Signed cards analysis

Jo Pan
tul02009@temple.edu
May 5, 2021

Abstract

Postcard dataset is uncommon for computer vision domain. In order to help an online postcard business, Signed Cards, to improve their search an recommendation system, this project tried to solve three vision tasks: image classification, object detection and image retrieval. This project is the initial work for evaluating computer vision models on this SignedCard dataset. In terms of method innovation, I improved the CLIP model with dual-input structure and a variety of aggretation methods.

1 INTRODUCTION

Online greeting card business SignedCards.com has a growing collection of greeting cards. To improve the search system and the recommendation system, the company has spent significant efforts in annotating cards with related labels, including labels for objects and abstract visual concepts, such as "Girly" and "Romantic". In the current practice, annotations are created by card designers with low consistency. In addition, the created annotations are often incomplete with missing labels. As the result, for SignedCard company, correctly categorizing the cards is non-trivial, error-prone, and labor intensive.

In this project, with the dataset my team gathered from SignedCard, I aimed to improve the existing annotation system and the recommendation system with computer vision models. In specifics, I explored and modified the state-of-the-art models in solving three vision tasks: 1) image classification, 2) object detection, and 3) image retrieval.

From the computer vision perspective, my data contains multiple challenges. First, unlike the popular vision dataset, my dataset is sparse with great variety. Most of the existing labels only have small amount of samples. For instance, in the special occasion category, more than 60% of the unique labels have less than 20 samples. In addition, the appearance of the visual objects is often diverse with stylistic drawing depiction. Thus, I expect regular supervised algorithms to have hard times in learning the general representation with this limited and sparse dataset.

The second challenge comes from classifying the abstract visual concepts. Abstract concepts classification is a fairly new computer vision domain. Ahres et al. [1] used supervised VGG-16 on predicting abstract concepts in the FLICKR dataset and reported precision less than 10%. Abstract concepts labels are usually more challenging than object labels because abstract concepts labels represent a larger variety of images than object labels.

The contribution of this project comes in three-fold. First, this is the initial work for building the postcard dataset. Also, this is the initial work for evaluating performance of common computer vision tasks on this dataset. Second, this project extends the research efforts in solving the abstract visual concepts challenging with the recent state-of-the-art models. Third, I further explained the reasons behind the strong performance of CLIP [2]. I also improved the CLIP structure with aggregator module to leverage the combination of image and text inputs.

2 DATA

The data set was collected directly from SignedCards and I have no authority to share with the public or the reviewers. The data set consists of a catalogue of greeting cards. Near duplicates, such as the same card image with different text on it, were removed to prevent over training. The data set comprises 8033 unique cards.

For each sample, cover image and cover text are provided as input of the visual tasks. There are two types of annotations: single-label and multi-label. Single-label categories include Holiday, Special Occasions, Messages, and Relationship. Each sample has one label for each single-label category. The top-10 labels and the number of unique labels for each single-label category is shown in Fig. 2.1. As shown, all categories are significantly imbalanced. For example, in the Special Occasion category, 'birthday' label has more than 200x samples than the remaining labels.

Features is the only multi-label category, which has a list of labels for each sample. Features category contains not only the object labels, but also the abstract labels, though often incomplete. There are 1334 unique labels for Features. However, many labels are unique in the dataset and are over-specified, such as "1950_FORD_CUSTOM". Thus, for the Features category, I focused only on the labels with 5 or more examples in the data set and discarded those labels with less than 5 examples.

Table 2.1 shows the summary of all categories, with the size of not-none samples (N) and the number of unique labels. Unless otherwise specified, all the reported results are the testing results from all N samples.

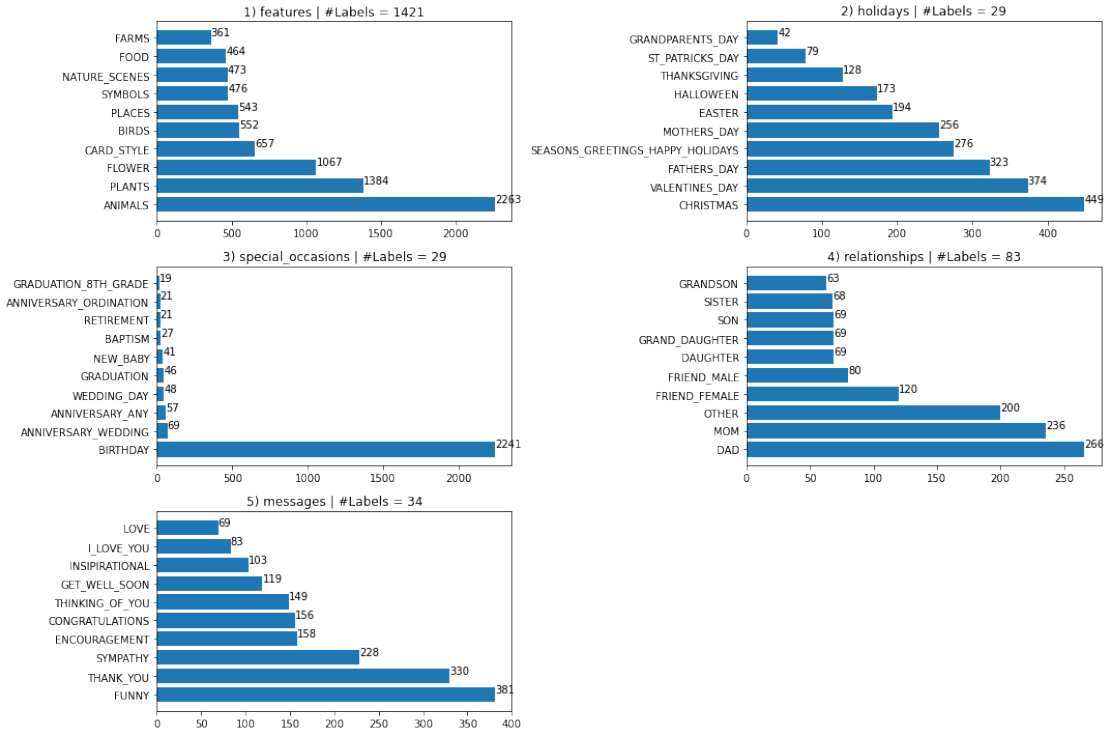


Figure 2.1: Top-10 labels for each single-label category. Total number of unique labels for each category is shown in the subtitles. Horizontal axis represents the sample counts for each labels.

Table 2.1: Summary of all annotation categories

Category	N	#Labels
Holidays	2427	29
Special Occasions	2735	29
Relationships	1847	83
Messages	2196	34
Object features	6699	500

3 Methods

In this project, I tried to solve for three visual tasks: image classification, object detection and image retrieval.

Label classifications are performed with the single-label categories, which are Holiday, Special Occasions, Messages, and Relationship. For each target category, I classify each image with one label under that category.

Object detection is performed with the multi-label category: Features. I try to detect the top-5 most likely objects present in the image. For this task, objects labels are derived from

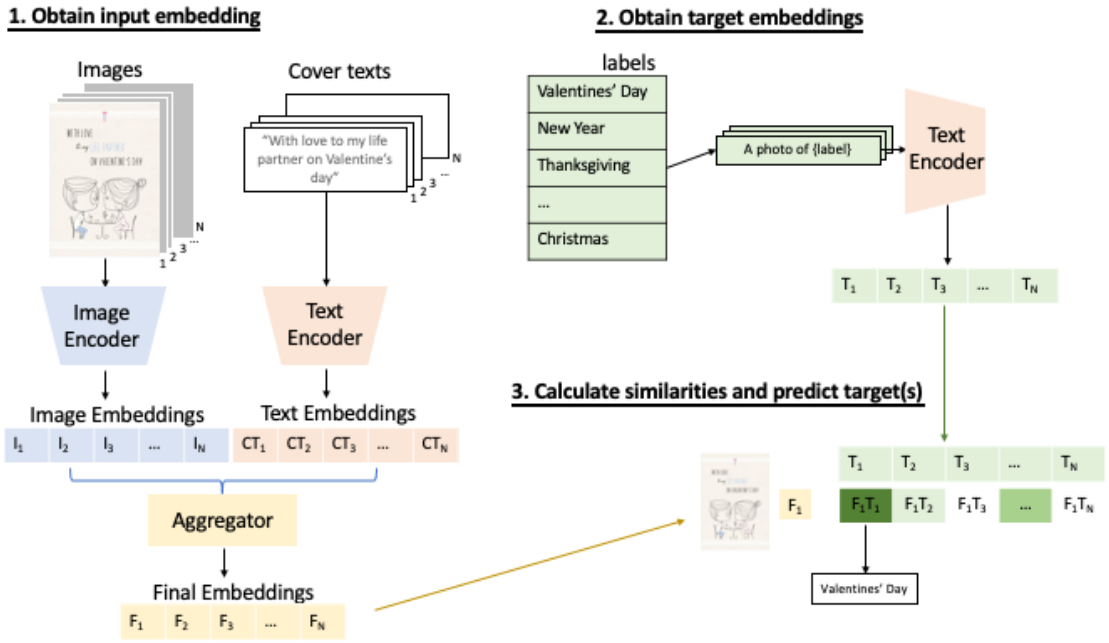


Figure 3.1: Summary of my approach. First, I obtain an input embedding with pre-trained image encoder, pre-trained text encoder and an aggregator. Second, I convert the target labels into sentence and obtain related target embedding. Lastly, I predict labels based on the similarity between input embedding and target embedding.

Features. I keep only object-related labels and remove abstract visual concepts labels. Unlike traditional object detection, no location will be predicted as the location information is not needed for the online retailer to annotate its cards.

For image retrieval, given a query image, I try to retrieve the images in the dataset with the same labels. Experiments are done for the single-value categories. I also try to retrieve images using text queries containing abstract concepts.

For all visual tasks, I used the general structure as illustrated in Fig. 3.1, which is inspired by CLIP [2]. CLIP is a model with an image encoder and a text encoder. It efficiently learns visual concepts from natural language supervision. Since CLIP was trained on an enormous dataset with 400 million (image, text) pairs, it has learnt a large variety of concepts and able to produce useful image representations for a large variety of tasks, without any dataset specific training. As the signedcards dataset also contains a large variety of objects, scenes and abstract visual concepts, I expect CLIP to be advantageous for the proposed vision tasks.

At the core of my approach is the idea of leveraging well-trained encoders, including CLIP and Universal Sentence Encoder [3], to obtain representative embedding for the input image, input cover text and the target labels. Unlike CLIP which only allows image input, I utilize both image and cover text. I modify the original CLIP structure by adding an aggregator function for building a more representative embedding than the original image embedding.

3.1 IMAGE ENCODER

For image encoder, I tested with pre-trained CLIP's image encoder and Resnets [4]. CLIP's image encoder follows the ResNet50 [5] structure. I directly used the pre-trained weights downloaded from GitHub and did no fine-tuning on specific targets. In terms of preprocessing, I first resized them to 224 x 224 pixels. Then, I performed center cropping on images. Lastly I normalized images with the mean and standard deviation provided by CLIP. After pre-processing, I passed the preprocessed images into CLIP's encoder and obtain the image embeddings

Besides using the pre-trained image encoder, I also tested including ResNet-18 and ResNet-152 [5] with no pre-trained weights. Due to the outstanding intra-class variability of visual content in these images, I expect it has a worser performance than CLIP's pretrained encoder.

3.2 TEXT ENCODER

For text encoder, I tested with CLIP's text encoder and Universal Sentence Encoder (USE) [3].

CLIP's text encoder is based on the Transformer [6]. Again, no fine-tuning was done with the CLIP's text encoder. In terms of preprocessing, labels are first converted to sentences. For single-value categories, the sentence for each label is constructed as "This is a photo of label, a type of category". For Features categories, I first manually derived the group of each label. For example, the group of "dog" label, is "animal". The sentence for each label is constructed as "This is a photo of label, a type of group", when group is available. For labels without any group, or itself is a group, for example "animal", the sentence is simply constructed as "This is a photo of label".

After that, all sentences, including cover text and the label sentences, are tokenized, bracketed with [SOS] and [EOS] tokens, and then mapped to unicodes. Lastly, sentences are passed into CLIP's text encoder and then a text embedding is obtained for each target label and each input cover text.

For Universal Sentence Encoder (USE) [3], which is also based on transformer. It is pretrained on textual data from Wikipedia, web news, web question-answer pages and discussion forums. Using a similar preprocessing process, I used USE to generate a text embedding for each cover text.

3.3 AGGREGATOR

Besides directly using either image embedding or text embedding as the input, aggregator module is experimented for combining the image embedding and text embedding. Due to the time limitation, I only tested two aggreagator methods. One is averaging the selected embedding, for example averaging CLIP's image embedding and CLIP's text embedding.

The second aggregator method is TRIG. I modified the original network [7] by removing the original resnet and LSTM layers and replacing them with pretrained visual and textual embedding. Then, I computed gated residual features as described in the original work. The TIRG layers are then trained with triplet marginal loss and with semi-hard triplet batch formation for each category.

Table 4.1: Classification accuracy of the single-value categories.

	holidays		**special occasions**		**relationships**		**messages**	
Input embedding	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
image	0.632	0.950	0.531	0.877	0.338	0.565	0.278	0.509
cover text	0.775	0.891	0.595	0.942	0.449	0.732	0.336	0.508
image & text (avg.)	**0.810**	**0.969**	**0.657**	**0.964**	**0.530**	**0.768**	**0.397**	**0.608**

4 Experiments and Results

4.1 Image classification

I tested with three types of input embedding: image embedding only, cover text embedding and the aggregated feature, which is the average feature of image and text embedding. TIRG is not tested for this task because of time limitation. USE is not tested because it can't map CLIP's imaging embedding trained directly with related USE target label embedding as they are trained with different feature spaces.

All the embedding are obtained with CLIP's encoders. Table 4.1 shows the average classification accuracy for all the single-value categories. As demonstrated, by using a simple average aggregation function, I significantly improved accuracy for all categories. The results matched expectation as the combined embedding provides more semantic contents than the individual embedding. Another observation is that, the Top-5 accuracy is significantly higher than Top-1 accuracy. This means that, for most cases, when the predicted label is not the ground truth label, the ground truth label is still one of the predicted top-5 most likely labels.

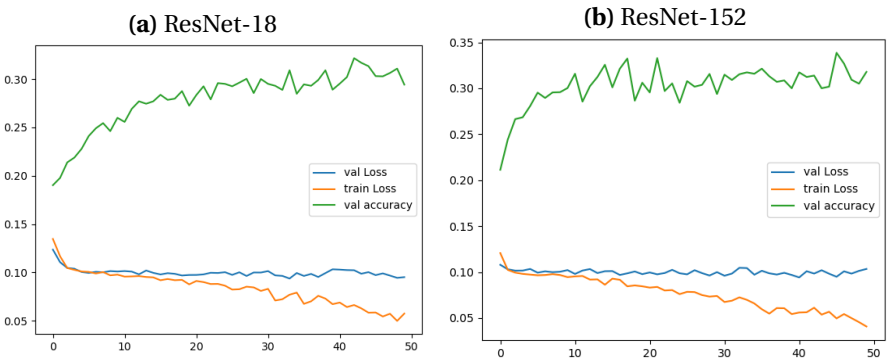
I also tested a baseline network, ResNet with no pre-trained weights. I used a split of 80% and 20% data for training and evaluation. ResNet-18 and ResNet-152 gives me around 32% and 34% accuracy for Holiday classification. The epoch-wise training and validating evaluation for ResNet-18 and ResNet-152 is shown in Fig.4.1. As expected, the dataset is too limited for training a network to learn such sparse label representations and thus causing a bad performance. Since the preliminary results on classifying Holidays were bad, I did not use it for image encoder for the following experiments.

4.2 Object detection

Since object detection is highly dependent on the semantic embedding and textual embedding provide very little information, I only used image embedding as the input embedding. For each image, I predicted the top-5 most related object labels. For post-processing, I expanded the predicted labels with related group label and relevant labels. For example, if one of the predicted label is "dog", I then add "animal" into the predicted labels.

Before post-processing, 62.4% of the samples have at least one object label being predicted correctly. After post-processing, the rate increased to 81.1%.

Figure 4.1: ResNets performance on classifying Holidays



Sample-wise accuracy is calculated as follows:

$$Acc = \frac{min(TP,5)}{min(lengthoflabels,5))} \tag{4.1}$$

Average accuracy is 34.1% before post-processing, and is 61.5% after post-processing. Fig. 4.3 visualize object detection results for 4 samples. As demonstrated, the model can detect objects on a wide variety of images, including the cartoonistic images, stylized images and natural images. Even for the case it missed the ground truth labels, it still predicted highly relevant object labels. Also, by examining the predicted results, I discover that the ground truth object labels are often incomplete, as it misses objects that appear on the images. Thus, even the average accuracy number is low, but I think the object detection performance with CLIP's embedding is solid.

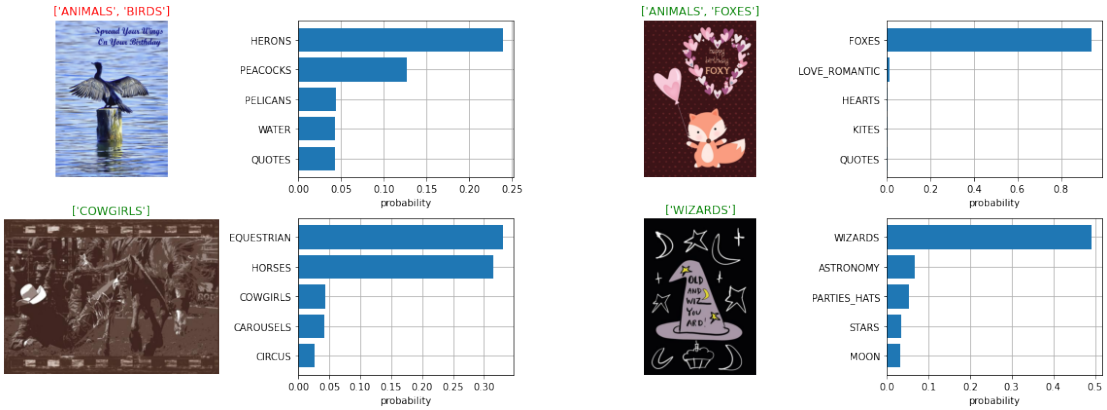


Figure 4.3: Object detection sample results. The label list on top of each images are the ground truth object labels. Red-color means before post-processing, none of its predicted labels is one of the ground truth, and green-color means at least one is the ground truth. Predicted labels with its softmax probabilities are plotted as bar graph.

4.3 IMAGE RETRIEVAL

For image retrieval, unlike other tasks, I evaluated on the test samples only, which is 20% of all the available samples. This is because methods required training also being evaluated. For each evaluated single-value categories, I used each sample's input embedding as the query and retrieved K images. For each sample, recall is calculated based on if the ground truth label of the query image appears in the top-K retrieved images. Table 4.2 shows the average recall. For the listed input types: "im." represents CLIP's image embedding; "text" represents CLIP's cover text embedding. "USE" represents USE's cover text embedding. "(A)" denotes that it is using average as the aggregator function. "(T)" denotes that it is using TRIG as the aggregator function.

For embedding generated without the need of training, using image embedding obtained from CLIP image encoder and text embedding obtained from USE text encoder performed the best across all aggregation schemes, and it is even better than the combination of all three types of embedding. One of the possible reason is the CLIP text encoder and the CLIP image encoder are both trained to capture the same set of semantic features. Since USE is trained separately, it is able to enhance CLIP's image embedding space by providing additional semantic features and thus allow a better performance.

For embedding generated with the need of training, four models are trained with triplet marginal loss and with semi-hard triplet batch formation for each category. The first model is CLIP, which I fine-tuned on original embedding. With fine-tuning, performance increased for Holidays category but not the other categories. The second model is the original TRIG model with its own LSTM text encoder and ResNet encoder. For all categories, it had a lower recall rate than the modified TRIG with non-original encoders. The third and forth model are the models which froze the encoder layer and only fine-tuned the TRIG's gated residual layers. The best performance came with using CLIP's image embedding and USE cover text embedding. Interestingly, except the third model, all other trained model performed worser than non-trained models for Messages category, which is a highly sparse category.

4.4 ABSTRACT CONCEPT RETRIEVAL

I also explored the possibility of retrieving images with abstract-concept text queries. The text queries were first encoded by CLIP's text encoder and compared to CLIP's encoded image embedding for all samples. USE text encoder was not used because it maps embedding into different feature space than CLIP's image embedding. No quantitative evaluation was done because most of the samples in the dataset has no abstract concept labels. Fig. 4.4 shows sample outputs. Overall, the retrieved results are highly impressive as most of them are highly relevant with the abstract concepts specified, even for the non-English query "bon voyage".

4.5 EMBEDDING ANALYSIS

Since both CLIP and USE show impressive results in the analyzed vision tasks without fine-tuning, thus I would like investigate the underlying feature spaces of the pre-trained embeddings. I used t-SNE to reduce all the 512-dimensional embedding obtained from the discussed encoders to 2 dimensions. Fig. 4.5 plots the t-SNE components for the samples with the top-10

Table 4.2: Top-K retrieval recall rate for all embedding generation method

	Holidays			Special Occasions			Messages		
Embedding	1	5	10	1	5	10	1	5	10
without training									
USE	0.679	0.648	0.605	0.889	0.839	0.822	0.502	0.466	0.415
text	0.687	0.69	0.647	0.871	0.844	0.832	0.478	0.414	0.368
im.	0.711	0.636	0.566	0.835	0.804	0.799	0.347	0.316	0.28
im. & USE (A)	0.77	0.717	0.66	0.877	0.828	0.819	0.5	0.462	0.417
im. & text (A)	0.785	0.741	0.702	0.866	0.828	0.814	0.485	0.406	0.374
im.&text&USE(A)	0.782	0.736	0.683	0.877	0.824	0.813	0.485	0.433	0.387
with training									
CLIP	0.799	0.733	0.7203	0.755	0.724	0.718	0.171	0.111	0.102
original TRIG	0.734	0.719	0.708	0.713	0.682	0.677	0.129	0.094	0.088
im. & USE (T)	0.831	0.777	0.754	0.85	0.815	0.802	0.728	0.707	0.697
im. & text (T)	0.767	0.741	0.713	0.712	0.687	0.674	0.134	0.105	0.1

Figure 4.4: Retrieval with text queries containing abstract concepts. The leftmost column shows the text queries used. The subtitles shows the probabilities.

labels in each category. Generally, Holidays category's embedding is the most well-separated. This gives a hint why the performance of all visual tasks have the highest accuracy on the Holiday category.

Since the aggregated embedding from CLIP's image embedding and USE's text embedding gives the highest image retrieval accuracy, I wanted to visualize how this aggregated embedding

map similar images into similar semantic feature space. As shown in Fig. 4.6, I embedded 1000 sample images in 2-dimension using t-SNE on the aggregated embedding. As demonstrated, without fine-tuning, the aggregated embedding are able to map images which share the same semantic meaning to be neighbors. For example, in Fig. 4.6, the images with dogs are mapped to the upper left corner and the images with the romantic theme are mapped to the upper right section.

5 Conclusions

In conclusion, I have evaluated visual tasks that are crucial for SignedCard website's recommendation system and searching system, including image classification, object detection and image retrieval. I demonstrated that CLIP and USE, which are both pre-trained on large variety of images, are useful for improving the performance. Also, I proved that CLIP's pre-trained embedding has learnt about abstract visual concepts and thus allowed good retrieval performance with abstract queries.

In terms of model improvement, I improved the original CLIP model by accepting dual-inputs: image embedding and text embedding. I also experimented a naive aggregation method, averaging, and an advanced gated residual method, TRIG. In the image retrieval evaluation, I noticed that by combining CLIP's pre-trained image embedding and USE's pre-trained cover text embedding, and fine-tuning the TRIG layers was the best model choice with highest recall rate.

Due to time constraint, some of the visual tasks are not evaluated completely. For example, for the retrieval experiments, only three categories are evaluated instead of all four single-value categories. As future works, to demonstrate the capability of proposed aggregation method, evaluation on bench mark datasets, such as ImageNet, can be done.

6 Acknowledgement

This project is a collaborative work done with the research team lead by Professor Longin Jan Latecki. Team members includes Sidra Hanif, Tom McLaughlin, Patrick Ammons, and Alexander Kim. Jo Pan (author of this paper) contributed at least 75% of the discussed experiments and 90% of the this paper.

Figure 4.5: t-SNE components for samples with the top-10 labels in each single-value category.

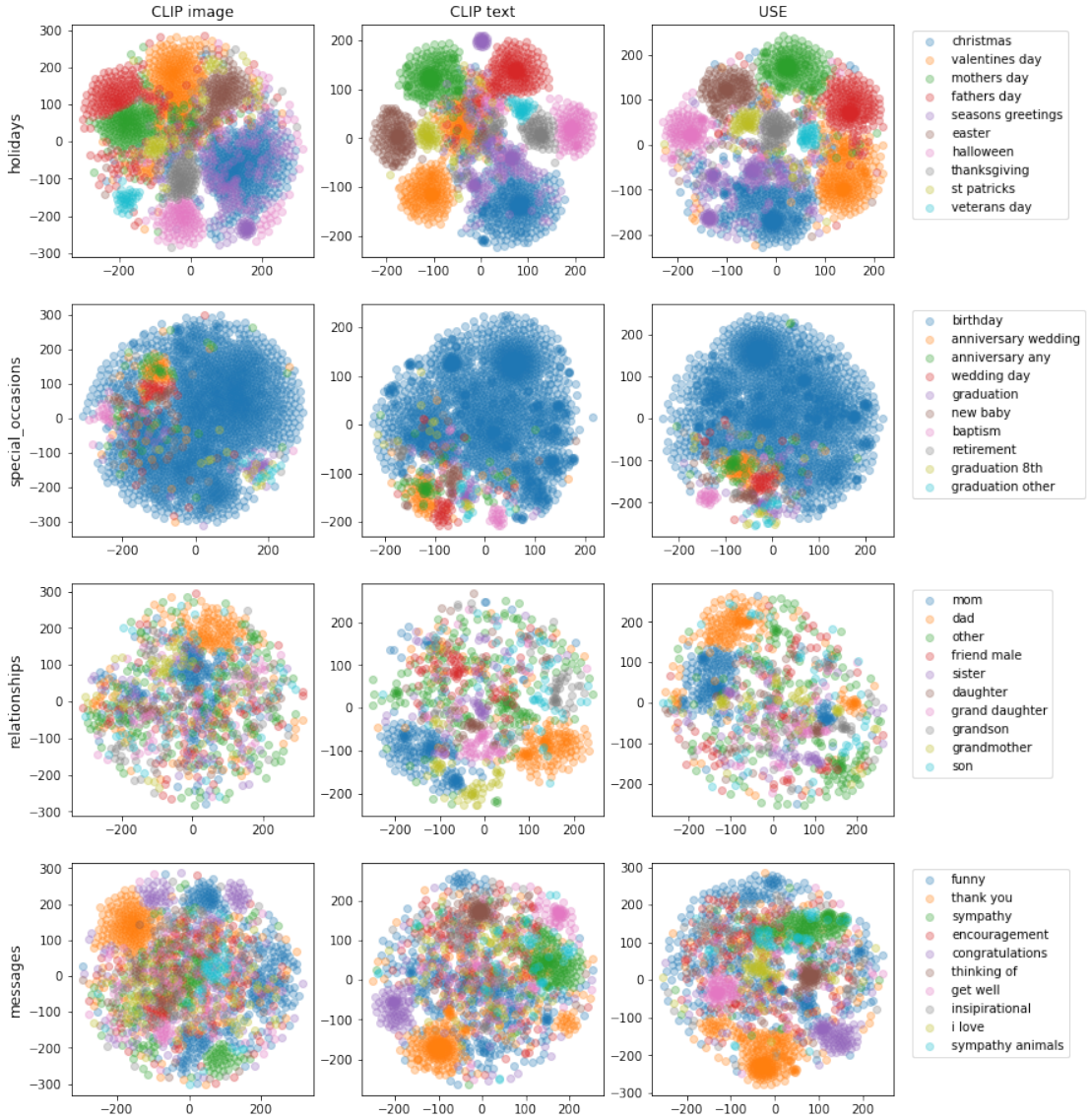
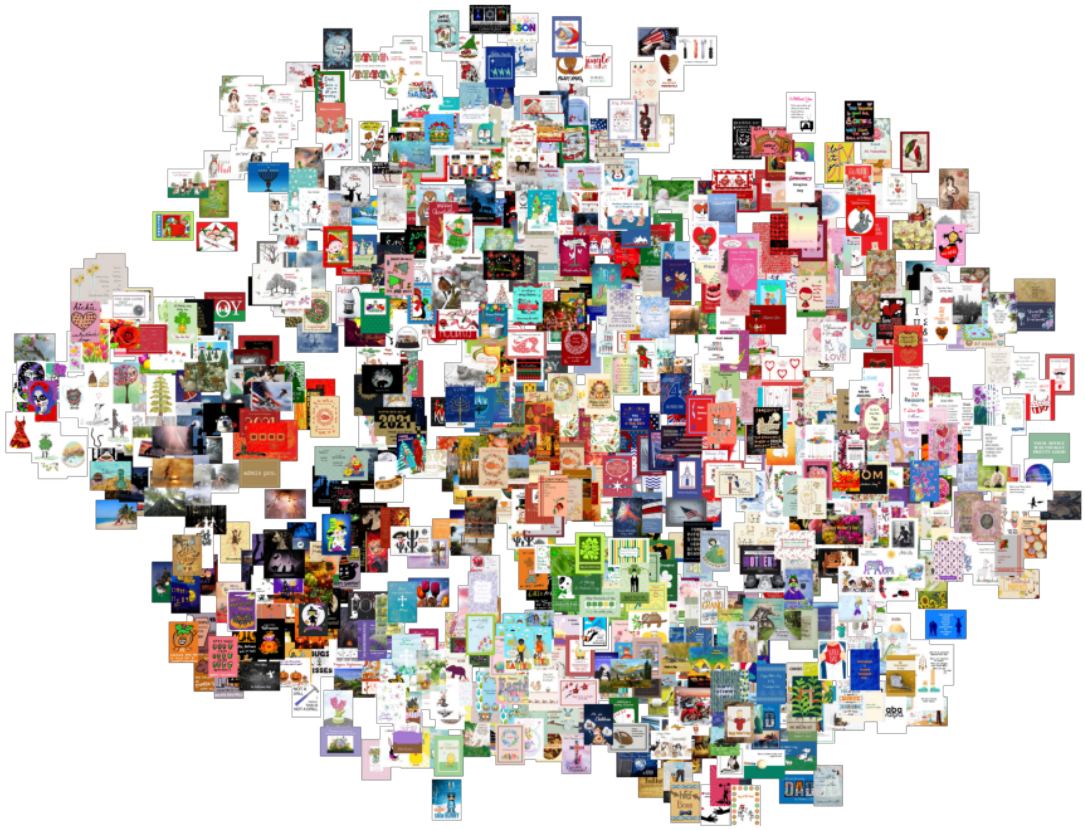


Figure 4.6: Embed 1000 images in 2d using the t-SNE on the aggregated embedding.



REFERENCES

[1] Youssef Ahres and Nikolaus Volk. Abstract concept and emotion detection in tagged images with cnns, 2016.

[2] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[3] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder, 2018.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2016.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[7] Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. Composing text and image for image retrieval - an empirical odyssey. *Thecvf.com*, pages 6439–6448, 2019.