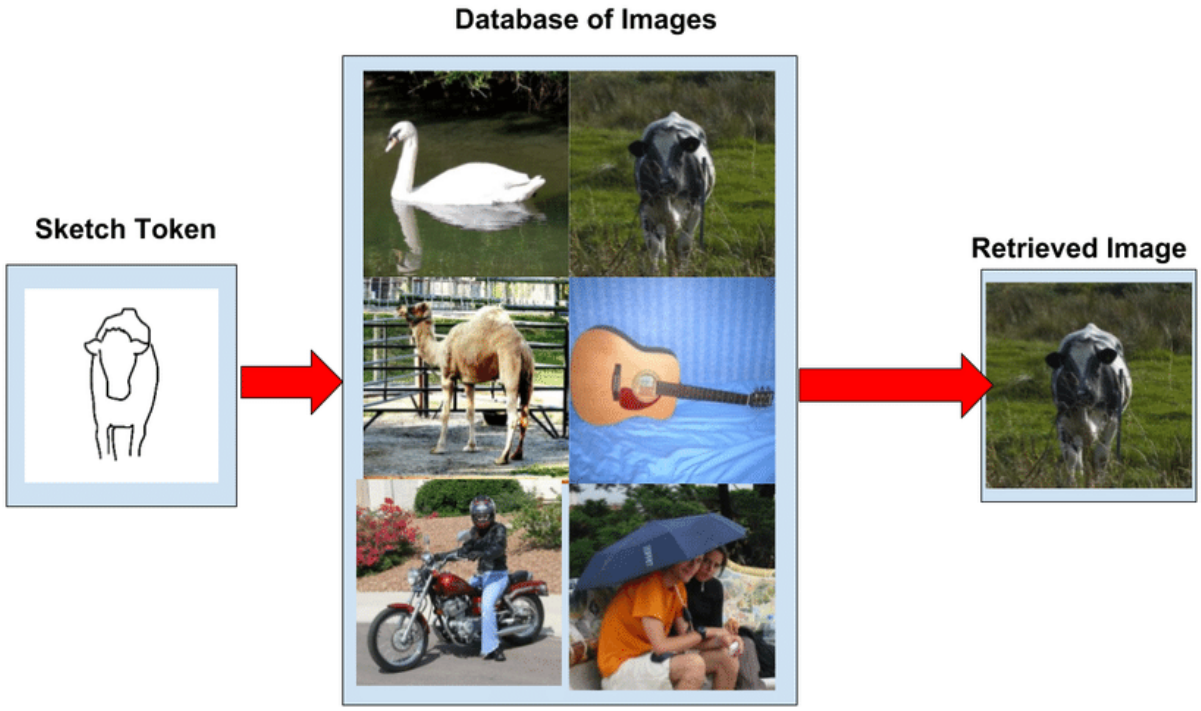
INTERNSHIP REPORT

INTRODUCTION

We are working on the problem of Zero-Shot Sketch-Based Image Retrieval. This internship started on 1st June 2021. This report is a compilation of the work we have done till now. It contains descriptions of methods that have worked, failed, or are at a stage that requires further analysis to proceed.

PROBLEM STATEMENT

The aim of this project is straightforward. We aim at using hand-drawn sketches to retrieve similar images. But the difference between the sketch modality and the image modality makes it challenging. We also aim at performing this task in a Zero-Shot scenario that depicts a real-world problem setting.



BACKGROUND

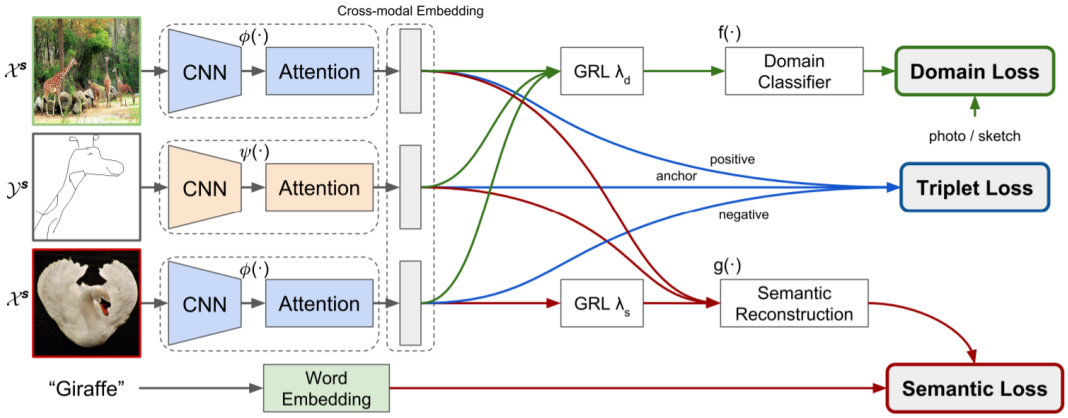
Information retrieval has been a widely studied task. There have been studies on text-to-image, image-to-text. The task of sketch-to-image makes the problem more challenging. Since sketch has very little information than the image, it becomes challenging to get images using sketches. People have tried to achieve this task by either mapping the sketches and images into a common embedding space and then clustering them together or by generating the missing information in the sketches and then trying to retrieve similar images. The Zero-Shot setting makes it even tricky as testing is done on classes that are not used for the training phase. Studies have employed the text modality to inculcate the zero-shot scenario.

Resources And Literature

The initial phase of this internship was spent on reading and understanding the problem and searching and getting an idea of the study done till now. Some of the major research papers which I referred to are given below

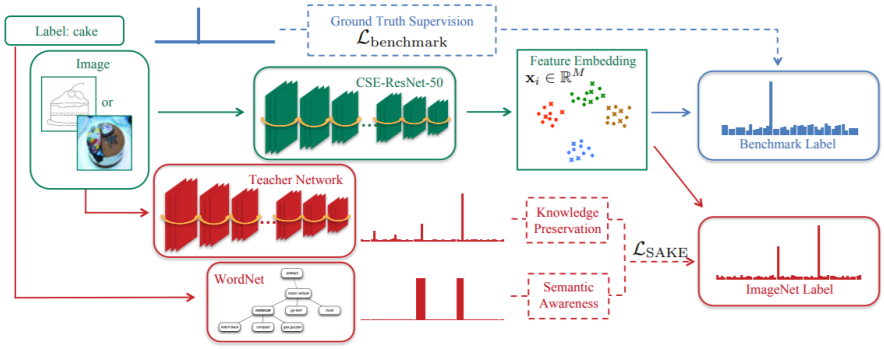
Doodle2Search: Practical Zero-shot training [Link]

This paper aims at mapping the sketches and images to a common embedding space. It emphasizes the problem of the large domain gap between sketches and images, the high degree of abstraction found in the human sketches, and a need for a large practical retrieval dataset. They solve the domain gap problem by using a domain classifier. To transfer the learned knowledge to unseen classes, they use the semantic reconstruction module.



Semantic-Aware Knowledge Preservation for Zero-Shot Sketch-Based Image Retrieval [Link]

This paper focuses on the problem of forgetting the learned features by the existing methods. It says that while fine-tuning the back-bone network, many models forget the precious learned weights, which leads to inefficient use of the knowledge gained. It tries to alleviate this problem by introducing a method to generate pseudo labels from the pre-training dataset and then fine-tuning the SBIR task on these labels.



These two papers form the basis for our model. To save the overhead of writing code from scratch we build our ideas on top of these papers. Some other papers which we referred are given below:

- Semantically Tied Paired Cycle Consistency for Any-Shot Sketch-based Image Retrieval [Link]
- Semantic Enhanced Sketch-Based Image Retrieval with Incomplete Multimodal Query [Link]
- A Zero-Shot Framework for Sketch-Based Image Retrieval [Link]
- Fine-Grained Image Retrieval: the Text/Sketch Input Dilemma [Link]
- Self-supervised Co-training for Video Representation Learning [Link]
- More Photos are All You Need: Semi-Supervised Learning for Fine-Grained Sketch-Based Image Retrieval [Link]
- Heterogeneous Attention Network for Effective and Efficient Cross-modal Retrieval [Link]
- Learning Cross-Aligned Latent Embeddings for Zero-Shot Cross-Modal Retrieval [Link]
- Zero-Shot Learning With Common Sense Knowledge Graphs [Link]

PROPOSED SOLUTION

We approach this problem by mapping the sketches and images to a common embedding space. We plan to introduce three novel ideas in this domain. We plan to use 1) a self-supervised learning strategy if possible, 2) a verbose text query instead of a single-word argument, and 3) inculcate an explainability factor to reason about a particular retrieval. Our motivation behind adopting self-supervised learning is to reduce the need and cost of a labeled dataset. A verbose text query along with the sketch will give the user an added degree of freedom to express the details in the image. Lastly, the explainability component will provide a reason for a particular retrieval so that the user understands how the query was interpreted.

DATASET AND EVALUATION RUBRICS

This section contains the details of the datasets, evaluation metrics, and objective functions that we have used to perform this task. There will also be some briefing over the general method we follow, but more details will be given in the next section.

Dataset

The two major datasets that have been used throughout this project are the 1) Sketchy Dataset (Link) and the 2) TU-Berlin dataset (Link). The Sketchy Dataset contains 125 object classes containing 75,471 sketches of 12,500 objects. The sketches of these datasets resemble the objects closely. The TU-Berlin dataset contains around 20,000 sketches distributed over 250 classes. We generally use an extended version of these datasets that are essentially larger for extensive experimentation.

Evaluation Metric

In general, the information retrieval tasks are evaluated by the Mean Average Precision metric. This is a very intuitive metric for any retrieval task. During retrieval, our priority is to get the relevant images along with getting them at the highest rank possible. For, example retrieving ten images from rank 1 to 10 is preferred over rank 11 to 20. Mean Average Precision (mAP) takes into account the ranking aspect as well. Refer to this for further details.

Objective Function

We have used the following objective functions in the prototypes that we have built. Not all of them have been used at the same time.

Triplet Loss

The Triplet Loss takes as input an anchor embedding (query), a positive image embedding (image belonging to the query's class), and a negative image embedding (any image not belonging to the query's class). Its motive is to bring the anchor and positive image closer and push away the anchor and the negative image.

$$\lambda(\delta_+, \delta_-) = \max\{0, \mu + \delta_+ - \delta_-\}$$

Where *delta_+ and delta_-* is the distance between anchor-positive and anchor-negative, respectively.

L2 Loss

L2 Loss is the usual Euler Distance between the embeddings. We have used it in one of our prototypes. It will be mentioned in the further sections.

Cross-Entropy Loss

This loss has been used for multiclass classification in one of our base papers on which we have built our method.

BackBone Pipeline

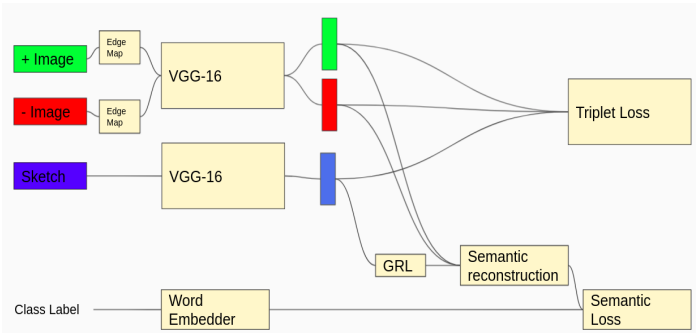
We take inspiration from the doodle2search (Link) paper and the SAKE (Link) paper to build our prototypes. We have made some prototypes on top of the doodle2search paper, which will be mentioned below. Our recent prototype is in progress, and it takes inspiration from the SAKE paper.

PROGRESS TILL NOW

In this section, I will explain in detail my work during this internship. A brief description is as follows. The internship began with the task of reading the various research papers on this topic. Reading the SOTA research papers was a first-timer for me, so it took some time. Each paper had reference to some concepts that had to be understood to proceed with the paper. My job was to get an idea of the ongoing work in this field, to go through and understand the algorithm used, to reason why a particular method has been used, and at the same time try to formulate a solid problem statement. After discussing with my mentors, the next step was to try out the codes of the papers I had read and reproduce the results. At first, this looked quite simple, but then the technical incompatibility of the libraries used in the code and their current versions raised many problems. Installing all the required libraries and replacing the ones which were depreciated, was quite a tedious task. Once, all the prerequisites were ready, running the code on my personal computer was not possible due to the limited memory. Acquiring a remote computing facility from my institute, transferring the code and huge datasets there, and finally, learning to use such a facility took some time. We decided to move ahead with the *doodle2search* paper as our base paper. The following sections will contain the details of the major prototypes which we built. There were some small experimental prototypes also while moving from one idea to the next. But, those will not be mentioned here as they were very trivial.

Prototype 1

Here our base pipeline was similar to doodle2search. Here we used a pre-trained VGG-16 network to get the image embedding. We trained on two different VGG-16 CNN, one for sketch the other for image. Additionally, we also passed the images through an edge_map module to get rid of the extra information present in the image. The algorithm is based on forming triplets of anchor, positive and negative image, and then calculating the triplet loss to optimize over it. Since we were focussing on the zero-shot problem, the class labels were also used as auxiliary information while training the model. There was a semantic reconstruction module employed to reconstruct the semantic information from the image and sketch embeddings. Semantic Loss was employed using this semantic information on the word embedding of the positive/anchor class to bring the embeddings of the anchor and positive image closer to the word embedding and push away the negative image embedding from the word embedding using a Gradient Reversal Layer. The triplet loss and the semantic loss were combined to train the model.



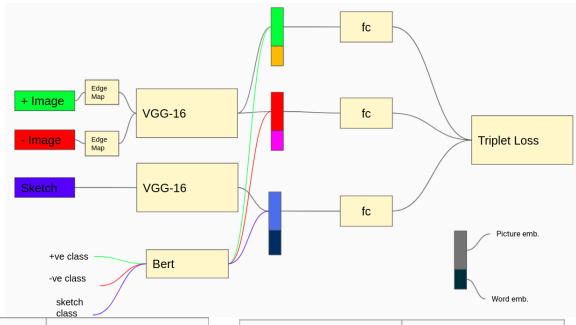
We used the Word2Vec model to get the word embeddings. The results are given in the figure. The results were not attractive. We reasoned for this by considering the use of semantic information. We were dependent on the efficiency of the semantic reconstruction module to transfer the seen information to unseen classes while testing. During testing, we only used the sketch as a query to retrieve the images. The text/ class label was not used while testing. Also, it was difficult to use the class labels in the right context. For example, the word embedding of class ‘bat’ (the bird bat) was closest to classes such as knife, sword, hammer. We concluded that word embeddings need to be improved and other methods to use the semantic information must be used.

Zero-shot	Sketchy Dataset
	mAP@all
Sketch to Image	0.251

Seen classes	Sketchy Dataset
	mAP@all
Sketch to Image	0.497

Prototype 2

The primary pipeline remained the same as prototype 1 except we used the BERT encoder to get the word embeddings. Additionally, we used semantic information by constructing a multimodal query instead of the semantic reconstruction model. The multimodal query for a sketch or an image is its embedding concatenated with its class. An example can be seen in the diagram. All the three multimodal representations were passed through a fully connected layer to adjust the embedding size. We used only triplet loss to train the model. For testing, we use the sketch and its class label (text) to form the multimodal representation of the query. The gallery of images is also concatenated with their respective class label to form the multimodal representation of the image gallery. To retrieve the images a simple clustering is performed with the query as the centroid. The results of this method were



Zero-shot	Sketchy
	mAP@all
Sketch to Image	0.923

Seen classes	Sketchy
	mAP@all
Sketch to Image	0.942

astonishing. The results are given in the table. The mAP was almost double the existing SOTA methods. But, after doing a thorough analysis we found that the retrievals were totally guided by the class labels and the sketch features had very little weight in the decision making process. For example, if we provided a cow's sketch with the class label as 'cow' then it would correctly retrieve all top-k retrievals of cows. But, instead, if we provided the class label as 'giraffe' then all the images retrieved would be of a giraffe although the sketch belongs to 'cow'. Moreover, by visually analyzing the retrievals (sketch query and the corresponding images) over the entire testing dataset, we found that the top-k retrievals were very less varied. Every sketch has some differences in its embedding, so the images retrieved should also have been different. Also, the testing scenario was not feasible as while testing, ideally, we will not have the information of class of the gallery of images. The gallery will be a homogeneous collection of images.

Prototype 3 and 4

Prototype 3 and 4 were almost similar to each other and were an extended version of prototype 2. As our training is based on triplet loss, it is important to choose enough triplets and the correct kind of triplets. In prototype 3 we used L2 loss, which took as input the multimodal query and the image embedding part of this query. Its task was to bring the multimodal query and the image embedding closer. This was done for all three, anchor, positive and negative. The motivation behind it was to devise a method to give more weight to the image/sketch embedding information in the multimodal query, which was one of the drawbacks of the previous prototype. For prototype 4, on top of prototype 3, we introduce another triplet loss. The motivation behind this was to inculcate triplets where, the anchor was concatenated with the positive class, positive image with positive class but this time the negative image was also concatenated with the positive class itself. This was done because our plan was to use a verbose text query while retrieving/testing and we thought of extracting the class information from the text and using it for better retrievals. So, as a part of a trial for this step, we used this triplet loss so that a sketch concatenated with the positve class should be closer to the positive image concatenated with the positive class, (as both the components are matching) and further away from the negative image concatenated with the positive class (as only one component is matching). This modification helped us to move a step closer to the testing scenario by having a homogeneous gallery of images concatenated with the class provided in the query (positive class). The numerical results have not been included here as we did not implement a robust testing scenario.

Prototype 5

Currently, we are working on this prototype. We use the SAKE paper (Link) as our baseline. In the initial phase, we just replaced the wordnet module with a knowledge graph module in the training pipeline. We do this because the knowledge graph is better than the wordnet in terms of finding semantic similarities between two words, which is an integral part of the pipeline. Also, the SAKE pipeline does not use the triplet loss, which avoids the problem of training the model on limited triplets. We adopted this pipeline as we got inspired by the Zero-Shot learning scenario described in this paper. It uses the knowledge graph, and they have a very generic zero-shot retrieval scenario. After using the knowledge graph on the SAKE pipeline we have got some improvements compared to the original paper. Our present task is to verify the changes so that we can be sure of the improvements we have got.

SAKE original pipeline	SAKE pipeline with concept net
mAPall=0.547	mAPall=0.568

CONCLUSION

We are working on prototype 5 and hope to move to the next stage of implementing our ideas. This experience has allowed me to work closely with some of the best researchers in this field. Working on this project has given me a

chance to expand my knowledge, get a good amount of experience in the technological stack used for coding and learn the process behind the building and development of a research project. I hope to complete this project soon.