Landmark Image Retrieval Using Representation Learning

Yu Wang
University of Toronto
russell.wang@mail.utoronto.ca

Yitong Wang
University of Toronto
yitong.wang@mail.utoronto.ca

Dongfang Cui
University of Toronto
dongfang.cui@mail.utoronto.ca

Abstract

In this paper, we extend two deep learning methods to perform image retrieval on Google Landmark Dataset v2. Image retrieval provides an efficient method that helps to search for related visual information within a large database. Image retrieval can be formatted as a representation learning problem where we construct an embedding space for querying similar landmark images. Our network uses ResNet-101 pre-trained on ImageNet for feature extraction and we apply two representation learning algorithms to maximize the intra-class compactness and the inter-class discrepancy of the embeddings extracted from the landmark images. We demonstrate that these two algorithms can be transferred to the image retrieval task beyond its application in the original domain, and one of the algorithms ArcFace obtains a superior retrieval performance.

1 Introduction

The primary goal of image retrieval is to query a base image by analyzing the relevance of all the contents in an image database and collecting data that is similar to the base image. A large-scale benchmark, the Google Landmarks Dataset v2 (GLDv2) [13] can be used to evaluate the performance and generalization of image retrieval techniques, and it contains more than 5 million images of human-made and natural landmarks worldwide.

(a) Query Image (b) Index Image (c) Index Image (d) Index Image

Figure 1: Images from GLDv2

In this paper, we implement two representation learning algorithms for landmark image retrieval. The first algorithm is inspired by *Generalized End-to-End Loss for Speaker Verification* [11], which was proposed to perform speaker verification by leveraging the centroids of the embedding vectors for different speakers to find representative clusters. The second algorithm, *Additive Angular Margin Loss (ArcFace)* [2], adds an angular margin to the angle between the features and target weights in each dimension of class, which modifies the cross entropy loss to achieve more distinguishable embeddings [2]. To compare the two algorithms, we use the same ResNet-101 [3] network pre-trained on ImageNet [1] as the encoder before the representation learning stage. We also design a variant of U-Net [9] as the baseline to learn low-dimensional embeddings during image reconstruction.

Preprint. Under review.

2 Related Works

Our work is related to image recognition, transfer learning, and representation learning. ResNet [3] has achieved great success and is widely used for image-related tasks. Transfer learning [7] has been used to mitigate data scarcity and has demonstrated promising results in many deep learning tasks. Cross entropy loss is commonly used for image recognition but it only focuses on classification accuracy with a low level of feature discrimination. To improve feature discrimination, Center Loss [8] was introduced to learn and reduce the distance between the features and their corresponding class centers. Multiplicative angular margin and additive cosine margin were introduced in SphereFace [5] and CosFace [12] to further push the features in different classes into a more compact space.

3 Methods and Algorithm

Our method is mainly composed of three components: Encoder, Projection Network, and Representation Learning Loss Function. Given an input batch of $N \times M$ images, where N is the number of different landmark classes and M is the number of images per class, we can input the image $\mathbf{x}_{ji}$ (j-th image from the i-th class) into the encoder network to get an output vector $\mathbf{r}_{ji}$ of dimension 2048. We use a pre-trained ResNet-101 as our encoder network and replace its last fully connected layer with a sequence of *dropout*, *fully connected* and *batchnorm-1d* layers, which compose our projection network. The projection network maps $\mathbf{r}_{ji}$ to an embedding vector $\mathbf{e}_{ji}$ of dimension 512, and L2-normalization will be applied to $\mathbf{e}_{ji}$. At inference time, $\mathbf{e}_{ji}$ is used for retrieving related images in the database. During training, we will apply the loss function to learn distinctive embeddings that are useful for retrieval tasks. We use mean average precision (mAP) as our primary metric to evaluate the embeddings extracted from the test and index set using the encoder of different algorithms.

3.1 Generalized End-to-End Loss (GE2E)

Our first retrieval algorithm is inspired by an existing approach to solve the speaker verification task [11]. The main idea is to construct a similarity matrix $\mathbf{S}_{ji,k}$ which holds the cosine similarity between each image embedding $\mathbf{e}_{ji}$ and the centroid $\mathbf{c}_k$ of all the embeddings for each landmark class k. We remove $\mathbf{e}_{ji}$ when calculating the centroid, which is the average of the embedding vectors within the same class. The similarity matrix will be scaled by two learnable parameters w and b.

$$\mathbf{S}_{ji,k} = w \cdot \cos(\mathbf{e}_{ji}, \mathbf{c}_k) + b, \qquad L(\mathbf{e}_{ji}) = -\mathbf{S}_{ji,j} + \log \sum_{k=1}^{N} \exp(\mathbf{S}_{ji,k}) \tag{1}$$

We use one of the two loss functions proposed in the paper – the "softmax" approach, where we try to make the embedding of each image close to the centroid of that landmark class's embeddings but also far from other classes' centroids. The whole process is shown in the figure below.

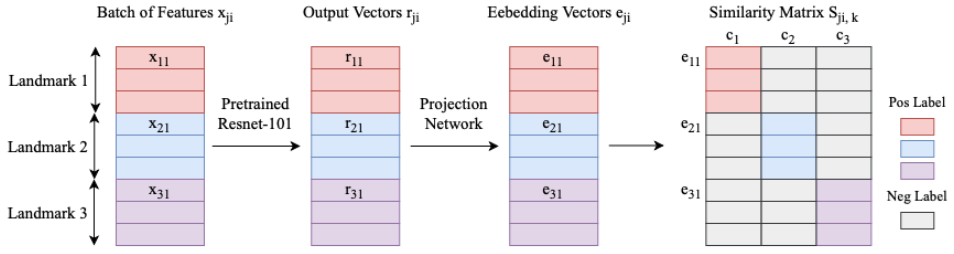


Figure 2: Training process of GE2E [11]

3.2 Additive Angular Margin Loss (ArcFace)

For the second retrieval algorithm, we use Additive Angular Margin Loss (ArcFace) [2], which is an approach originally leveraged to obtain distinguishable features in the face recognition task. Image features $\mathbf{e}_{ji}$ are firstly extracted by the encoder and projection network, and then we take the multiplication of normalized features and weights by using a fully connected layer to get the vector $\cos(\theta)$ (logit), which measures the similarity of weights and features in each dimension of

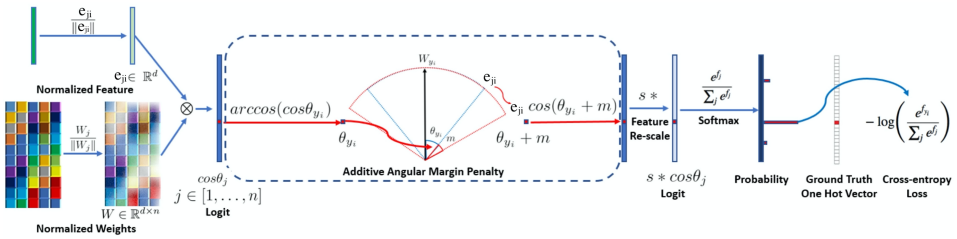


Figure 3: Training flow supervised by Additive Angular Margin Loss [2]

class. The key component of this algorithm is to add an Additive Angular Margin Penalty m to the similarity angle θ. The cosine value of the summation angle will be the new logit after re-scaling. The remaining procedures are the same as using the softmax loss.

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}} \tag{2}$$

The Additive Angular Margin Loss is defined as above. As for the geometric interpretation, since the features and weights are already normalized, the angle θ can directly control the arc length and the corresponding geodesic gap within the neighbouring classes on the hypersphere.

4 Experiments and Discussion

4.1 Data and Training

We perform our analysis of the two algorithms on GLDv2 [13], which is split into 3 sets of images: train, index, and test, each with 4132914, 761757 and 117577 images. Since GLDv2 was constructed in a noisy manner, we use a cleaned version of the train set [14], which contains 158047 landmark images with 81313 different classes. We adjust the size of all images by padding black and scaling them to 224×224. Some data augmentations are applied during training, including random flipping, brightness adjustment and Gaussian noise. We fetch $N \times M$ images as a batch into training, where $N = 48$ different landmark classes and $M = 5$ images per class. We chose 5 images per class to reduce the probability of having duplicates in a batch since around 30% of landmark classes have fewer than or equal to 5 images. Our encoder network ResNet-101 is pre-trained on ImageNet and the final embedding size is 512 for both representation learning algorithms. We train the networks using SGD with an initial learning rate of 0.01 and we decrease it by half after 25K steps.

We design a customized U-Net [9] as the baseline model, where the encoder is replaced by the same pre-trained ResNet-101 for a fair comparison. U-Net is a better variant of the traditional autoencoder [4], where the skip connections transfer some input features from encoder to decoder, adding more expressive power to the network. We train the U-Net with the reconstruction loss, trying to learn how to project images onto a latent space.

4.2 Mean Average Precision (mAP)

Mean average precision is the most common performance metric used in the context of object detection and information retrieval [13]. We use $\mathrm{mAP@100}$ to evaluate our models, where the average precision is calculated using the top-100 ranked images. This metric is defined as:

$$\mathrm{mAP@100} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{\min(m_q,100)}\sum_{k=1}^{\min(n_q,100)}\mathrm{P}_q(k)\mathrm{rel}_q(k) \tag{3}$$

Where Q represents the number of query images in the index set, m_q refers to the total number of ground truth labels, and n_q refers to the number of retrieved images when calculating the average precision(AP) for each query. The precision at rank k is the number of correctly retrieved images divided by k. The relevance function is an indicator function that outputs 1 if a retrieved image is relevant to the query and outputs 0 otherwise.

To evaluate $\mathrm{mAP@100}$, we first run our encoder network on full test and index set to extract image embeddings, and create a kNN ($k = 100$) lookup for each test embedding by using the cosine

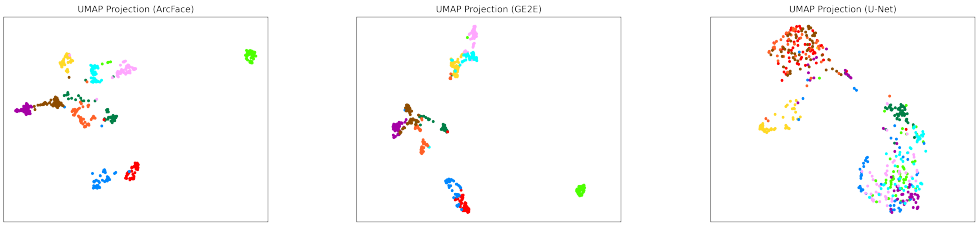
Index set size	12563	22051	41011	78962	154826	382409	761757 (full size)
U-Net (public)	0.0633 (100%)	0.0548	0.0490	0.0389	0.0306	0.0232	0.0173 (27.33%)
GE2E (public)	0.1738 (100%)	0.1567	0.1375	0.1208	0.0991	0.0772	0.0619 (35.62%)
ArcFace (public)	**0.3312 (100%)**	**0.3174**	**0.3023**	**0.2865**	**0.2656**	**0.2372**	**0.2142 (64.67%)**
U-Net (private)	0.0648 (100%)	0.0541	0.0438	0.0368	0.0302	0.0246	0.0197 (30.40%)
GE2E (private)	0.1820 (100%)	0.1684	0.1493	0.1303	0.1104	0.0864	0.0714 (39.23%)
ArcFace (private)	**0.3578 (100%)**	**0.3441**	**0.3252**	**0.3041**	**0.2819**	**0.2599**	**0.2332 (65.17%)**

Table 1: $\mathrm{mAP@100}$ on different size of index set. The "public" and "private" tags indicate the subset of the index images that are labelled in GLDv2. The percentage (%) represents performance degradation with increasing index set size.

distance between test and index embeddings. mAP score is calculated based on the lookups. We can see that ArcFace outperformed the other two models to a large extent in terms of the mAP score. Furthermore, we conduct analysis on the generalization and scalability of the three models. We find the performance degradation by calculating $\mathrm{mAP@100}$ on different index set sizes and dividing them by $\mathrm{mAP@100}$ at the smallest index set size. The generalization of ArcFace is superior to the other two as the score drops with the slowest rate against increasing index set size.

4.3 UMAP Embedding Projection Comparison

The final 512-dimensional embeddings obtained by the three models have been projected to 2D using the UMAP algorithm [6]. UMAP is a popular technique for dimension reduction and is better at preserving the global structure of the embeddings than t-SNE [10]. We sample 10 landmark classes, each with around 50 images to extract the embeddings.



(a) ArcFace (b) GE2E (c) U-Net

Figure 4: 2D UMAP projections with each color representing a landmark class

The embeddings of the baseline U-Net have a noisy embedding space where images from different classes can be projected to the same region, even though we see some distinctive regions (green and yellow). The GE2E embeddings, on the other hand, demonstrate a larger intra-class compactness since the nature of GE2E is to push images close to the centroids of its class and further from the centroids of other classes. Therefore, the gap between the embeddings within the same class and their corresponding class centers is much smaller. ArcFace achieves a better inter-class discrepancy as it introduces the additive angular margin, which makes the geometric distance between the neighbouring classes more evident. Different landmark classes are thus more distinguishable in the embedding space. Both GE2E and ArcFace show a higher level of feature discrimination, but overall ArcFace is more superior in terms of both intra-class compactness and inter-class discrepancy.

5 Summary

We have extended two representation learning algorithms, GE2E and ArcFace, which were originally used for speech verification and face recognition respectively, to solve the task of landmark image retrieval. The final results demonstrate the capability of cross-domain-utilization of these two algorithms, indicating that they can be transferred to the image retrieval task and obtain satisfying performance. Based on extensive experiments, we have found that ArcFace outperforms the GE2E model in image retrieval and has better potential for other representation learning tasks.

References

[1] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[2] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. arXiv:1801.07698 [cs.CV]

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs.CV]

[4] Geoffrey E Hinton and Richard Zemel. 1994. Autoencoders, Minimum Description Length and Helmholtz Free Energy. In *Advances in Neural Information Processing Systems*, J. Cowan, G. Tesauro, and J. Alspector (Eds.), Vol. 6. Morgan-Kaufmann. https://proceedings.neurips.cc/paper/1993/file/9e3cfc48eccf81a0d57663e129aef3cb-Paper.pdf

[5] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. 2018. SphereFace: Deep Hypersphere Embedding for Face Recognition. arXiv:1704.08063 [cs.CV]

[6] Leland McInnes, John Healy, and James Melville. 2020. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426 [stat.ML]

[7] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. 2014. Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 1717–1724. https://doi.org/10.1109/CVPR.2014.222

[8] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2007. Object retrieval with large vocabularies and fast spatial matching. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2007.383172

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.04597 [cs.CV]

[10] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605. http://www.jmlr.org/papers/v9/vandermaaten08a.html

[11] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. 2020. Generalized End-to-End Loss for Speaker Verification. arXiv:1710.10467 [eess.AS]

[12] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. 2018. CosFace: Large Margin Cosine Loss for Deep Face Recognition. arXiv:1801.09414 [cs.CV]

[13] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. 2020. Google Landmarks Dataset v2 – A Large-Scale Benchmark for Instance-Level Recognition and Retrieval. arXiv:2004.01804 [cs.CV]

[14] Shuhei Yokoo, Kohei Ozaki, Edgar Simo-Serra, and Satoshi Iizuka. 2020. Two-stage Discriminative Re-ranking for Large-scale Landmark Retrieval. arXiv:2003.11211 [cs.CV]

Appendix A. Individual Contribution

Overall, our team has an excellent collaboration together. Each team member made equal contributions to this work. Yu Wang is responsible for setting up the training platform on AWS lambda, and building the model training pipeline. He also looked into and implemented the GE2E model. Yitong Wang mainly worked on data preprocessing, including data cleaning and designing our customized data loader for GLDv2. She performed the data compatibility verification between different models. She also researched for our performance metric mAP and evaluated our models against the metric. Dongfang Cui collaborated with Yu Wang on setting up the training platform. He is responsible for the data preparation and migration between the platforms. He also researched the re-implemented the ArcFace model. All team members contributed to background research and literature reviews.

Appendix B. Project Github Repository

Our project is located at: `https://github.com/rwang97/landmark-retrieval`.