Google Universal Image Embedding Challenge

EEE 511, Fall 2022, Team 9

Sumant Kulkarni, Dhanraj Bhosale,
Pratyush Pandey, Shayal Shamsu

The Challenge

Google Universal image embedding challenge is a competition hosted by Kaggle in collaboration with Google research and Google lens.

The specific challenge is to build a single universal image embedding model capable of representing objects from multiple domains at instance level.

The task is to not only determine the generic category of an object (e.g., an arch), but also the specific instance of the object (”Arc de Triomphe de l'Étoile, Paris, France”)

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.

Introduction

- Traditionally, research on image embedding learning has been conducted with a focus on per-domain models.
- Generally, papers propose generic embedding learning techniques applied to different domains separately rather than developing generic embedding models
- **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 expects 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, performs a k-nearest-neighbors lookup, and scores the resulting embedding quality.

Simulation - Live Demo

- We have developed a dataset of 150 random images with following categories for simulation purpose

Bulldog	Truck	Pizza	Tower	Shirt
German shepherd	Car	Cake	Statue	Dress
Labrador	Plane	Burger	Arch	Pant

- We obtained embeddings for these database images and also for test image
- Using Euclidean Distance, we find the top 5 images with embeddings similar to the test image

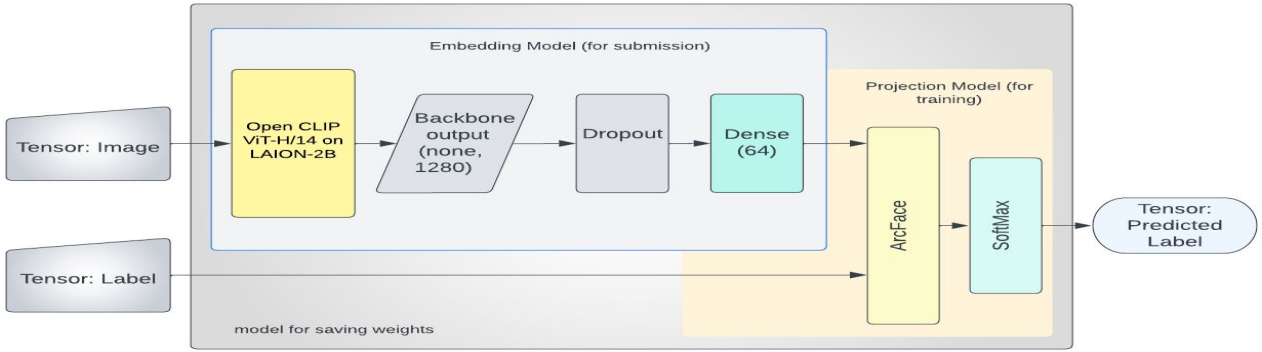
Method - Datasets

- Imagenet - https://www.image-net.org/index.php - 1K classes
- Products -10K - https://products-10k.github.io/ - 10K classes
- Google Landmark Recognition 2021 - Top 7k class images - https://www.kaggle.com/competitions/landmarkrecognition-2021/data
- Total 17K classes of objects
- To reduce the dataset size, each dataset has only 50 images per class
- 90% data is used for training while 10% is used for validation.
- 478185 images for training , 62828 images for validation

Method - Datasets

Dataset sample with images and corresponding class labels

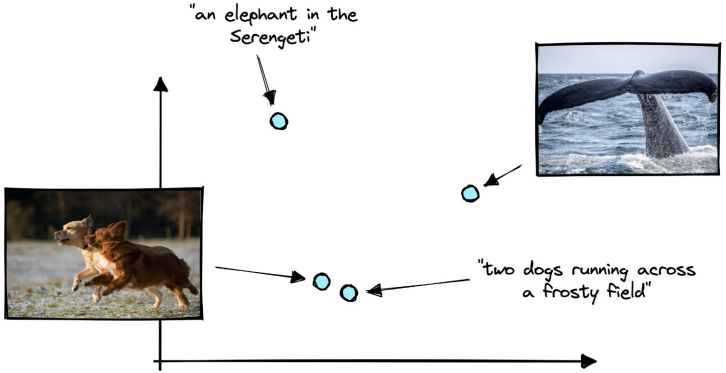
Method - Model Architecture

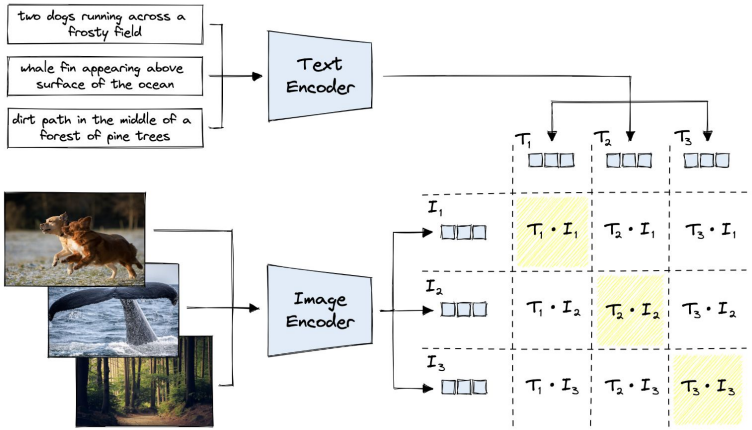


- **Embedding Model: Backbone(CLIP) + Dropout + Dense(64-D) + L2 Norm**
  - Transformer based OpenClip-ViT-H-14 model (Backbone) for feature extraction
  - We use pre-trained weights on a two-billion-scale imagetext pairs dataset LAION-2B
  - Dropouts (0.2) layer, Dense layer (64-D) and L2 Norm
- **Training Model: Backbone(CLIP) + Dropout + Dense(64-D) + ArcFace + Softmax(17691 classes)**
  - We use ArcFace layer during training for better margin classification and to improve softmax loss

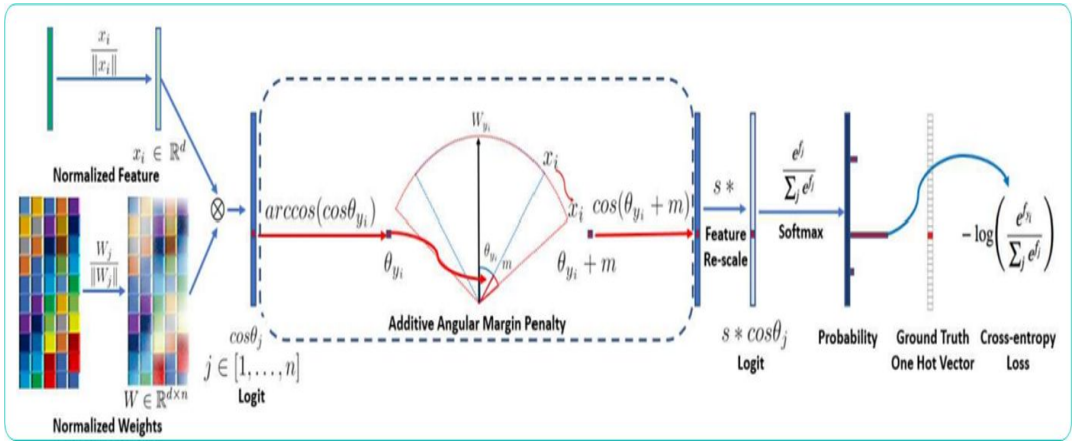
Method - CLIP Model (Backbone)

- Computer vision models in particular perform well in specific tasks, but often fail to generalize to tasks they have not been trained on
- OpenAI's CLIP (Contrastive Language-Image Pre-Training) closes this gap by reframing the problem and using the contrastive pre-training.
- Instead of predicting label text, CLIP is training on predicting how likely this image corresponds to text.
- Due to 'Zero-Shot' capabilities, CLIP models can be applied to nearly arbitrary visual classification tasks
- For a strong baseline, we use the transformer based OpenClip-ViT-H-14 for feature extraction
- Pre-trained weights on a two-billion-scale imagetext pairs dataset LAION-2B from huggingface.co



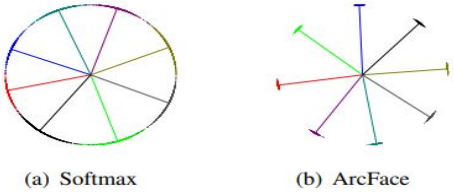


Method - Arcface Layer



SoftMax loss function

$$-\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}$$



(a) Softmax (b) ArcFace

x_i denotes the deep feature of the i-th sample, belonging to the y_i-th class.
W_j^T denotes the j-th column of the weight W and b_j is the bias term.
The batch size and the class number are N and n, respectively.

ArcFace loss function

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

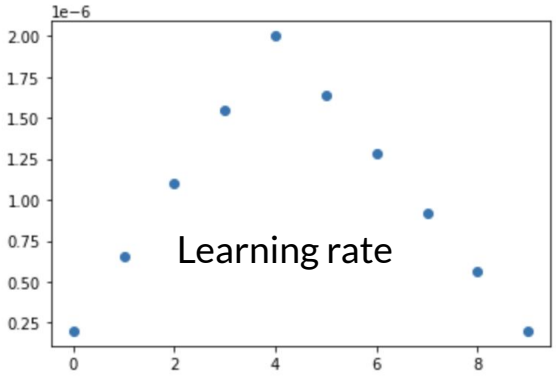
where θ_j is the angle between the weight W_j and the feature x_i
s - feature scale, the hypersphere radius
m - angular margin penalty

Method - Configuration

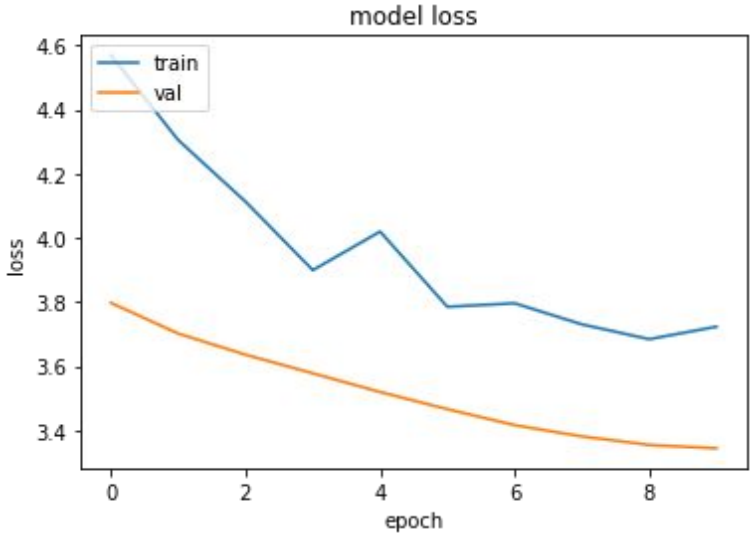
- We used Kaggle Notebook for training the model. All required computing resources are made available by Kaggle in the notebook.
- TPU V3-8 ,8 cores,128 gb ram
- Access to datasets is also easily available in the kaggle environment.
- We built the model using Tensorflow and TPU. It takes about 3 hours to train the model in the environment, for 10 epochs.
- The embedded model with its weights is to kaggle for scoring
- We simulated the performance of the embedding model by computing embeddings of test images from various object types.

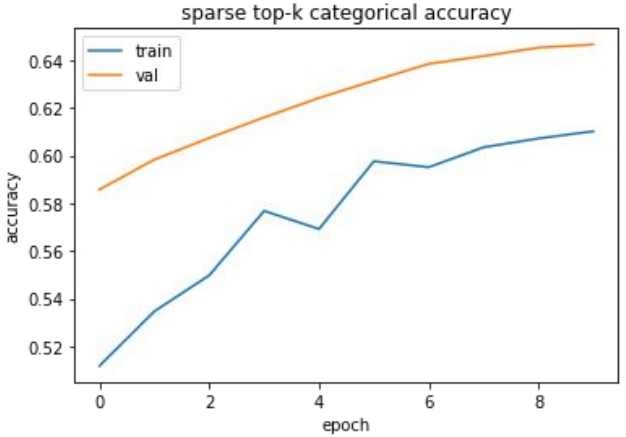
Method -HyperParameters

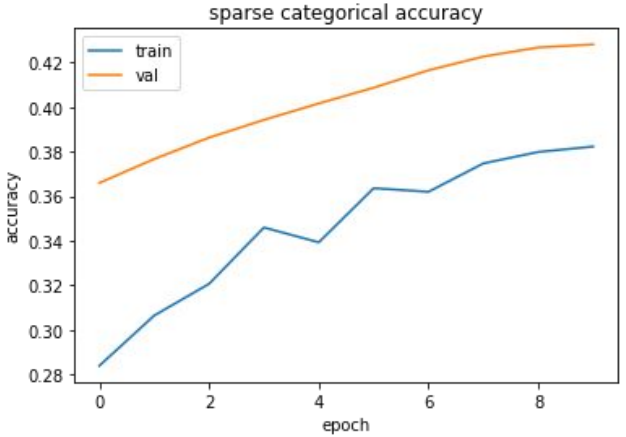
- Batch size is set at 200
- Linear Warmup With Linear Decay as the learning rate schedule
- epochs= 10
- Optimizer : **Adam**
- Data Augmentation : Horizontal flip, resize, shift, scale, rotate, cutout, random brightness, contrast, and RGB-shift
- Additive angular margin loss or **ArcFace** and margin of **0.3**
- **Cross entropy loss** is calculated



Results - Loss & Accuracy







Results - Kaggle Competition

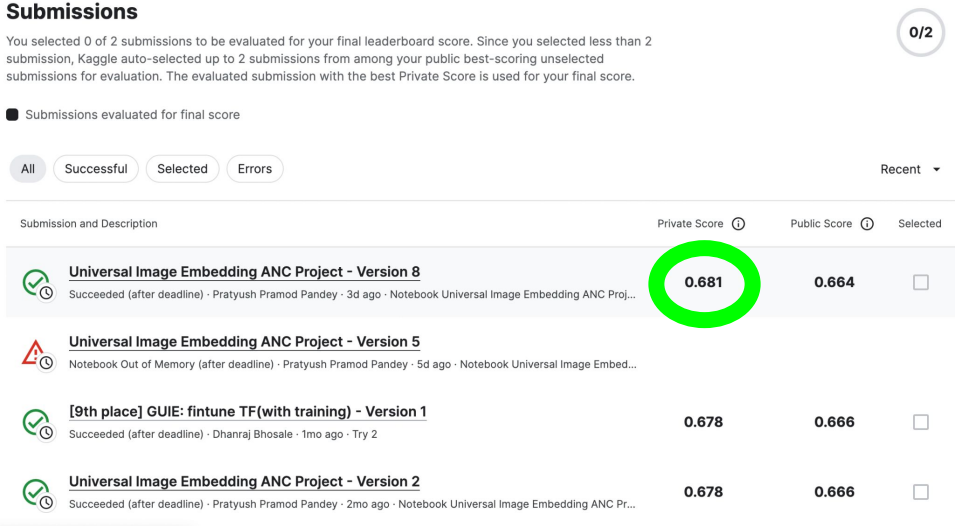
The model trained and submitted by us for the competition scored 0.681 which is at par with the top 10 score amongst all competitors. It is trained with 17691 different categories.

[GitHub Repo Link]

LeaderBoard

Prize Winners

#	△	Team	Members		Score	Entries	Last	Code
1	—	[cuilab.ai] 北京大学医学部生物信息系崔庆华实验室			0.728	206	2mo	
2	—	Xiao			0.709	181	2mo	<>
3	—	---			0.692	312	2mo	
4	▲ 3	Ivan & Simjeg & CLIP-Art			0.688	146	2mo	
5	▼ 1	NS embedding			0.688	362	2mo	<>
6	▼ 1	IRonCLIP			0.685	139	2mo	<>
7	▲ 1	no name			0.682	300	2mo	
8	▼ 2	MOONMOON			0.682	63	2mo	
9	▲ 2	Akihiro Katsura			0.677	78	2mo	<>



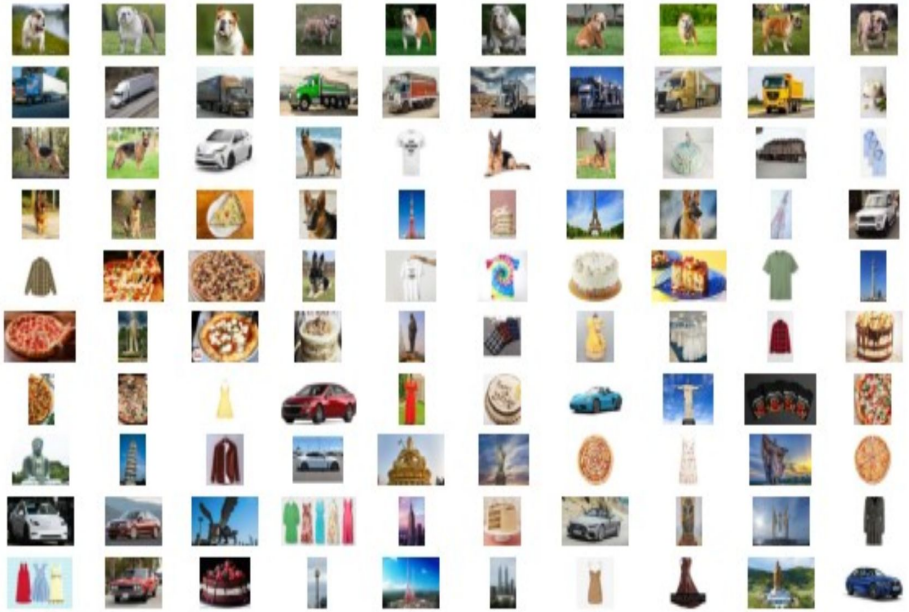
Simulation Results

Test Image and Top 5 Matches

```
1/1 [==============================] - 80s 80s/step

Top 5 Matches are:
```

Dataset sorted on embedding similarity



Future Enhancements

- Model to take care of tricky cases like identifying difference between an actual Eiffel tower and a souvenir or replica of the Eiffel tower
- Enhancing the classification categories beyond the current 17K classes which might make the embeddings even more universal. This would require training with additional datasets
- New approaches for model design may be considered
- New approaches for model training may be used. E.g. Continuous training may be deployed in applications where images are continuously processed like Google lens

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

Questions

Top 5 Matches