Google Universal Image Embedding Challenge

EEE 511, Fall 2022, Team #9, Progress Check 2

Sumant Kulkarni, Dhanraj Bhosale, Pratyush Pandey, Shayal Shamsu

Abstract

Google Universal image embedding challenge is a competition hosted by Kaggle in collaboration with Google research and Google lens. The competitors have to develop a Machine Learning model which can retrieve relevant database images for a given query image. The specific challenge is to build a single universal image embedding model capable of representing objects from multiple domains at instance level. Image datasets comprise of a variety of object types like Apparel, Artwork, Landmarks, Furniture and Packaged goods. We have used one of the ML model submitted by a Top 10 competitor as reference and have replicated the same. It makes use of a CLIP backbone model and Arcface for similarity learning. The model trained and submitted (late submission) by us for the competition scored 0.678 which is at par with the Top 10th score amongst all competitors. It is trained to classify 17691 different object types.

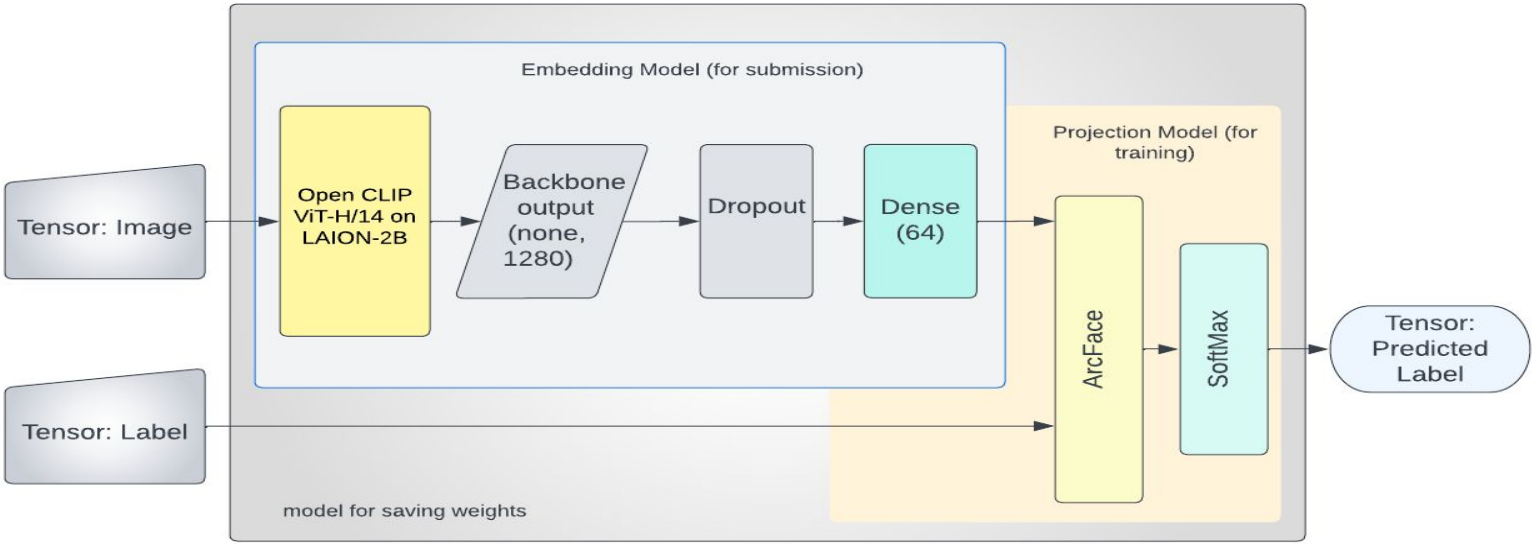
Introduction

- Image representations are a critical building block of computer vision applications. Traditionally, research on image embedding learning has been conducted with a focus on per-domain models.
- Generally, papers propose generic embedding learning techniques which are applied to different domains separately, rather than developing generic embedding models which could be applied to all domains combined.
- Instance-Level Recognition (ILR) is tackled by training a deep learning model with a large set of images. Capturing features of all object domains in a single dataset and training a model that can distinguish between them is a challenging task.
- The competition is structured in a representation learning format. Competitors have to create a model that extracts feature embedding for the images and submit the model via Kaggle Notebooks. Kaggle runs the model on a held-out test set, perform a k-nearest-neighbors lookup, and score the resulting embedding quality.
- Google research believes that this multi domain ILR is the key to real-world visual search applications, such as augmenting cultural exhibits in a museum, organizing photo collections, visual commerce and more.

Datasets

- Imagenet - https://www.image-net.org/index.php (Available for free to researchers for noncommercial use)
- Products -10K - https://products-10k.github.io/ (Available for free for non-commercial research and educational purposes)
- Google Landmark Recognition 2021 - https://www.kaggle.com/competitions/landmarkrecognition-2021/data (Dataset is part of Kaggle competition in 2021)
- To reduce the dataset size, this dataset has only 50 images per class
- 90% data is used for training while 10% is used for validation.

Model Architecture



Embedded model for Inference (Model Submitted for scoring):
Backbone(CLIP) + Dropout + Dense(units=64) + L2 Norm

Training: Backbone(CLIP) + Dropout + Dense(units=64) + ArcFace + Softmax (classes=17691)

ML Model Architecture

- Embedded Inference Model:
  - The CLIP(Contrastive Language-Image Pre-Training) is a neural network developed by OpenAi and is trained on a variety of **image-text** pairs.
  - OpenCLIP trained on ViT-H/14 on LAION-2B has the highest accuracy of 78%
  - Due to ‘Zero-Shot’ capabilities, CLIP models can be applied to nearly arbitrary visual classification tasks which fits perfectly in our use case.
  - Dropouts are used to prevent overfitting
  - L2 Norm Regularization is used to penalize insignificant parameters by suppressing their weights
- Training Model:
  - We replace L2Norm by ArcFace during training in front of the above embedded model for classification. The weights learned by Arcface are not overly dependent on the input dataset and it helps in reducing the training epochs.
  - In ArcFace, the angular margin ($\cos\theta$) is calculated by normalizing features and FC (Dense) layer weights and taking the inner product. The loss is calculated by applying Softmax to $\cos\theta$. Then arccos is applied to the $\cos\theta$ values and an angular margin of $+m$ is added only for the correct labels.

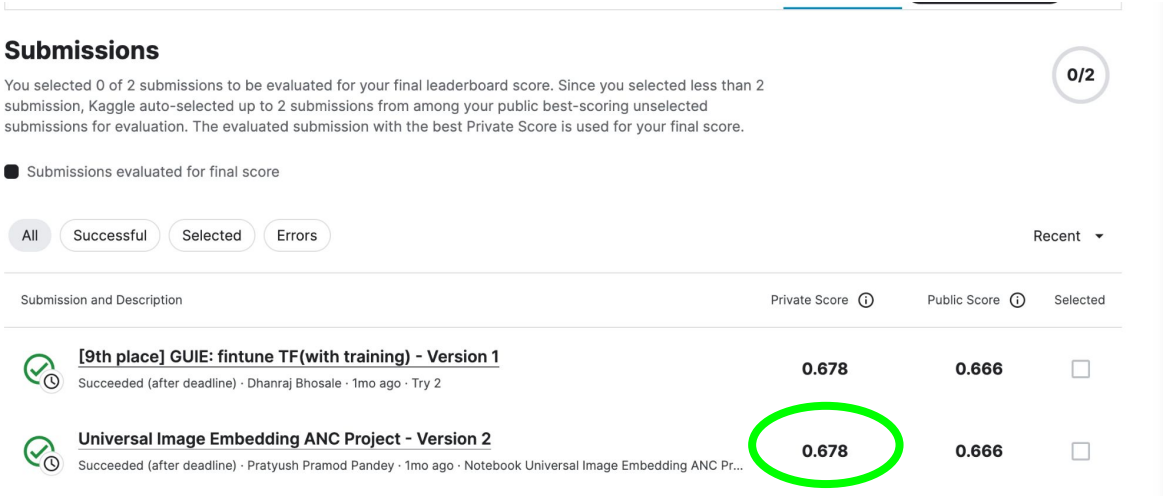
Implementation

- We used Kaggle Notebook for training the model. All required computing resources are made available by Kaggle in the notebook.
- Access to datasets is also easily available in the kaggle environment.
- We built the model using Tensorflow and TPU. It takes about 3 hours to train the model in the environment, for 10 epochs.
- The embedded model with its weights is submitted in zipped format to kaggle for scoring
- We are simulating the performance of this trained model on our laptop by predicting embeddings of test images from various object types.

Results

The model trained and submitted by us for the competition scored 0.678 which is at par with the top 10 score amongst all competitors. It is trained to classify 17691 different object types.

[GitHub Repo Link]



Simulation

- In addition to the score obtained from the Kaggle competition, we are developing a simulation to demonstrate the model performance during our presentation
- We will be using 2 images each from 3 different object type datasets for the simulation
- We will run the model prediction on these test images and obtain corresponding image embeddings.
- We will then compare the embeddings of images and demonstrate that similar images have similar embeddings,
- And images from different object types have dissimilar embeddings

References

- Google AI Blog - Introducing the Google Universal Image Embedding Challenge, August 4, 2022, Posted by Bingyi Cao and Mário Lipovský, Software Engineer, Google Lens, https://ai.googleblog.com/2022/08/introducing-google-universal-image.html
- Baseline model implementation for the Kaggle universal image embedding - https://github.com/google-research/googleresearch/tree/master/universal_embedding_challenge
- Training data-efficient image transformers & distillation through attention - https://arxiv.org/pdf/2012.12877.pdf
- Transformers for image recognition at scale - https://arxiv.org/pdf/2010.11929.pdf
- Refernce Code notebook for implementation - https://www.kaggle.com/code/akihirok/9th-place-guie-fintune-tf-clip-with-training
- https://huggingface.co/laion/CLIP-ViT-H-14-laion2B-s32B-b79K

Plan for Remaining Activities

- ❏ Develop the simulation to demonstrate the model – Sumant and Shayal - by Nov-15
- ❏ Develop the Project presentation – All team members – Nov-11 to Nov-15
- ❏ Document the First draft Project report – All team members – Nov-1 to Nov-20
- ❏ Group Review of the Project report – All team members – Nov-16 to Nov-28
- ❏ Document and submit the Final Project report – All team members – Nov-28 to Dec-9

Thank You