6th Place Solution to Google Universal Image Embedding

Socratis Gkelios
Department of Electrical
and Computer Engineering
Democritus University of Thrace, Greece
sgkelios@ee.duth.gr

Anestis Kastellos and Savvas A. Chatzichristofis
Intelligent Systems Laboratory
Department of Computer Science
Neapolis University Pafos, Cyprus
{kastellosa},{s.chatzichristofis}@nup.ac.cy

Abstract

This paper presents the 6th place solution to the Google Universal Image Embedding competition on Kaggle. Our approach is based on the CLIP architecture; a powerful pre-trained model used to learn visual representations from natural language supervision. We also utilized the SubCenter ArcFace loss with dynamic margins to improve the distinctive power of class separability and embeddings. Finally, a diverse dataset has been created based on the test's set categories and the leaderboard's feedback. By carefully crafting a training scheme to enhance transfer learning, our submission scored 0.685 on the private leaderboard.

1. Introduction

Even though image retrieval is a well-studied computer vision task, many challenges still need to be addressed to approach humans' perception of visual similarity [5,6]. For the first time, Google launched an image retrieval competition, the Google Universal Image Embedding, that focuses on the techniques' universality and generalization ability.

In this context, the competition has three main challenges:

- The training dataset is not provided, so the contenders were responsible for assembling their dataset to train their models.
- The categories of the test distribution significantly differ, and, considering this is an image retrieval task on the instance-level, the model should be able to handle intra-class similarity on many different categories that do not contain similar characteristics.
- The upper limit of embedding's size is 64.

We explored instance recognition/retrieval datasets close to the test's distribution categories to tackle the first challenge. We emphasized powerful pre-trained neural network models trained on large-scale datasets for the second challenge. For the final challenge, we utilized Principal Component Analysis (PCA) so as to reduce the dimensionality of the embedding, retaining as much information as possible. Our final solution is based on the CLIP architecture [12] trained on the LAION-2B dataset, a subset of LAION-5B [13]. Also, we utilized the ArcFace [3] loss function, a popular loss function on Kaggle competitions that maximizes the intra-class separability. Specifically, we adopted the Sub-Center ArcFace [4], which is a variant of the original ArcFace, while also using dynamic margins [7]

2. Method

2.1. Model Architecture

Our approach employed the CLIP architecture proposed for the image-to-text task. Two public implementations leveraged different datasets for the training procedure: OpenAI's CLIP [12] and OpenCLIP [16]. In the first case, the model is pre-trained on Imagenet22K, and the best-performing model is ViT-L, while in the second case, the training was performed on the LAION-2B dataset, with the best-performing model being ViT-H. We used only the image encoder part of the original topology in both cases. The results strongly suggested that the OpenCLIP model was significantly better than openai's, mainly because ViT-H has more learning capacity than ViT-L. So we proceeded with ViT-H CLIP. We also modified the head by taking the output of the projection layer and feeding it to a BN-Dropout-FC block. The dropout rate is set to 0.2, and the FC downsizes the 1024-dimensional embedding to 256. The 256 vector passes through the ArcFace layer to obtain the classification logits.

To handle the instance retrieval task, we used the Sub-Center ArcFace. This variant of the original ArcFace improves robustness when noisy training data. This is achieved by relaxing the constraint that every class has a single center by introducing more sub-centers per class. Models trained in this manner can showcase better representation capability and produce high-quality embeddings.

Although our training dataset does not contain extreme imbalance, some classes are underrepresented ($\approx$5 images per class) compared to others ($>$ 40 images per class). So dynamic margins are used to alleviate this problem. In essence, the margins are allocated dynamically per class depending on the number of instances representing them. We facilitate the training procedure by assigning bigger margins on underrepresented classes to make separating them easier. We defined the upper bound to 0.45 and the lower bound to 0.005.

2.2. Datasets

We explored a plethora of datasets closely related to the test's distribution. Specifically, we used subsets of the following datasets for our best score: Google LandmarksV2 [15], Products10k [1], Food-101 [2], iMaterialist, Fashion200k [8], DeepFashion [10], RP2K [11], Stanford Cars [9], Stanford Online Products [14], MET Artwork dataset [17] and Storefronts-146[1]. Although iMaterialist does not contain instance labels, we created manually, with the help of the pre-trained CLIP, about 400 additional labeled furniture images. We tried to approximate the percentage of each category on the test set for the most important categories. In our initial experiments, we had around 200k images in our training set but noticed that the dataset size and the public leaderboard score followed a similar trend. As a result, we assembled a dataset of $\approx$655k images from the categories mentioned above. In most datasets, we selected classes randomly with at least three samples.

3. Experiments

3.1. Training details

The OpenCLIP presents state-of-the-art performance on zero-shot tasks; therefore, the model offers high-quality image descriptors out-of-the-box. The challenge was to train this model without altering its learned weights and causing a significant distribution shift. We froze the CLIP backbone during our first training attempts and trained the head for five epochs with a 0.0001 learning rate (lr). After this step, we unfroze the ViT-H until resblock 15 and then trained for another epoch with the lr reduced by a factor of 10. As we started adding more data to our pipeline, we noticed, we could train the backbone even more, and for this reason, we adopted the following scheme that applies different lr to the backbone and the head,

$$LearningRate = \begin{cases} 1e^{-7}, & for\ model.backbone \\ 1e^{-4}, & for\ model.head \end{cases}$$

and trained the model for four epochs. Again we only unfroze the model from resblock31 to resblock 15 for three main reasons:

- We did not observe improvement by unfreezing more blocks
- Computational complexity and training time
- Initial layers usually contain low-level generic features, so we did not want to alter the learned features from a bigger dataset

Remarkably, this scheme performed worse when the smaller dataset was used. Therefore, we set the batch size to 32 and employed the Adam optimizer.

We used the following augmentations during training inspired by past Kaggle instance level competitions: Horizontal flip, image compression, shift, scale, rotate, cutout, random brightness, contrast, and RGB-shift.

3.2. Validation strategy

We attempted to create a validation set with classes that were not seen in training, with and without matching the distribution of the test set. After extracting the image descriptors (with pooling), we trained a NearestNeighbor algorithm that generated a ranking list of the closest matches. We evaluated the ranking list using the mAP metric. Unfortunately, none of our validation schemes correlated adequately with the public leaderboard, so technically, our only validation set was the public leaderboard. We used the local validation only to estimate if there was a problem with the new approaches. Also, we measure the per-vertical mAP for each category to see if we can infer any additional insights on which category our model seems to struggle with.

3.3. Inference

During inference, the images were resized to 224×224 to match the model's expected input. We also normalized the images according to our training pipeline. Finally, we tested two approaches for reducing the dimensionality of the final vector to 64 as the model output a 256-dimensional vector. The first was simply using average pooling, while in the second, the features were transformed using PCA. We fitted PCA on a public dataset shared on Kaggle[2] comprised of 130k images from all categories of the test set. The performance with PCA was significantly better than the average pooling by 0.016 on the public leaderboard and 0.005 on the private set. The final solution was comprised of a single model without

4. Conclusions

This paper presented our approach to the Google Universal Embedding Challenge. We showed that by incorporating different lr to the backbone and the head, the performance

[1] https://www.kaggle.com/datasets/kerrit/storefront-146

[2] https://www.kaggle.com/datasets/rhtsingh/130k-images-512x512-universal-image-embeddings

of the pre-trained CLIP model could be improved significantly. Additionally, PCA can be utilized effectively to reduce the dimensionality of the final vector by providing a robust embedding.

References

[1] Yalong Bai, Yuxiang Chen, Wei Yu, Linfang Wang, and Wei Zhang. Products-10k: A large-scale product recognition dataset. *arXiv preprint arXiv:2008.10545*, 2020. 2

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 2

[3] Jiankang Deng, Jia Guo, Tongliang Liu, Mingming Gong, and Stefanos Zafeiriou. Sub-center arcface: Boosting face recognition by large-scale noisy web faces. In *European Conference on Computer Vision*, pages 741–757. Springer, 2020. 1

[4] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 1

[5] Socratis Gkelios, Yiannis S. Boutalis, and Savvas A. Chatzichristofis. Investigating the vision transformer model for image retrieval tasks. In *17th International Conference on Distributed Computing in Sensor Systems, DCOSS 2021, Pafos, Cyprus, July 14-16, 2021*, pages 367–373. IEEE, 2021. 1

[6] Socratis Gkelios, Aphrodite Sophokleous, Spiros Plakias, Yiannis Boutalis, and Savvas A. Chatzichristofis. Deep convolutional features for image retrieval. *Expert Systems with Applications*, 177:114940, 2021. 1

[7] Qishen Ha, Bo Liu, Fuxu Liu, and Peiyuan Liao. Google landmark recognition 2020 competition third place solution. *arXiv preprint arXiv:2010.05350*, 2020. 1

[8] Xintong Han, Zuxuan Wu, Phoenix X Huang, Xiao Zhang, Menglong Zhu, Yuan Li, Yang Zhao, and Larry S Davis. Automatic spatially-aware fashion concept discovery. In *Proceedings of the IEEE international conference on computer vision*, pages 1463–1471, 2017. 2

[9] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 2

[10] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[11] Jingtian Peng, Chang Xiao, and Yifan Li. Rp2k: A large-scale retail product dataset for fine-grained image classification. *arXiv preprint arXiv:2006.12634*, 2020. 2

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1

[13] Christoph Schuhmann, Romain Beaumont, Cade W Gordon, Ross Wightman, Theo Coombes, Aarush Katta, Clayton Mullis, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 1

[14] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[15] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2575–2584, 2020. 2

[16] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971, 2022. 1

[17] Nikolaos-Antonios Ypsilantis, Noa Garcia, Guangxing Han, Sarah Ibrahimi, Nanne Van Noord, and Giorgos Tolias. The met dataset: Instance-level recognition for artworks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 2