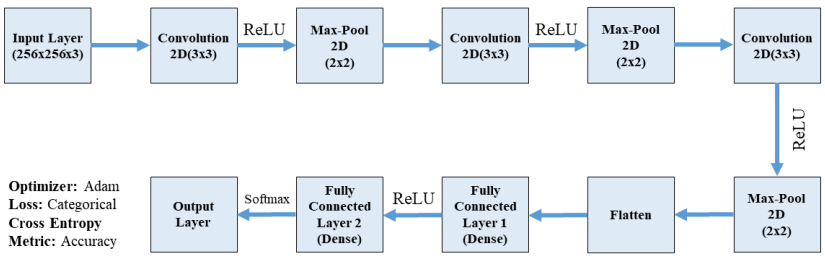

Fig. 1. Illustration of the Trained Network

Data Augmentation Due to limited available sketch data, the training data was augmented using the framework's image generator. In addition to having the original sketch images, each sketch image produced two additional sketch images based on different augmentation parameters such as scale factor, shear range, zoom range and horizontal flip. The data augmentation showed better results concerning training accuracy and loss in the final model training as compared to the initial instances of model training.

3.2 Retrieval

The retrieval module we use the pretrained deep neural network classifier that we trained and saved in the previous module to compare and classify a newly seen free hand sketch image drawn on a python canvas or the image is provided by the user. The model predicts the class of the the input image using the pretrained classifier and yields the predicted result as text which then passed onto retrieve real image corresponding to the predicted class. In order to retrieve the corresponding real images, we incorporate Bing's image search python API[23] to search the predicted term and collect the urls of the real images. Due to unavailability of local structured corresponding real image dataset, it was impractical for us to retrieve the results from a local database as making such local image database is extremely time consuming and out of the scope of this project time window. As a result, we were motivated to use existing online image search API such as Bing's. The code that uses the API takes the predicted key term of the input image and retrieves the relevant urls containing the corresponding images. In order to keep the retrieval module and its visualization simple, we only consider the first nine retrieved urls. Subsequently, we use the retrieved urls to recover the image content of the corresponding links. The retrieved images are then show to the user as an image gallery to the user in a grid view. Fig.2 illustrates the abstraction of the retrieval module of our sketch-based image retrieval system.

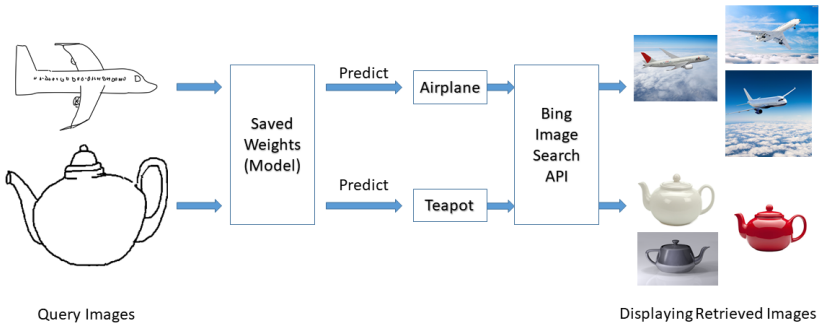

Fig. 2. Illustration of the Retrieval Module

4 Results

Although the validation accuracy during the training phase was relatively low compared to the state of the art SBIR models, the trained model performs well in terms of predicting unseen sketch images in retrieving real images from prediction. After experimenting with the retrieval module with the trained classifier, we found out that predicting exact class from sketch images does not perform very well yet it is very good to generalize unseen data following the idea of fine-grained classification. For example, in some cases, our trained model classifies an unseen sketch of a cat as a tiger and an unseen sketch sketch of a blimp as helicopter. These misclassifications lead to low accuracy in validation in our model and since we did not introduce the underlying idea of fine-grained classification, this will lead to a bad performance in the prediction phase. Whereas in fine-grained classification, these misclassifications would be considered as correct prediction where for example, tiger, lions, leopards, cats would be all under one class "cat". This misclassification phenomenon can be called inter-class ambiguity. Fig.3 shows the training accuracy and training loss during the final model training phase. It can be concluded that the training accuracy increases over time with each increasing epochs whereas the validation accuracy does not perform likewise due to the above-mentioned phenomenon in the validation phase in each epoch. On the other hand, the training loss decreases as the number of training epoch increases that corresponds to increasing training accuracy whereas the validation loss is relatively high due the same effect in the validation phase of the model training.

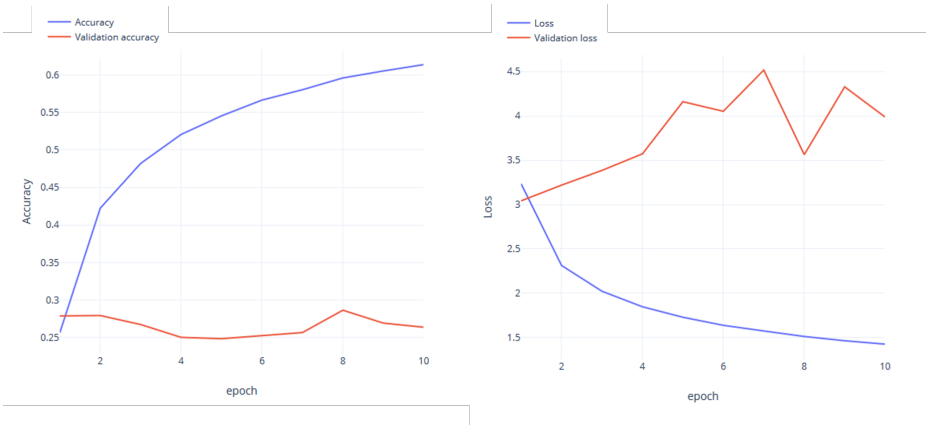

Fig. 3. Model training learning rate. The left figure shows the accuracy(training and validation) rate in training time per epoch and the right figure shows the loss rate both in training and in validation phase per epoch

4.1 Validation

In order to get a global view of how the trained model performs on predicting each of the classes it was trained for, we test the classifier with the validation dataset for each class and record the accuracy metric. We found out that there are classes that have zero accuracy and classes with as high accuracy as 70 percent. The reason behind low accuracy is, in the validation dataset the sketches are different from the train sketch images which results into incorrect prediction hence low accuracy. Test images have to correspond to the style of sketches that were used in the train dataset. On the other hand, the sketches of the classes that are different from the other classes in the trained model and do not share visual similarity, results in high accuracy. For example, jack-olantern, pineapple castle, etc. do to share visual similarity with other classes in the trained model, hence yielding higher accuracy. Table.1 shows the top ten high accuracy classes whereas Table.2 shows some of the low accuracy classes.

Class Name	Accuracy
jack-o-lantern	73.07
pineapple	70.00
tank	64.75
tiger	62.50
castle	61.25
cabin	54.08
skyscraper	53.75
jellyfish	48.35
hedgehog	41.25
windmill	38.42

Table 1. Validation results: Classes with highest accuracy

Class Name	Accuracy
rocket	05.34
scorpion	05.00
beetle	04.89
motorcycle	03.75
saw	03.36
lizard	02.53
trumpet	02.50
swan	01.25
zebra	01.25
songbird	00.98

Table 2. Validation results: Some classes with low non-zero accuracy

5 Discussion And Limitations

From the model training learning rate results in Fig.3, we can comment that since the training accuracy is showing an expected increasing rate in increasing number of epochs, with more training time and/or more powerful hardware, it is possible to achieve better accuracy in training hence a better model for the prediction in generalizing unseen data. Likewise, the training loss rate over time also supports this above mentioned claim. On the contrary, the validation accuracy and calidation loss shown in the figure are strong motivation to introduce fine-grained classification in future training because free hand sketches are hard to classify as exact classes they belong to rather it is relatively much better in predicting a super-class. Furthermore, the neural network model can be trained in more robust way by going in depth of the network architecture and by tweaking the model in with different hyper-parameters. Since the project focuses on

media retrieval rather than deep learning techniques, it is out of the project scope to explore the deep neural network more in depth. On the other hand, we focused our project outcome to propose a complete framework pipeline and implement the proposed framework for sketch based image retrieval in which the performance of the system can be dramatically improved by changing the neural network or even incorporating multiple pre-trained network for SBIR.

The major drawback and limitation faced during the project was availability of powerful hardware for the training module. The training was carried out on a HP ZBook G4 notebook with CPU 8-core Intel Core i7 7700HQ 2.80 GHz, 16GB RAM and with NVIDIA Quadro M1200 with 4GB of video memory as GPU. The final training took around 13 hours to finish. Because of slow training, exploring with different neural network was not practical in the given time window. On the other hand, we strongly believe that with more powerful GPU, the training module can be dramatically improved in order to achieve higher prediction accuracy.

6 Conclusion

In this project, we explored the basics of deep learning, more specifically convolutional deep neural network in the field of sketch based image retrieval. In order to do that, we got familiarized with one of the most popular deeplearning framework being used by researchers and developers, keras and tensorflow. Furthermore, we incorporated available image search RestFUL API to retrieve the real images based on the sketch image prediction. We found out that exact sketch image classification is difficult yet it can achieve higher accuracy in predicting sketches if fine-grained classification is introduced. Due to hardware limitation and the limitation of not being able to use large enough dataset for training the model, the accuracy did not reach the expected level yet we strongly believe that exploring the neural network more in depth and training the model with more time will produce better performance in predicting free hand sketches.

Source Code Repository The processed dataset, source code for Model Training and Retrieval can be found in the followint link.
`https://github.com/galib360/Media_Retrieval_SBIR/`

References

1. Li, Y. , Li, W. A survey of sketch-based image retrieval. Machine Vision and Applications **29**(1083), (2018) https://doi.org/10.1007/s00138-018-0953-8
2. Chalechale, A., Naghdy, G., Mertins, A.: Sketch-based image matching using angular partitioning. IEEE Trans. Syst. Man Cybern. 35(1), 2841 (2005)
3. Chans, Y., Lei, Z., Lopresti, D.P., Kung, S.Y.: A feature-based approach for image retrieval by sketch. In: SPIE International Symposium on Voice, Video and Data Communications, pp. 220231 (1997)

4. Del Bimbo, A., Pala, P.: Visual image retrieval by elastic matching of user sketches. IEEE Trans. Pattern Anal. Mach. Intell. 19(2), 121132 (1997)
5. Rajendran, R.K., Chang, S.F.: Image retrieval with sketches and compositions. In: IEEE International Conference on Multimedia and Expo, pp. 717720 (2000)
6. Eitz, M., Hildebrand, K., Boubekeur, T., Alexa, M.: A descriptor for large scale image retrieval based on sketched feature lines. In: Eurographics Symposium on Sketch-Based Interfaces and Modeling, pp. 2936 (2009)
7. Eitz, M., Hildebrand, K., Boubekeur, T., Alexa, M.: Sketch-based image retrieval: Benchmark and bag-of-features descriptors. IEEE Trans. Vis. Comput. Graph. 17(11), 16241636 (2011)
8. Hu, R., Barnard, M., Collomosse, J.: Gradient field descriptor for sketch based retrieval and localization. In: IEEE International Conference on Image Processing, pp. 10251028 (2010)
9. Hu, R., Collomosse, J.: A performance evaluation of gradient field hog descriptor for sketch based image retrieval. Comput. Vis. Image Underst. 117, 790806 (2013)CrossRefGoogle Scholar
10. Hu, R., Wang, T., Collomosse, J.: A bag-of-regions approach to sketch based image retrieval. In: IEEE International Conference on Image Processing, pp. 36613664 (2011)
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 10971105 (2012)
12. Li, Y., Hospedales, T.M., Song, Y.Z., Gong, S.: Fine-grained sketch-based image retrieval by matching deformable part models. In: In British Machine Vision Conference (BMVC) (2014)
13. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Trans. Pattern Anal. Mach. Intell. 32(9), 16271645 (2010)
14. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: Learning to retrieve badly drawn bunnies. In: SIGGRAPH (2016)
15. Yu, Q., Liu, F., Song, Y.Z., Xiang, T., Hospedales, T.M., Loy, C.C.: Sketch me that shoe. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
16. Yu, Q., Yang, Y., Song, Y., Xiang, T., Hospedales, T.: Sketch-a-net that beats humans. In: British Machine Vision Conference, pp. 7.17.12 (2015)
17. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 19 (2015)
18. T. Bui, L. Ribeiro, M. Ponti, J. Collomosse, "Sketching out the details: Sketch-based image retrieval using convolutional neural networks with multi-stage regression", Computers & Graphics, vol. 71, pp. 77-87, 2018.
19. The Sketchy Database, `http://sketchy.eye.gatech.edu/`. Last accessed 19 May 2019
20. How Do Humans Sketch Objects?, `http://cybertron.cg.tu-berlin.de/eitz/projects/classifysketch/`. Last accessed 19 May 2019
21. Tensorflow Homepage, `https://www.tensorflow.org/`. Last accessed 19 May 2019
22. Keras Homepage, `https://keras.io/`. Last accessed 19 May 2019
23. Quickstart: Search for images using the Bing Image Search REST API and Python, `https://docs.microsoft.com/en-us/azure/cognitive-services/bing-image-search/quickstarts/python/`. Last accessed 26 May 2019