Large Scale Landmark Recognition via Deep Learning

Jifu Zhao

Department of Nuclear, Plasma, and Radiological Engineering
University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA

Abstract

Landmark recognition is one kind of object recognition problem that hasn't been well solved. Due to large number of landmarks and highly imbalanced dataset, the classical methods used for object recognition cannot be directly applied. This paper presents the application of triplet network for large scale landmark recognition. Through fine-tuning pretrained convolutional network network (CNN) and minimizing triplet loss, the triplet network can learn appropriate metric such that most similar images can be retrieved through k-nearest neighbor (KNN) algorithms. The performance of the proposed method is evaluated on real-world landmark recognition dataset.

Keywords: Landmark recognition, Triplet network, Deep learning, Metric learning

1. Introduction

Over the recent years, with the rapid development of deep learning techniques, especially convolutional neural networks (CNN), people have achieved huge progress in computer vision. Different kinds of network architectures have been proposed. Some widely used architectures include VGG [1], ResNet [2], Inception Network [3] and so on. On the famous ImageNet Large Scale Visual Recognition Challenge (ILSVRC), these algorithms have achieved higher accuracy than human beings.

CNNs have been proven to be one of the best solutions for computer vision tasks, which have been widely used in different areas, such as autonomous cars and automatic face recognition. The fast development of deep learning and computer vision techniques have changed people's lifestyle. However, classical object recognition tasks generally require large amount training images. For example, CIFAR-10 dataset [4] contains 60,000 training images in 10 different classes. ImageNet dataset [5] contains 14,191,122 images in 1,000 classes. Large amount of training images is one of the key factors to guarantee the success of CNNs. In addition, object recognition system usually use fully-connected layer as the structure for the final one or several layers, which is reasonable for the problem with only a few output classes.

However, there exists a special kind of problem called one-shot learning [6] that cannot be easily solved with the classical methods. One-shot learning aims to learn information from one, or only a few, training images. And there could be a huge amount of classes. These two factors require new solutions. There are several famous one-shot learning examples, such as face recognition/verification and famous street-to-shop [7] systems. Over the past several years, a series of solutions have been proposed to solve one-shot learning problems. Siamese network [8] has been proposed for different problems including authorship and image recognition. Face recognition or face verification have been well studied [9, 10]. Triplet-based networks have been studied for audio and image retrieval problems [11, 12, 13, 14, 15, 16]

This paper focuses on landmark recognition from images. There are thousands of landmarks exist. Some are very popular and some are less popular. People around the world take countless photos that contain

Email address: jzhao59@illinois.edu (Jifu Zhao)
URL: All code is available in GitHub: https://github.com/JifuZhao/Landmark-Recognition (Jifu Zhao)

different landmarks, which provide enough training images. Due to highly imbalanced dataset and large amount of different landmarks, this paper focuses on one-shot learning for landmark recognition. More specifically, with some reference images that contain different landmarks, given a new image, this paper wants to design an algorithm that can automatically detect the landmarks the new image contains. Fig. 1 shows an illustration of the landmark recognition problem.

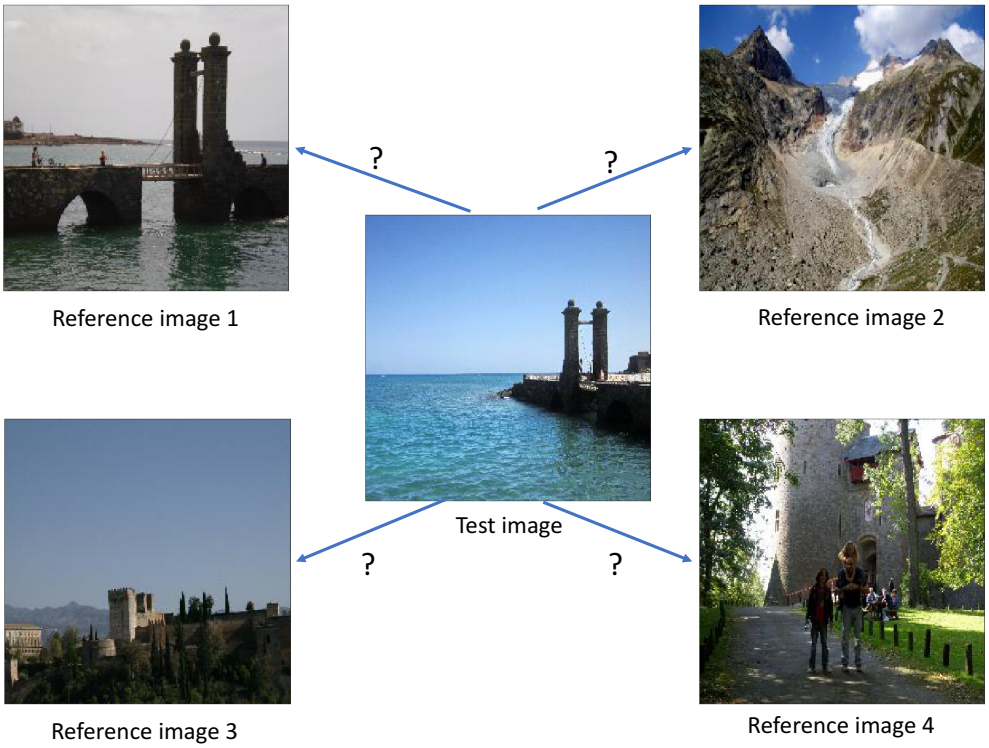

Figure 1: Illustration of landmark recognition problem

2. Data

For a successful landmark recognition system, training data are very important. Large amount of training images that contain as many different landmarks as possible can help build more robust systems. The data used in this work come from Google-Landmarks dataset, which is released by Google on early March 2018. The whole Goolge-Landmarks dataset contains human-made and natural landmarks across the world. There are 1,225,029 training images in 14,951 different landmarks. A subset of 12 randomly sampled images are shown in Fig. 2.

As shown in Fig. 2, the training images are pretty noisy. The landmarks may not be at the center of the whole image and some landmarks might be blocked by humans. All these factors make it quite challenging. Fig. 3 shows the count distribution of different landmarks sorted by the number of images for each landmark. It is clear that for some popular landmarks, such as Eiffel Tower and Mount Fuji, there are large amount training images available. For example the most popular landmarks contains more than 5,000 images. However, for less popular landmarks, the available images are few, and some landmarks only have one image.

Limited by the computation resources, in this work, only a subset of training images are used. More specifically, there are at most 10 training images kept for each landmark. Based on this criterion, 113,783 training images in 14,943 different landmarks are chosen as the training set. The count distribution is shown

Figure 2: Subset of sample images

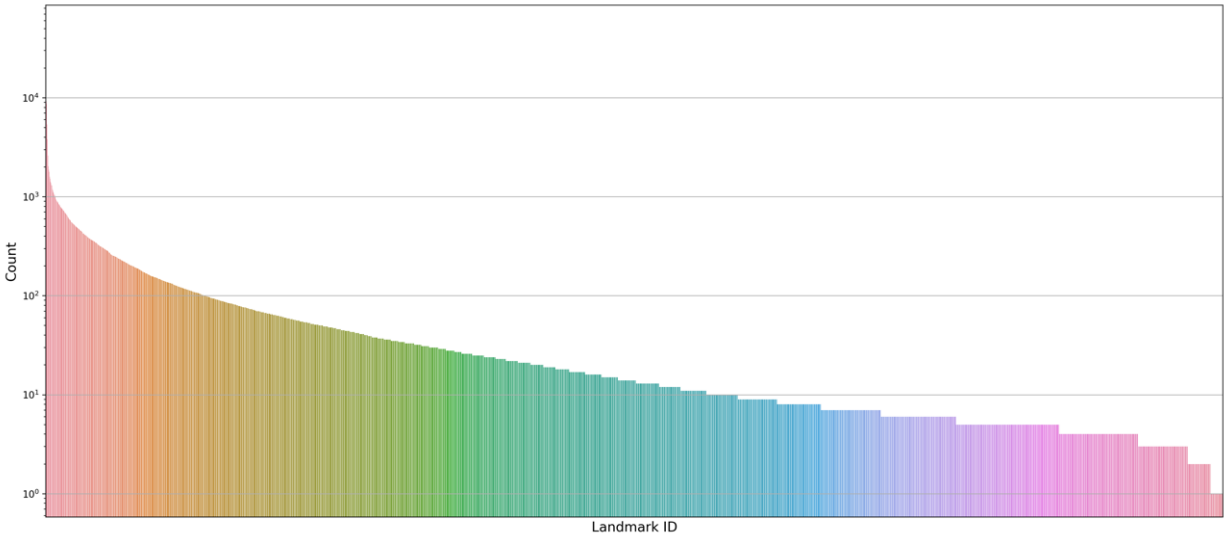

Figure 3: Count distribution of Goolge-Landmark dataset

in Fig. 4. In addition, 22,255 images with 7,675 different landmarks are chosen as the validation set and 22,391 images with 14,436 different landmarks are chosen as the test set.

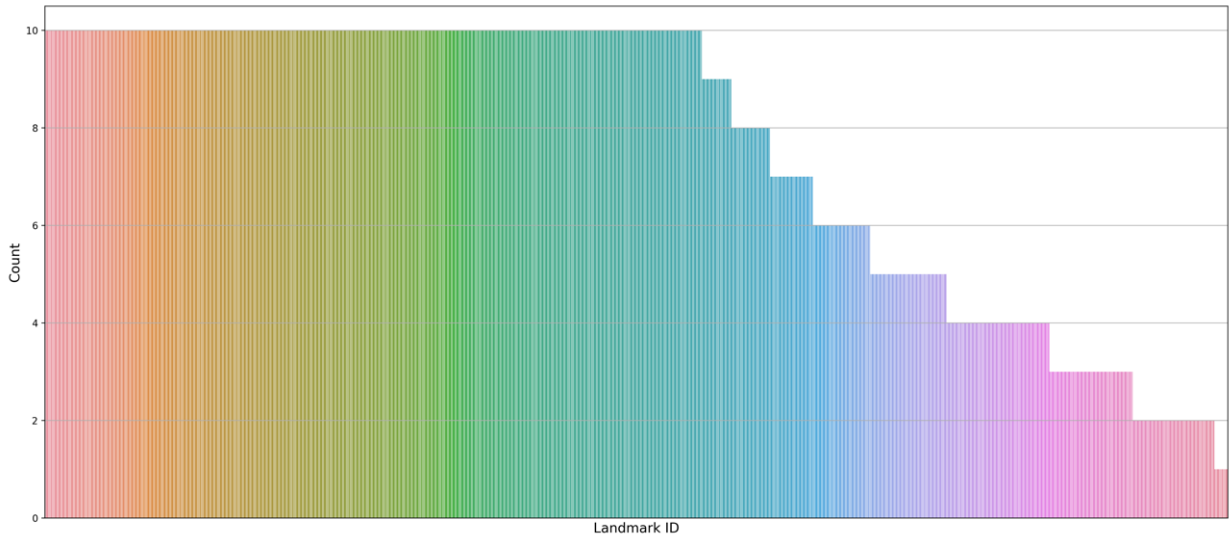

Figure 4: Count distribution of training images

3. Methodology

As discussed in Section 1, with 14,943 different landmarks, classical multi-class classification networks which use fully-connected layer as the output layer is not appropriate. To solve this problem, this work focuses on metric learning. Through appropriate algorithms, the original images can be represented in a new space such that similar images will be close to each other while different images will have larger distance. Fig. 5 clearly shows this process.

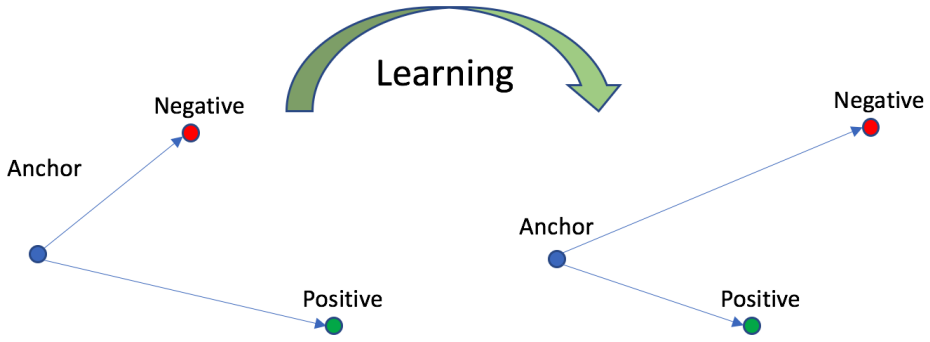

Figure 5: Illustration of metric learning

Given an image (Anchor image) to determine the landmark it contains, there are two extra images used, Positive image and Negative image. Positive image contains the same landmark with the Anchor image while Negative image contains different landmarks. When the system doesn't work correctly, the distance between the Anchor image and the Positive image would be larger than the distance between the Anchor

image and the Negative image. Through metric learning, the system should be able to learn a metric such that the distance between the Anchor image and the Positive image is smaller than the distance between the Anchor image and the Negative image. In this way, the landmark of the Anchor image can be predicted using the landmark from the closest images.

The idea in Fig. 5 leads to so-called triplet loss [9]. Suppose the Anchor image, Positive image, and Negative image are denoted by a, p, and n and the system defines some representation function $f(\cdot)$. Then the Euclidean distance between a and p is $||f(a)-f(p)||_2^2$ and the distance between a and n is $||f(a)-f(n)||_2^2$. Through training, the system should be able to distinguish positive and negative images such that:

$$||f(a) - f(p)||_2^2 <= ||f(a) - f(n)||_2^2 \tag{1}$$

To make the system more robust, some margin α is added such that:

$$||f(a) - f(p)||_2^2 + \alpha <= ||f(a) - f(n)||_2^2 \tag{2}$$

From above equations, triplet loss $L(a, p, n)$ is defined as:

$$L(a, p, n) = max\{||f(a) - f(p)||_2^2 + \alpha - ||f(a) - f(n)||_2^2, 0\} \tag{3}$$

The goal of the training is to minimize the triplet loss defined in Eq. 3. Based on above description, the final system is described in Fig. 7.

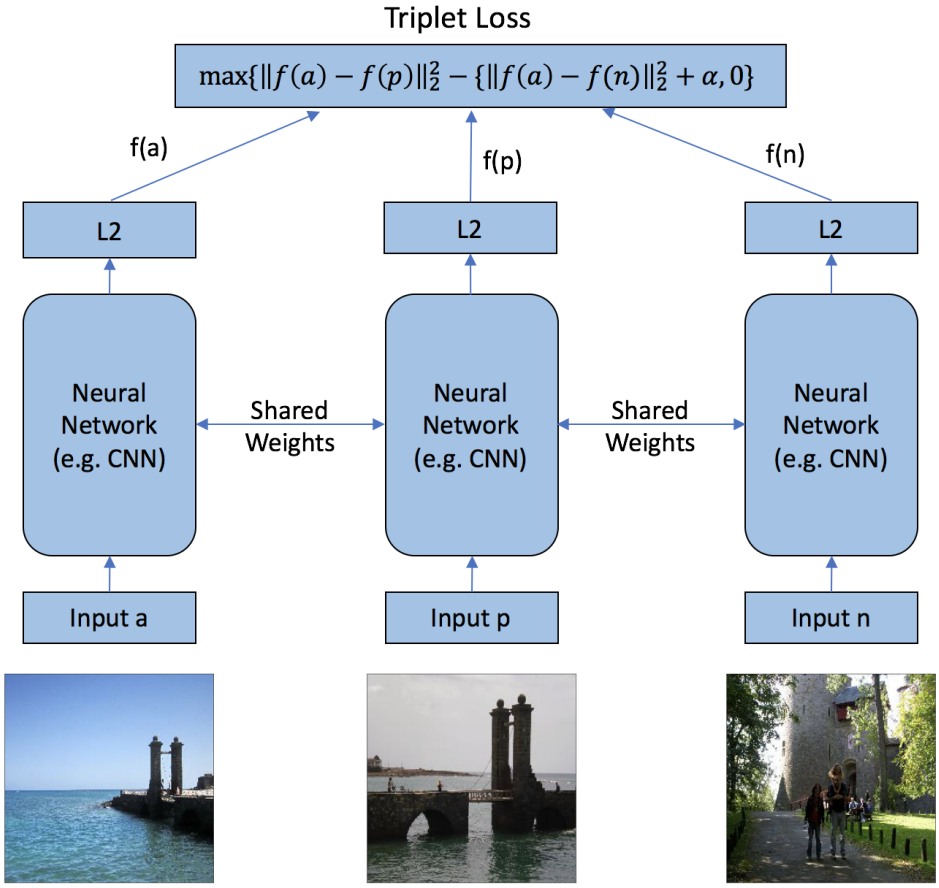

Figure 6: Triplet network overall architecture

In Fig. 6, there are three images a, p, and n as the input images. The input images are first processed by the same base-network. For computer vision tasks, CNN is generally used as the base-network. To define appropriate distance functions, the output from the base-network is normalized using L2 normalization. Finally, the triplet loss is computed using Eq. 3. The whole network is optimized using backpropagation. The whole system is implemented using Keras with TensorFlow as backend.

As shown in Fig. 6, the performance of the whole system is determined by the base-network. Building a CNN from scratch is suggested when there are enough training images and advanced hardware. However, in this paper, limited by computation resources, building base-network from scratch is not appropriate. In addition to building CNN from scratch, pretrained CNNs can also be utilized. The simplest way is to directly apply the pretrained CNNs on landmark recognition problem without training. Since the base-network is not optimized for landmark recognition, the performance of the system is limited. Another choice is to fine-tune the pretrained CNNs. For pretrained CNNs, the lower convolution layers are usually assumed to encode more generic and reusable features, while the higher convolution layers encode specialized features [17]. Based on this idea, during training process, the weights for lower layers will be freezed and only the weights for higher layers are trained. Through backpropagation, the pretrained CNNs can be optimized for landmark recognition.

Figure 6 depicts the training process for base-network. With the base-network, the next step is to determine the landmark for unseen images. Given a large number of training images that contain different landmarks, for the unseen images, this turns out to be finding the most similar images such that the landmark for unseen images are inferred by the most similar images, which can be solved using k-Nearest Neighbor (KNN) algorithm. The whole system is shown in Fig. 7.

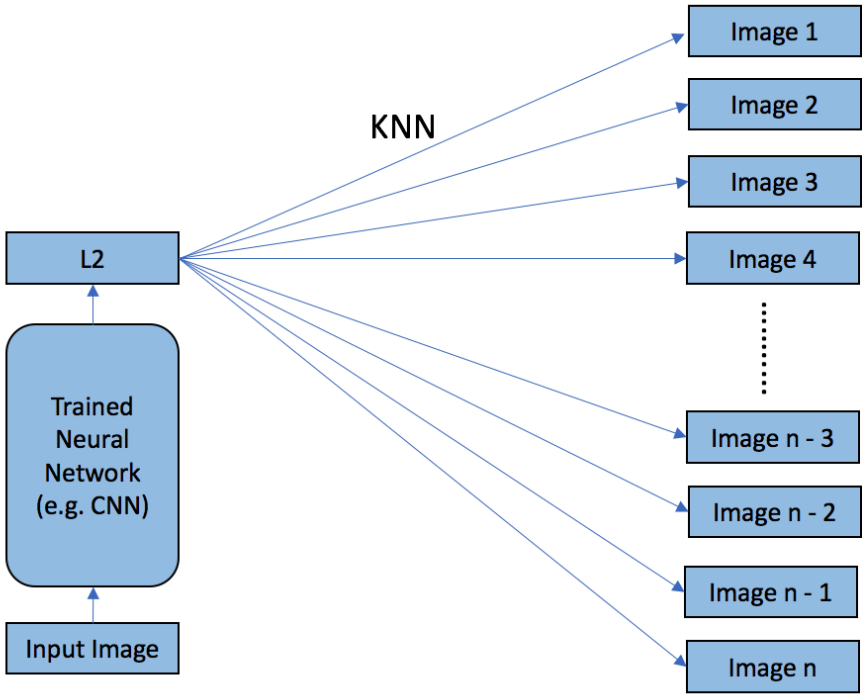

Figure 7: Overall landmark recognition system architecture

In Fig. 7, given a new input image, the pretrained neural network will compute its representation in a new space. Through KNN algorithm, the system will find the most similar images in the reference image pool and make prediction based on the retrieved images.

4. Results

As discussed in Section 3, this paper implements two types of pretrained CNNs: naive implementation and fine-tuning. More specifically, pretrained VGG16 and InceptionV3 models on ImageNet dataset are implemented. Fig. 8 shows the performance of different algorithms. VGG16 only and InceptionV3 only models implement pretrained VGG16 and InceptionV3 without further training, while VGG16 triplet and InceptionV3 triplet models are the models with fine-tuning. As a comparison, this paper also shows the result of random guess. It is clear that, as the number of retrieved images increases, the accuracy also increases, which is expected. The best top-1 accuracy is from triplet InceptionV3 model, which gives 47% top-1 accuracy.

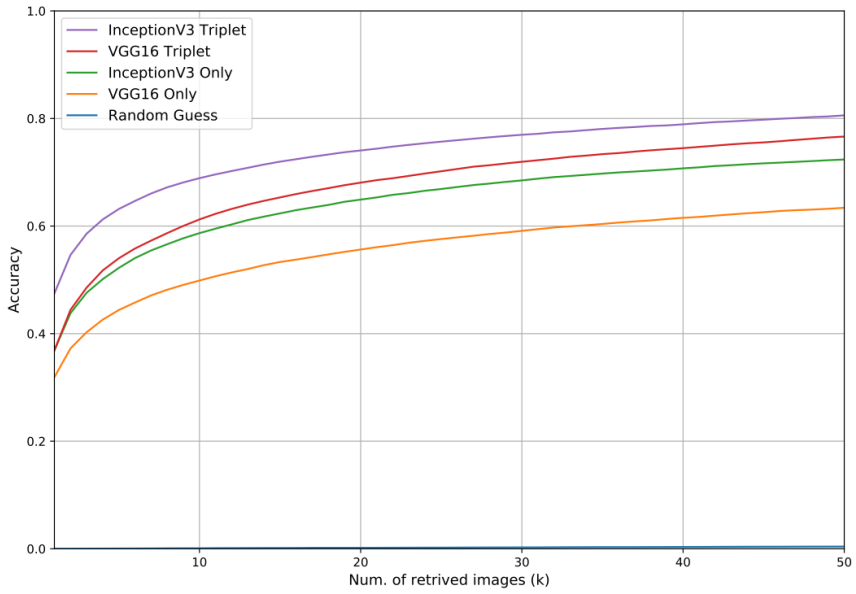

Figure 8: Top-k image retrieval accuracy for different numbers of retrieved images

To show the performance of the model, Fig. 9 shows the top-6 retrieved images for randomly sampled 5 images with InceptionV3 triplet model. From Fig. 9, for some images, the built system can accurately predict the landmarks the unseen images contain. However, for some unseen images, the system cannot make accurate prediction based on most similar image.

5. Conclusion

This paper studies one specific one-shot learning problem, landmark recognition. Considering the object recognition as metric learning problem, this paper implements triplet network and fine-tunes pretrained CNN models to extract useful features from the original images. With KNN algorithm, this paper achieved 47% top-1 accuracy on separate test dataset.

However, there are further procedures exist to improve the performance of the system. As discussed in Section 2, the input images are pretty noisy. Through image processing, for example, adjusting the image such that the landmarks are at the center of the images and irrelevant objects are removed, the performance of the system might be further improved. In addition, due to computation resource limitation, currently, only less than 10% of the total training images are actually used and the structure of the base-network hasn't been explicitly explored. All these factors might influence the performance of the system. The future work will focus on processing the input images and exploring better base-networks.

Figure 9: Top-6 image retrieval results from sample query images

6. References

[1] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[4] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[6] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

[7] Si Liu, Zheng Song, Guangcan Liu, Changsheng Xu, Hanqing Lu, and Shuicheng Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3330–3337. IEEE, 2012.

[8] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

[9] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[10] Jingtuo Liu, Yafeng Deng, Tao Bai, Zhengping Wei, and Chang Huang. Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv preprint arXiv:1506.07310*, 2015.

[11] Jiang Wang, Thomas Leung, Chuck Rosenberg, Jinbin Wang, James Philbin, Bo Chen, Ying Wu, et al. Learning fine-grained image similarity with deep ranking. *arXiv preprint arXiv:1404.4661*, 2014.

[12] Kevin Lin, Huei-Fang Yang, Kuan-Hsien Liu, Jen-Hao Hsiao, and Chu-Song Chen. Rapid clothing retrieval via deep learning of binary codes and hierarchical search. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 499–502. ACM, 2015.

[13] M Hadi Kiapour, Xufeng Han, Svetlana Lazebnik, Alexander C Berg, and Tamara L Berg. Where to buy it: Matching street clothing photos in online shops. In *ICCV*, pages 3343–3351, 2015.

[14] Xi Wang, Zhenfeng Sun, Wenqiang Zhang, Yu Zhou, and Yu-Gang Jiang. Matching user photos to online products with robust deep features. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 7–14. ACM, 2016.

[15] Devashish Shankar, Sujay Narumanchi, HA Ananya, Pramod Kompalli, and Krishnendu Chaudhury. Deep learning based large scale visual recommendation and search for e-commerce. *arXiv preprint arXiv:1703.02344*, 2017.

[16] Xiaoyu Qi, Deshun Yang, and Xiaoou Chen. Audio feature learning with triplet-based embedding network. In *AAAI*, pages 4979–4980, 2017.

[17] Francois Chollet. *Deep learning with Python*. Manning Publications Co., 2017.