Team 41 Final Report: Deep Text-Image Retrieval

Angela Zheng, Chengrui Raymond Li, Sarah Jennings, Asa Harbin, Ramachandren Shankar

1 INTRODUCTION

When searching for images on platforms such as Google Image Search, users input text queries and then search algorithms run over related images' metadata (captions, embedded web page content, filenames etc.) but not the images themselves. For example, when searching the query "cat", images are returned based on if the text "cat" is associated with the image through either a caption, related content on the web page in which the image is embedded, or a filename. This is the key weakness of the current practice, i.e., inferring the semantic content of images with metadata rather than using the actual content of the image. If the image metadata is incorrect or unavailable, the results can be easily skewed. Our goal for this project is to create a novel image search system that can search over image content independent of these external attributes.

2 PROBLEM DEFINITION

We will develop an application where users enter text queries and then retrieve related images, agnostic to the semantics of the text query and the content of the images that are being searched over. We will deliver a 3-part system: a browser-based user interface where users can enter text queries and interact with related images, a program that performs calculations to find relevant images with user query data inputs, and a specialized database for storing images in a manner conducive to such calculations.

Formally, our problem is defined as follows. Given a text query q, a tokenization function τ, a set of images D, a deep text encoder E_T that projects to R^m, and a deep image encoder E_I that projects to the same space, we aim to rank the returned image results d according to:

$$argmax_{d \in D} sim(E_T(\tau(q)), E_I(d))$$

where sim denotes the similarity between the embedding vectors. We will utilize the cosine similarity metric due to its inclusion in the training process of the deep encoder networks.

3 LITERATURE SURVEY

Prior research into inferring semantic content of images with metadata uses hierarchical clustering [1] of online image search results, where metadata of HTML, links, and other related textual information in the associated web page were used to form and relate information to an image. EnjoyPhoto [2] is an example where location data, camera settings, and other metadata stored within a photo is utilized to determine the relevance of the photo to search queries. A key weakness of the current practice is exposed here: when textual metadata is not present or is unrelated to the image itself, this approach of using textual metadata ultimately fails [3].

Other methods of search have been researched and documented before. For example, Guigle [4] uses screenshots sorted by parsing elements from an associated app APK. VINS [5] employed students and Upwork freelancers to manually label images. Yale Image Finder (YIF) [6] parses words that are part of an image using OCR and saves the text found as search results. All these methods failed to rely on the visual information contained within the image, and instead focused on other ways to extract pre-existing text.

Additionally, there have been computer vision projects that used image content rather than text. Reverse Image Search [7] demonstrates how deep learning is used to find similarities between two images by matching relevant features. However, only the visual comparison between images was explored, and it does not explore how these features can be accessed through text based search queries.

WebSeer [8], an image search engine, relies on information from text, basic image content (e.g. color, compression type), and basic image structures (e.g. faces).

With language, previous research has shown that pre-training on unlabeled data (unsupervised learning) can improve performance on future tasks. Results indicate that this text processing as pre-training (transfer learning) on web-scaled data actually surpasses pre-training done on high quality, crowd-funded NLP datasets [9]. However, this research is limited as it is only applicable to text processing, and not to image processing.

Lastly, in terms of utilizing both vision and language for classification, there have been attempts to improve fine-grained classification by using natural language descriptions and explanations. A two-stream model (CVL), combining CNN-based vision and language streams to learn latent semantic representations, was created in 2017 [10]. However, this research was limited as it was trained on a relatively small dataset.

4 PROPOSED METHODS

Our approach will leverage advancements in large-scale, multimodal deep learning to conduct text-guided image search over deep representations of the two modalities in a shared embedding space. A key innovation is that we will **search over compressed representations of images instead of metadata**, which is the classical approach to image search [11] utilized by the current state-of-the-art Google Images. As a result, our model should intuitively retrieve a more accurate selection of images as described by the text query, leading to more precise image search. This improved quality, in addition to the fact that previously unsupported metadata search queries could now work, suggests great potential for our model to make significant improvements in the image search domain and therefore better than state of the art.

Concretely, we will utilize an **open-sourced OpenAI model called CLIP** (Contrastive Language-Image Pretraining) [12]. CLIP is composed of two transformer-based neural networks. One takes in images, and one takes in strings of text, with both output vectors in a shared representation space. The models are trained concurrently against each other via a cosine-similarity-based contrastive objective. We compare the semantic similarity of samples from image and text modalities by observing the dot product of their deep representations, and return results based on the magnitude of this similarity score.

In other words, for any text query, we first encode it as a vector with the CLIP text encoder, then compare it against encoded vector representations of all images in the database via the cosine similarity (normalized dot product) score. Then, we return the related images based on the highest similarity scores. This leads us to another key innovation: **the searching process inherently includes the ranking process for returning the most relevant results to a user**, which rids the need for an additional ranking algorithm to be run after search. We expect our similarity score technique to succeed because of the scale of the CLIP encoder models; they are trained on over 400 million positive image-text pairs and even more unlike pairs.

SimCLR [13] is a novel method for learning effective visual representations of data without human supervision, by maximizing "agreement" between differently augmented views of the same data via a contrastive loss in the latent space. SimCLR is simple and has outperformed prior work in this field. However, deep metric learning based on contrastive loss suffers from slow convergence. The multi-class N-pair loss function addresses this by generalizing the loss with joint comparisons and reduces computational burden for evaluating deep embedding vectors [14]. CLIP draws heavily from these two papers and uses a batch construction technique similar to ConVIRT [15], an unsupervised method of bidirectional contrastive objectives to learn image representations from text (limited to medical imaging).

The last key innovation in our approach is the utilization of a **"vector database", optimized for vector similarity lookups**. Vector databases have been studied in detail [16] as a method to store and access information at speed several magnitudes faster than others. TOPOVT [17] was created to store and present 1:25000 or larger scale

topographic vector data, allowing institutions to efficiently retrieve and use it even as the database was being constantly updated.

Currently, image search and ranking results vary over time because algorithms are sensitive to new images and their metadata being added to the internet; statistics must be recomputed across entire image sets in this case. For us however, **once we have a vector representation of an image in the database, our approach does not need to recompute it**. The only new computation to perform when an image is added to the database is obtaining its vector representation. Thus, the use of vector databases in our project greatly reduces the storage space and time spent retrieving data.

As for the visualization, we will create a web application to search for and display the top 5 most-likely results after inputting a text query, and a web of the top 50 most-likely results. We propose this interactive visualization as it will serve as both a familiar yet novel interface for users to explore the results of their search query. This will be familiar in the sense that users input their text query into a search bar, click the button "Search", then see the produced image results on the webpage. Meanwhile, our project also provides novelty in that users will be able to see which images are most similar to their provided query and by how much, while also being able to interact with these results. We accomplish this by having the top 50 images undergo k-means clustering and are then group them into sets of similar images, which are then displayed as an edge-node graph visualization. The green edges of the graph vary in both thickness and saturation. Thicker, less-saturated edges connect images of a higher similarity score while thinner, highly-saturated lines connect less similar images. Additionally, we include the functionality for a user to be able to **click on a desired result to search for more images similar to that user-selected result**. Users can then find images with even more relevance to their desired search query.

Furthermore, we include the capability for **interactive tSNE visualization** of the image embeddings in our system. The quality of image embeddings is crucial for our system, and we can evaluate this quality using tSNE. The tSNE algorithm

Figure 1: Results from Search Query: "Cat"

Figure 2: Results from Search Query: "Cat on car"

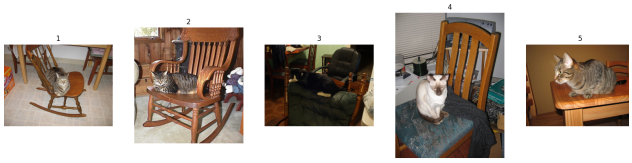

Figure 3: Results from Search Query: "Cat on chair"

[18] projects a set of high-dimensional vectors to a lower-dimensional space in such a way that relative distances between the vectors are preserved as well as possible. Our encoder networks represent images in R^{512}, which is humans are not able to interpret. By projecting them to R^2 with tSNE, the representations become much more interpretable.

Since our image encoder networks are trained with a cosine similarity objective, the distances between points of a tSNE embedding should reflect the relative similarity of the images the points represent. This renders the high-dimensional embeddings interpretable for our users. For example, a user may upload a set of images belonging to known classes and determine the utility of the encoder network by visualizing how well classes are clustered in a two-dimensional tSNE embedding space. Because this only requires knowledge of simple scatter plots, **both technical and non-technical users can verify the efficacy of our encoding approach**. For an example of this capability, please refer to the Experiments section.

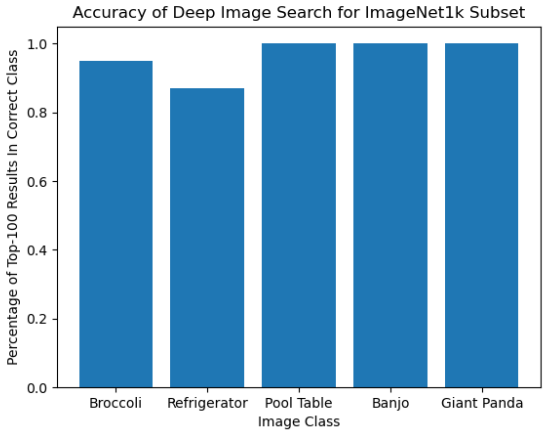

Figure 4: Demonstration of Retrieval Accuracy

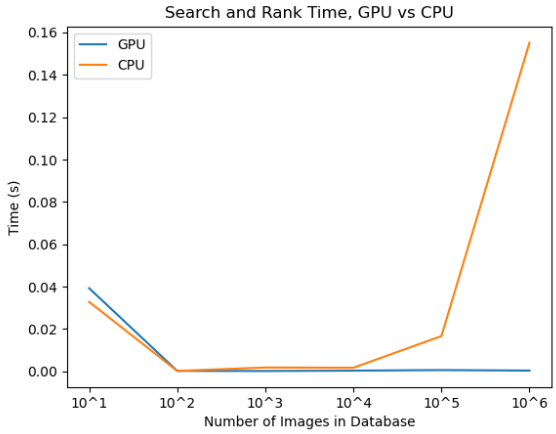

Figure 5: Demonstration of Scalability

5 EXPERIMENTS AND EVALUATIONS

We present four ways we have evaluated our approach. We utilized the **ImageNet1000 dataset**, consisting of around 1.5 million images, each belonging to one of 1000 classes like “canoe” or “mushroom”. We selected this as it is a standard benchmark in deep learning literature and because it was not utilized during the encoder networks’ training, making it a more accurate gauge for determining our approach’s success for end users, who will feed unknown images and text queries to our system.

Our first result is a qualitative demonstration of the system’s capabilities. In Figures 1-3, we demonstrate the top-5 results returned by our system ranked left to right for the given query. These results exemplify the granularity of our approach: not only can we search over the general semantic content of an image, but also the full range of semantic features in the image. This unlocks novel capabilities for end-users like searching for a specific image with highly descriptive text queries, instead of simply image searching across broad categories.

Our second result quantifies the system’s accuracy. We tested the percentage of the top-100 image results belonging to a class specified in the text query. For example, if the class was "refrigerator," we entered that as the text query, encoded it with the text encoder, and calculated the cosine similarity of that vector with the embeddings of all the images in the dataset, returning the top 100 images by highest similarity. We then report the percentage of returned images labelled as the "refrigerator" class in the ImageNet1000 dataset. We repeat this experiment for a random subset of five classes in the dataset and showcase the results in Figure 4. Note that the accuracy of our top-100 search results is very high, and even reaches 100% for three of the classes.

Our third result demonstrates the scalability of our system. Our goal is to allow users to search over huge databases of images with text queries. As these databases grow, the speed of our system must not noticeably decrease in order to provide a high-quality user experience. In this experiment, we demonstrate the search and ranking time across databases containing amounts of images in increasing orders of magnitude. As shown in figure 5, when performing search and ranking with CPU only, our system’s retrieval time increased vastly with the number of images in the dataset. Hence, we elected to optimize our system computations with a GPU, because our query-images similarity calculation can be refactored into one large dot product over an entire vector database’s worth of image embeddings. The time taken to execute searching and ranking grew at a far more gradual rate with this method, indicating that the system will be efficient

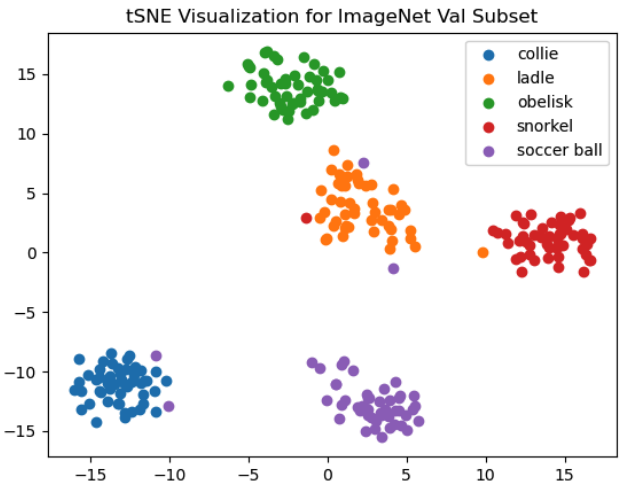

Figure 6: Demonstration of tSNE

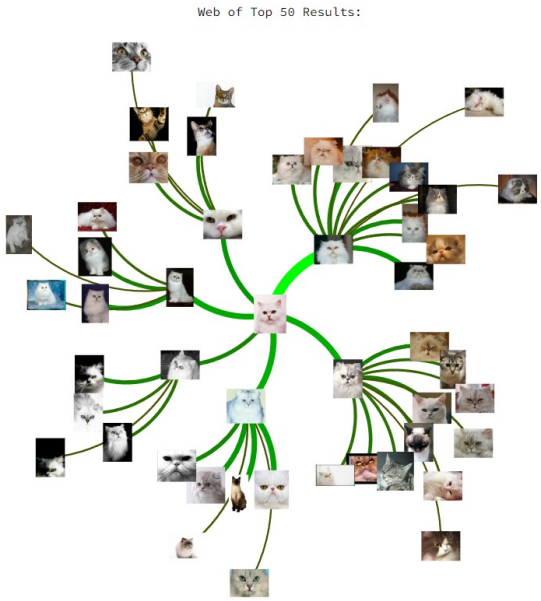

Figure 7: Screenshot of Webpage

even with a larger number of images. For a database of a million images, the search-and-rank process never took more than four milliseconds with only a single GPU.

Our fourth result is an example of the output of our interactive tSNE visualization system. To verify the effectiveness of our image encoding networks, users may first populate our system with a set of images and examine tSNE representations of various subsets. In the example in Figure 6, we have chosen five classes from the ImageNet dataset for reference. We sample from the validation dataset to ensure that the model has not been exposed to the images during training, simulating a real-life scenario. Note that the points are clustered tightly intra-class and well separated between-classes. This indicates that our system effectively distills the semantics of the images.

A screenshot of a top 50 results web for the query "cat" is shown above in Figure 7.

6 CONCLUSIONS AND DISCUSSION

For this project, **all team members have contributed a similar amount of effort**. As a result, we were able to reach our goal of creating a novel image search system that can search over image content independent of external attributes.

The impact and significance that our project has on current image search systems is quantifiable and fixes a key issue with current practice — inferring the semantic content of images with metadata rather than using the actual content of the image. In this case, if image metadata is incorrect or unavailable, the results of the search can easily be skewed.

We developed an interface where users can enter text queries and retrieve related images, agnostic to the semantics of the text query and the content of the images that are being searched over. The system consists of 3 parts: a browser-based user interface where users enter text queries and interact with related images, a program that performs calculations to find relevant images with user query data inputs, and a specialized database for storing images in a manner conducive to such calculations.

As with all studies, there are limitations. Unfortunately, a risk of our approach is the reliance on the dataset used to train the encoder networks. Even after training on a massive dataset and using the multimodal contrastive learning technique — expanding the space of images and text that each model can encode effectively — it is still possible for users to enter text queries far outside the limits

of the training distribution to be encoded effectively. However, this risk's payoff is that a vast number of text queries can be encoded based on their semantics and searched for in a completely predictable amount of time.

Another limitation is that the quality of the returned results is dependent on the set of images that are in the database. If there is no image in the database related to an entered text query, the results will be poor. For example, although this dataset includes a variety of photos spanning from refrigerators to broccoli to cats, it does not contain any images of squid. Though we take on this risk by using a limited, relatively small (compared to the public internet) set of images to search over, the payoff is that this approach has been shown to scale predictably to image sets of arbitrary size.

Future implications of this project include improvements to the quality of image search results, which have the potential to impact billions of users worldwide. Our proposed method improves on the quality of image search results by conducting the search process directly over the images' semantic content. Furthermore, there are situations where image search is currently not possible, specifically collections of images that do not have rich textual metadata such as photo albums. Our proposed method allows for image retrieval via text queries in this new set of scenarios. Extensions to this project may include refining our system and scaling the project for larger datasets, and eventually evolving into a novel video search system as well.

7 BIBLIOGRAPHY

[1] D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen, "Hierarchical clustering of www image search results using visual, textual and link information," in *Proceedings of the 12th Annual ACM International Conference on Multimedia*, MULTIMEDIA '04, (New York, NY, USA), p. 952–959, Association for Computing Machinery, 2004.

[2] L. Zhang, L. Chen, F. Jing, K. Deng, and W.-Y. Ma, "Enjoyphoto: A vertical image search engine for enjoying high-quality photos," in *Proceedings of the 14th ACM International Conference on Multimedia*, MM '06, (New York, NY, USA), p. 367–376, Association for Computing Machinery, 2006.

[3] K. Desai and J. Johnson, "Virtex: Learning visual representations from textual annotations," 2020.

[4] C. Bernal-Cárdenas, K. Moran, M. Tufano, Z. Liu, L. Nan, Z. Shi, and D. Poshyvanyk, "Guigle: A gui search engine for android apps," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pp. 71–74, 2019.

[5] S. Bunian, K. Li, C. Jemmali, C. Harteveld, Y. Fu, and M. S. Seif El-Nasr, "Vins: Visual search for mobile user interface design," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, (New York, NY, USA), Association for Computing Machinery, 2021.

[6] S. Xu, J. McCusker, and M. Krauthammer, "Yale Image Finder (YIF): a new search engine for retrieving biomedical images," *Bioinformatics*, vol. 24, pp. 1968–1970, 07 2008.

[7] P. N. Singh and T. P. Gowdar, "Reverse image search improved by deep learning," in *2021 IEEE Mysore Sub Section International Conference (MysuruCon)*, pp. 596–600, 2021.

[8] C. Frankel, M. J. Swain, and V. Athitsos, "Webseer: An image search engine for the world wide web," 1996.

[9] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *CoRR*, vol. abs/1910.10683, 2019.

[10] X. He and Y. Peng, "Fine-grained image classification via combining vision and language," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, jul 2017.

[11] J. Donahue and K. Simonyan, "Large scale adversarial representation learning," 2019.

[12] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021.

[13] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020.

[14] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information*

Processing Systems (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, Curran Associates, Inc., 2016.

[15] Y. Zhang, H. Jiang, Y. Miura, C. D. Manning, and C. P. Langlotz, "Contrastive learning of medical visual representations from paired images and text," 2020.

[16] R. Stata, K. Bharat, and F. Maghoul, "The term vector database: fast access to indexing terms for web pages," *Computer Networks*, vol. 33, no. 1, pp. 247–255, 2000.

[17] A. Yilmaz and M. Caniberk, "Real time vector database updating system: A case study for turkish topographic vector database (topovt)," *International Journal of Engineering and Geosciences*, vol. 3, 06 2018.

[18] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," 2008.