Vectorized Similarity Search in Multi-modal Databases

Yaohai Zhou, Jo-Hua Wu, Yu-Ruei Chang, Yi-An Chen

Our Demo

http://18.237.16.206:8501/

ABOUT US

We are four graduate students from Master of Engineering.

There are different types and formats of data in the world

Text

Images

Html

Audio / Video Data


How to make good use of them?

SOLUTION

IDEA

Integrate data with values of different scales, distributions, and representations into a global feature space.
(i.e., multi-modal databases)

Make data easily accessible and usable for different purposes.

METHOD

Embedding vectors generated from heterogeneous multimodal data, and enabling similarity search.

INPUT

heterogeneous data
(Texts & Images)

OUTPUT

Captions /
Similar Images

WORK PROCESS

1

EMBEDDING

Embed texts and images into high-dimensional spaces based on the techniques of Contrastive Language-Image Pre-training (CLIP)

2

SIMILARITY SEARCH

Enable similarity search for heterogeneous data at the level of embedded vectors

WORK PROCESS

3

EVALUATION & FINE TUNE

- Use recall rate and error rate to evaluate the result of similarity research.

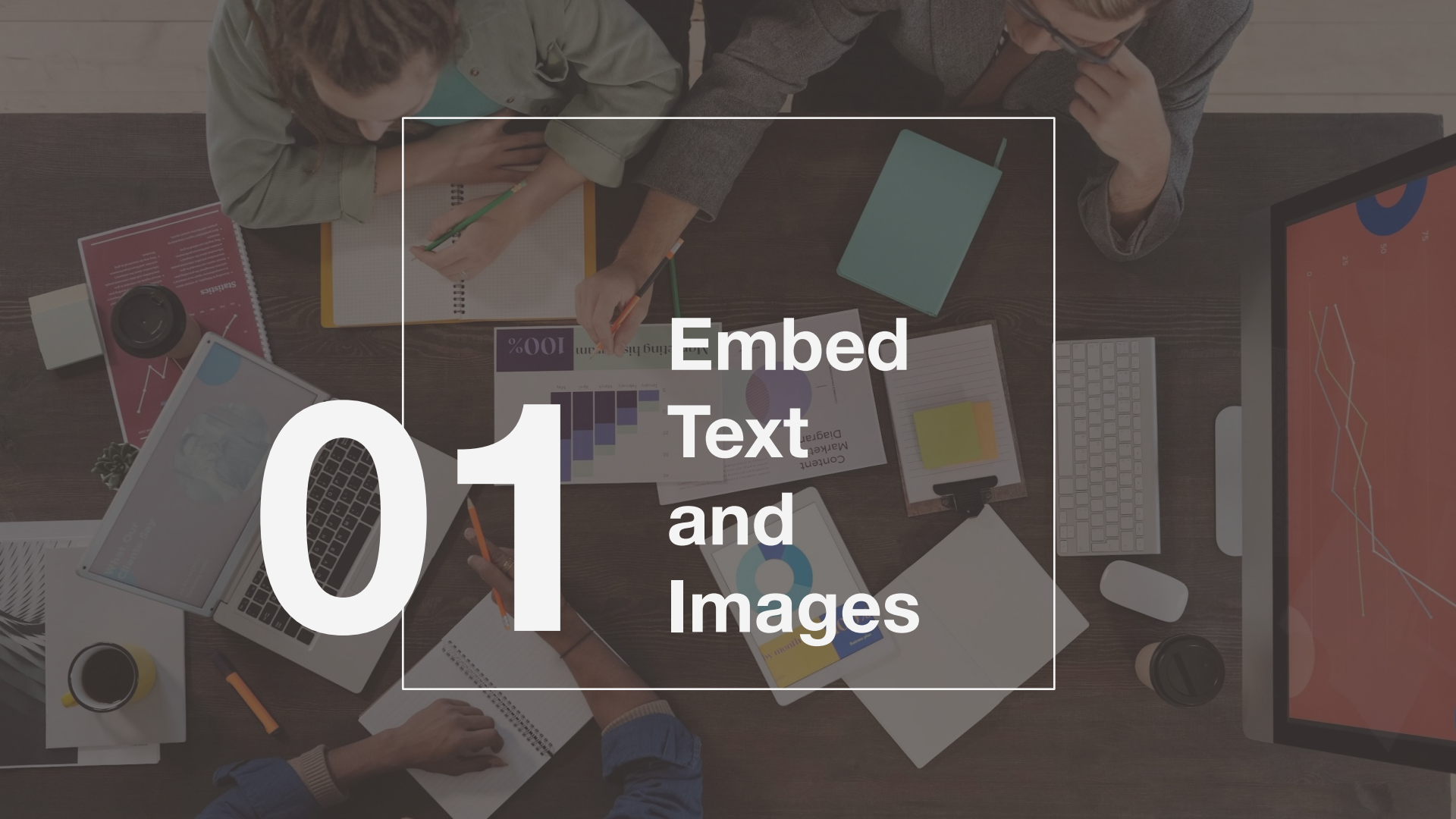
- Fine Tune the model

4

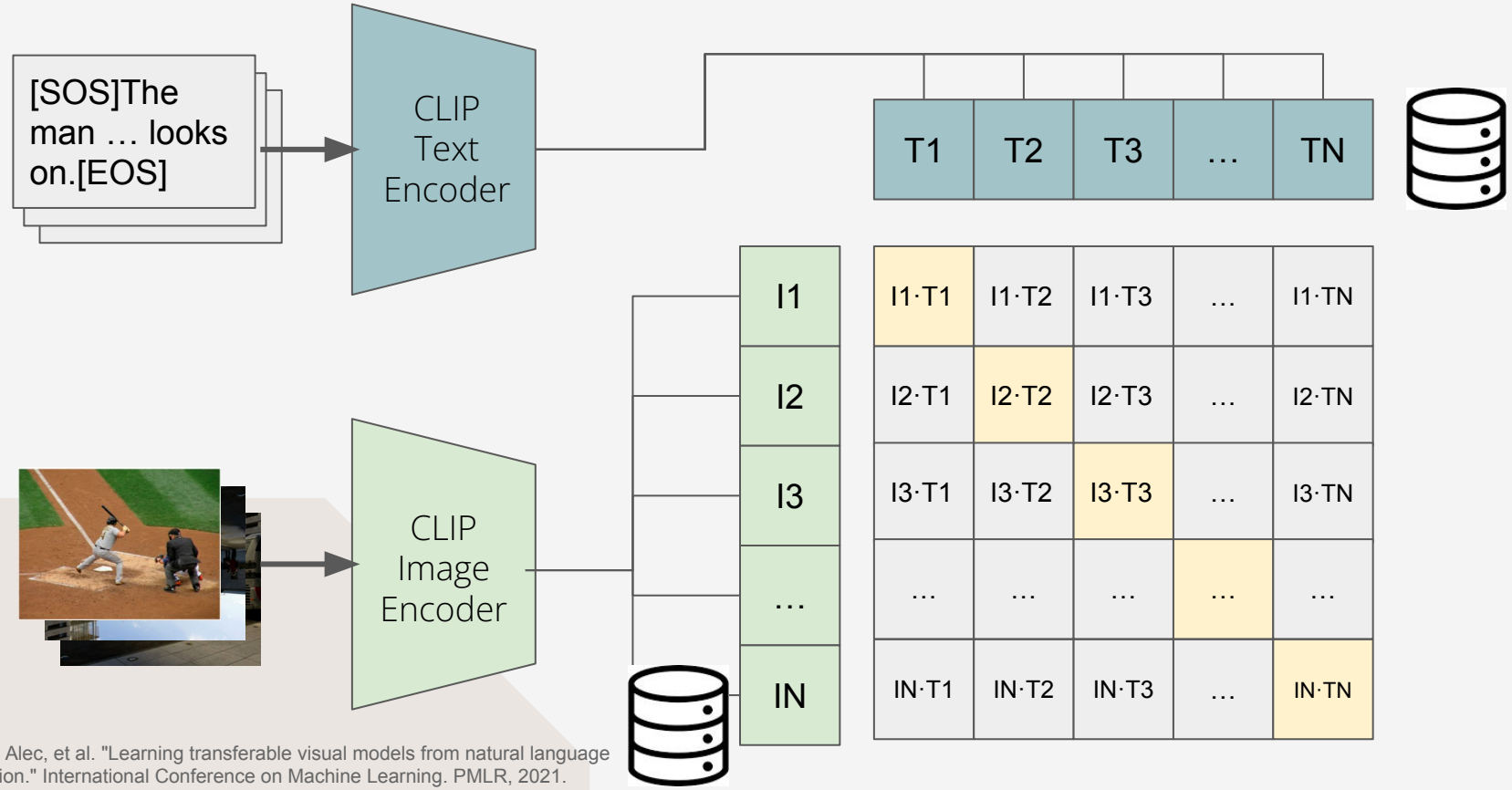
USER INTERFACE

Build an user interface by using Streamlit

DEMO TIME

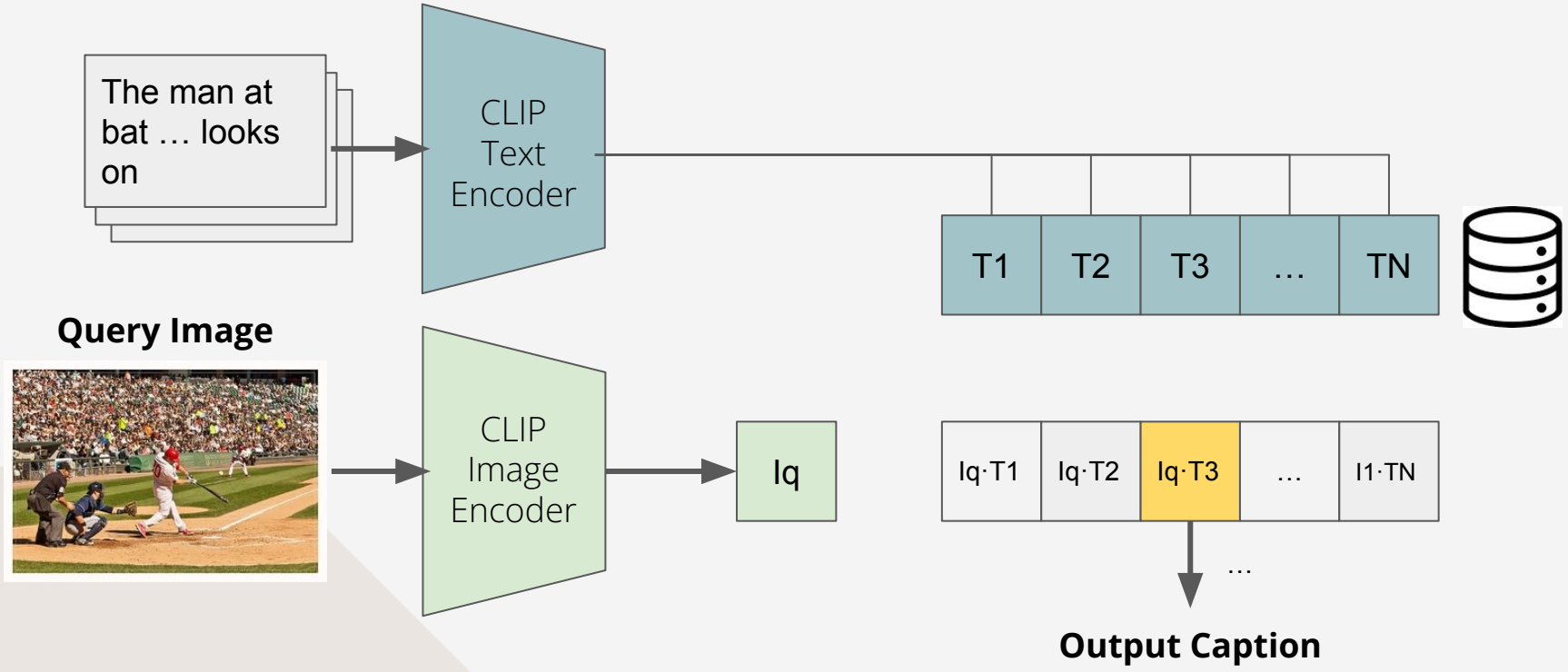

01
Embed
Text
and
Images

CLIP: Build Embedding Vector Database



Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International Conference on Machine Learning. PMLR, 2021.

CLIP: Query Embedding Vector Database



Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International Conference on Machine Learning. PMLR, 2021.

MSCOCO Image Captioning Dataset

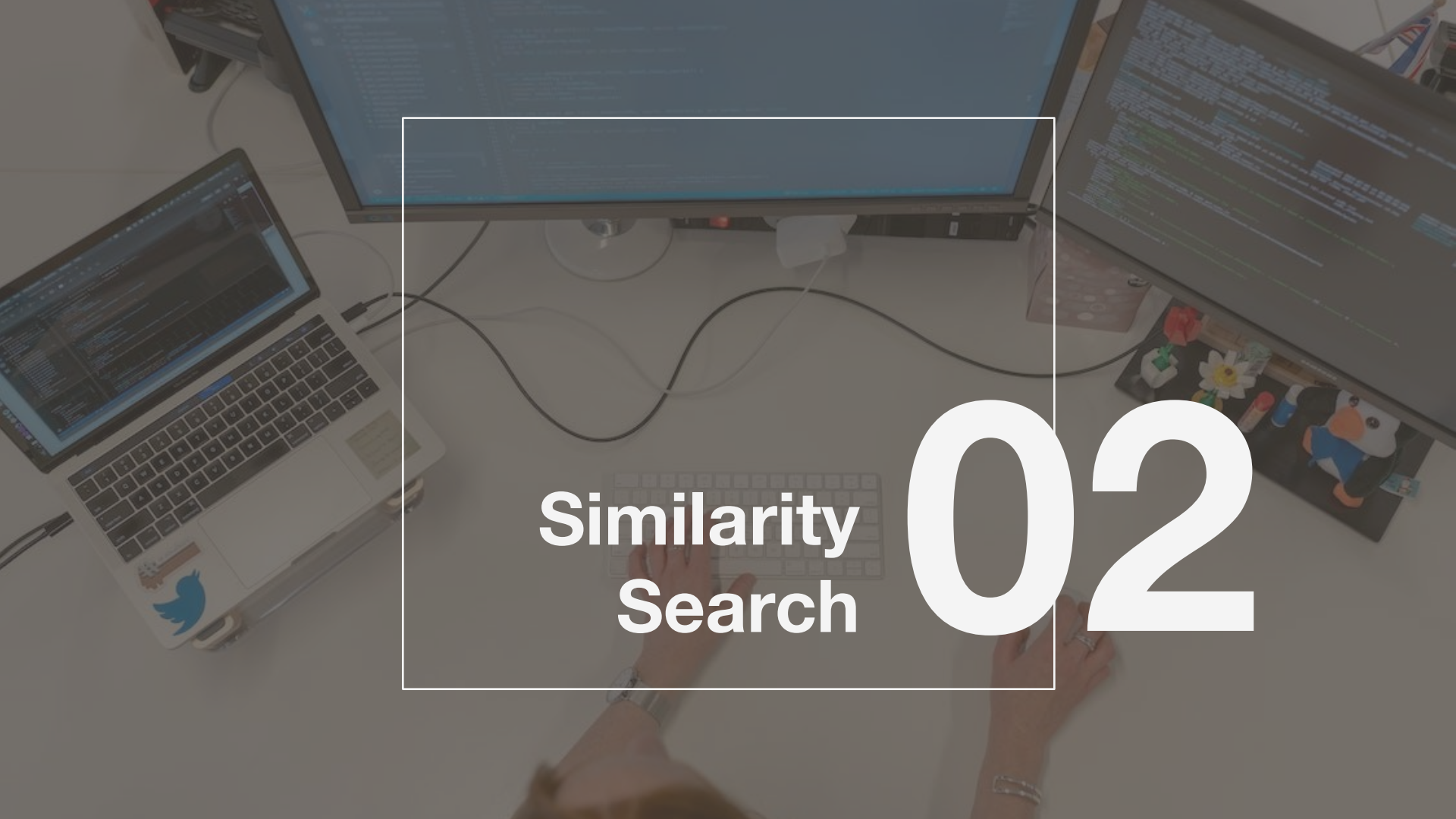
The man at bat readies to swing at the pitch while the umpire looks on.

Dataset	Train	Val
Images	82783	40504
Captions	414113	202654

Random Sample 80000 {Image & Caption} Pairs

Database	Main	Test
{Image& Caption}	80000	80000
Usage	Generate Embeddings & Finetune	Evaluate

https://cocodataset.org/#home



Similarity Search 02

k-Nearest Neighbor (kNN) Search

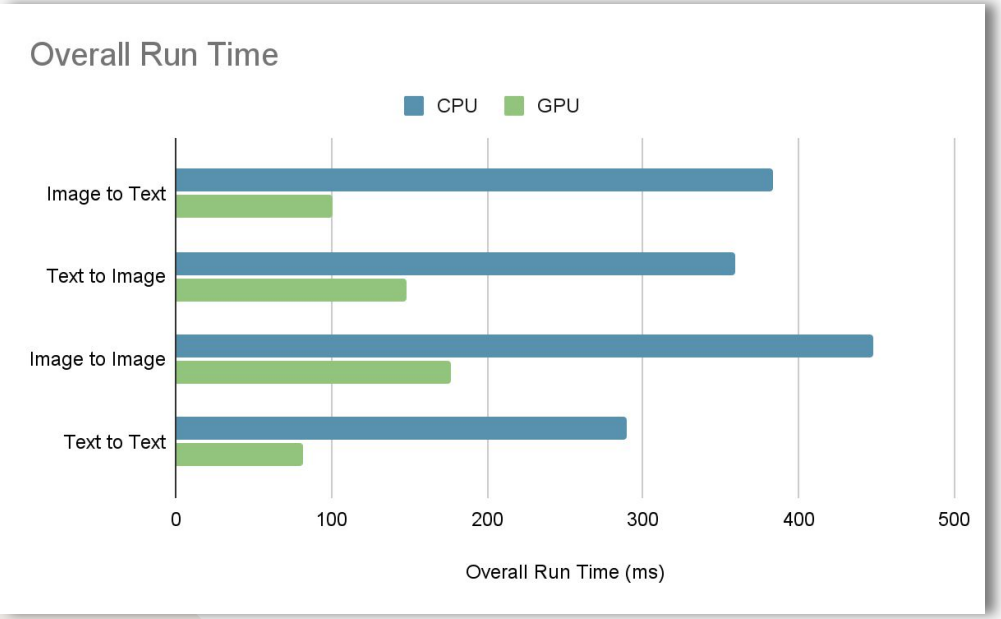
Compute similarity scores with all candidates and return the most k similar ones

- Distances/Similarities: L2, dot, cosine
- Complexity: n*d + k*logn
- GPU acceleration with torch package

https://www.ibm.com/topics/knn

kNN GPU Acceleration

GPU makes overall run time 3x faster

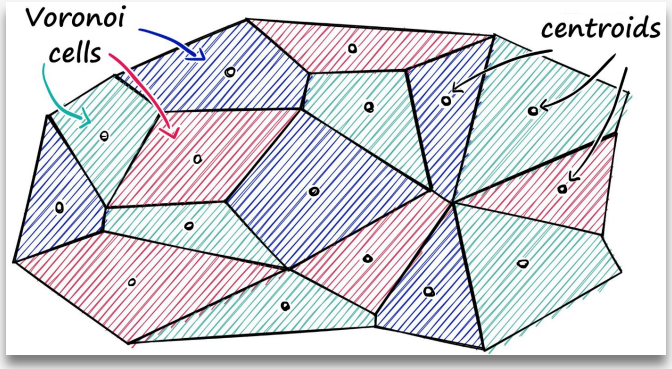


Faiss Approximate Nearest Neighbor (ANN) Search

Index based on inverted file (IVF)

- Preprocess: partition into s groups
- Search:
  - Computes similarities with s group centroids (s*d)
  - Select **t groups** with the most similar centroids (t*logs)
  - Compute similarities with all candidates in **t groups** ([n/s]*d*t)
  - Return the most k similar ones (k*log[t*n/s])

- Complexity: **ANN ≤ kNN**
  - ANN: [t/s]*n*d + s*d + t*logs + k*log[t*n/s]
  - kNN: n*d + k*logn

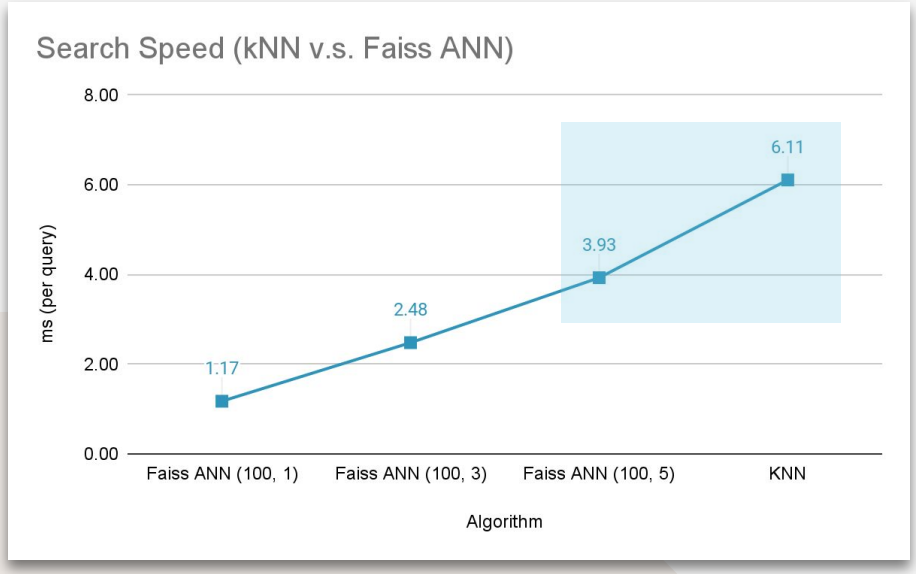


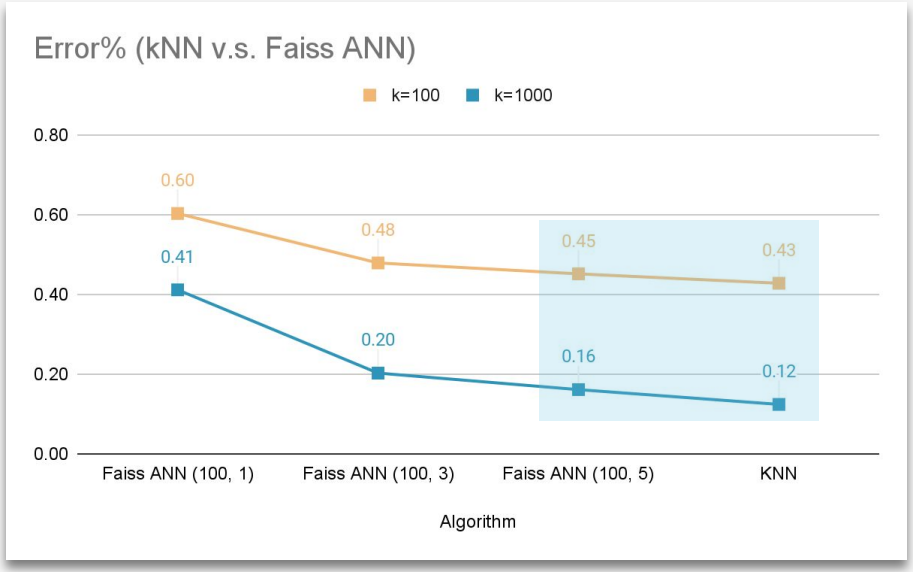
https://www.pinecone.io/learn/faiss-tutorial/

Efficiency and Performance Trade-off

36% faster, but only 3% accuracy drop

[Image to Text] kNN v.s. Faiss ANN(s, t)





03 Evaluation & Fine tune

Evaluation

Miss rate (false negative rate) is used as error rate to evaluate the model performance:

$$Miss\ rate = \frac{\sum e_i}{\#\ of\ samples}$$

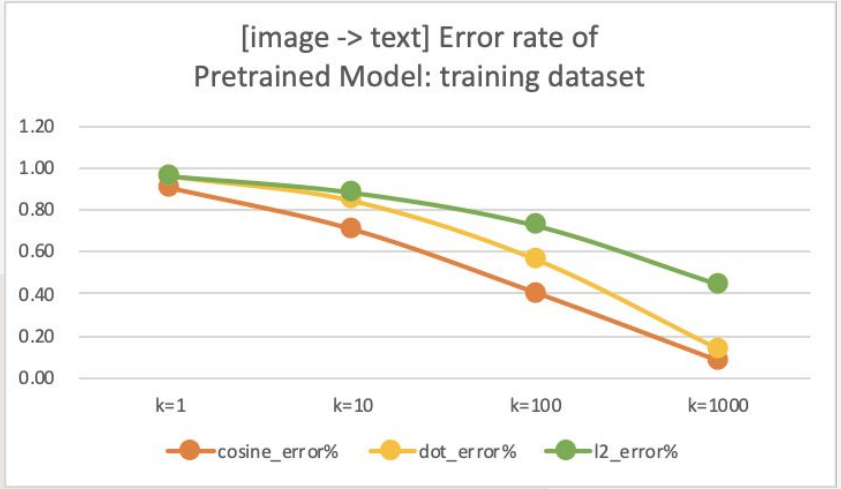
An error (ei) is defined as:

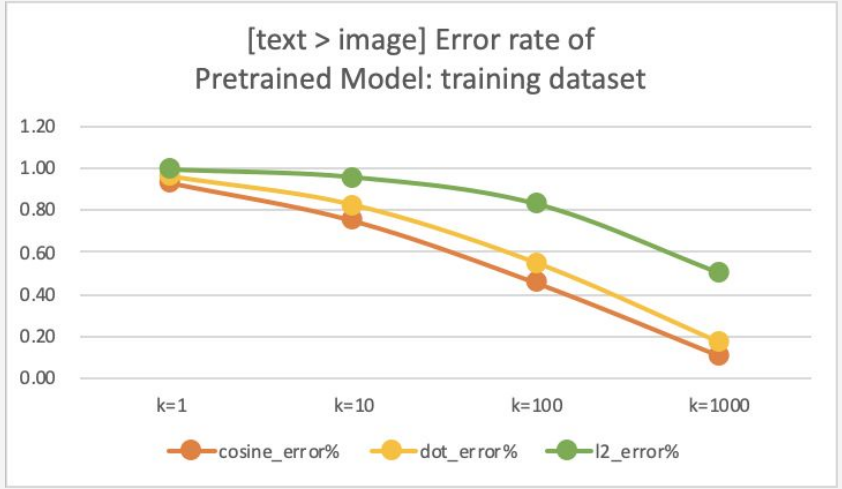
$$e_i \begin{cases} e_i = 1;\ when\ k\ neighbors\ does\ not\ contain\ the\ correct\ answer \\ e_i = 0;\ when\ k\ neighbors\ contains\ the\ correct\ answer \end{cases}$$

$$where\ \ k = \{1, 10, 100, 1000\}$$

Evaluation

- As the value of k goes up, the error rate goes down.
- The lowest error rate happens when distance method = **'Cosine'** in pretrained models for both [image to text] and [text to image] function.



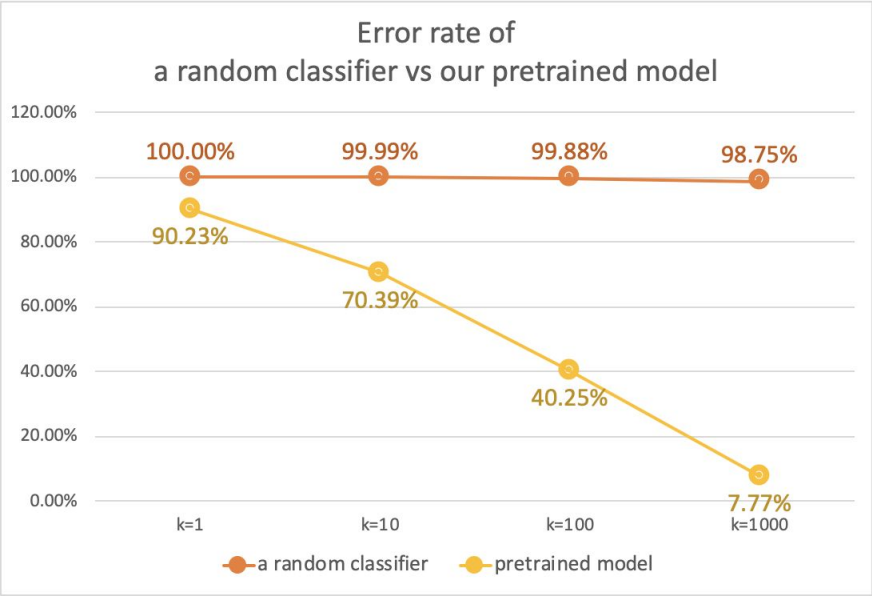


Evaluation

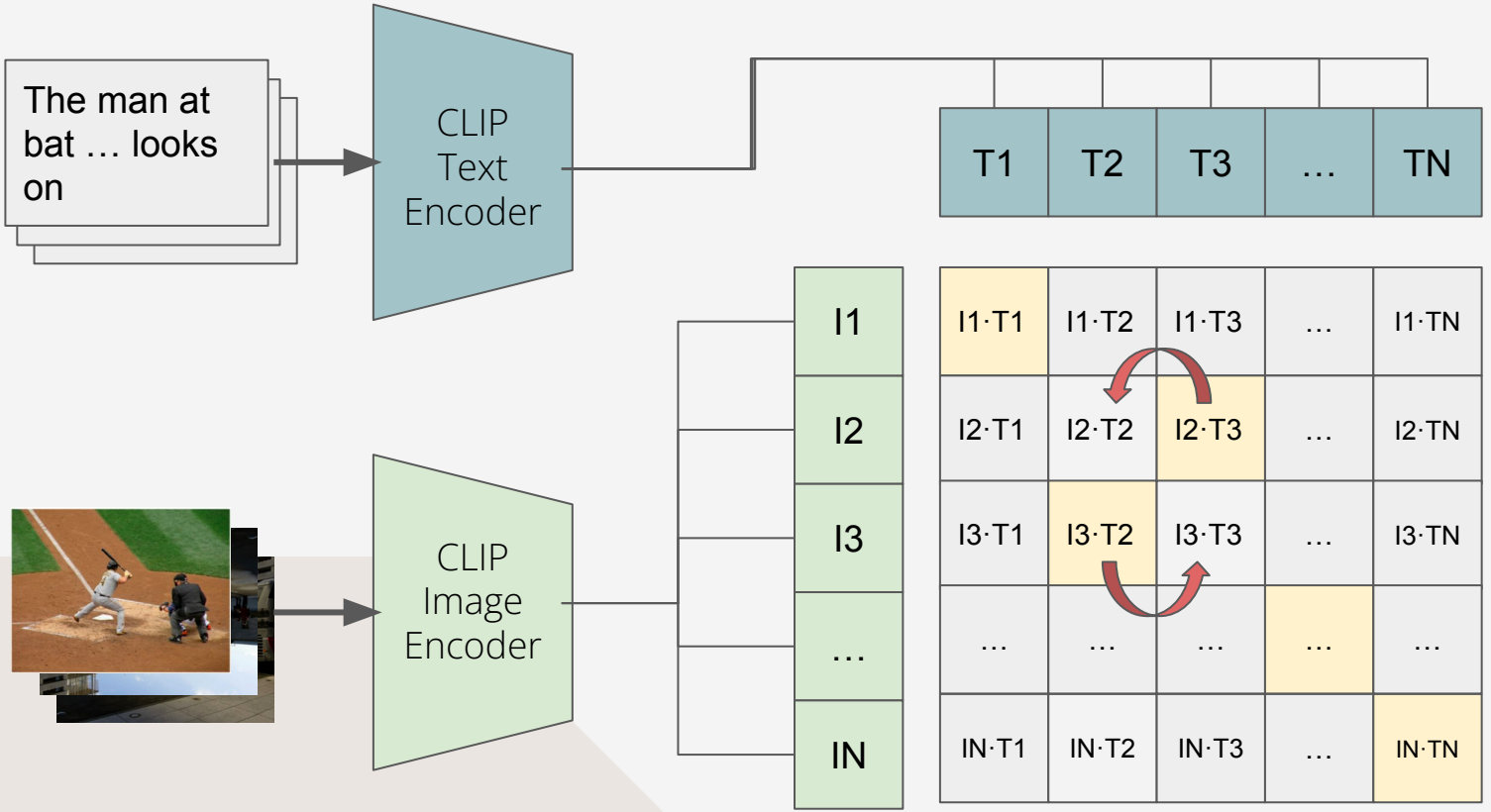
- The miss rate for a random classifier (Random guess) is:

$$1 - \frac{k}{\#\ of\ samples}$$

- Our model has 10% less error than the random classifiers when k=1, and twelve times less error when k=1000.



CLIP Finetune on MSCOCO Dataset

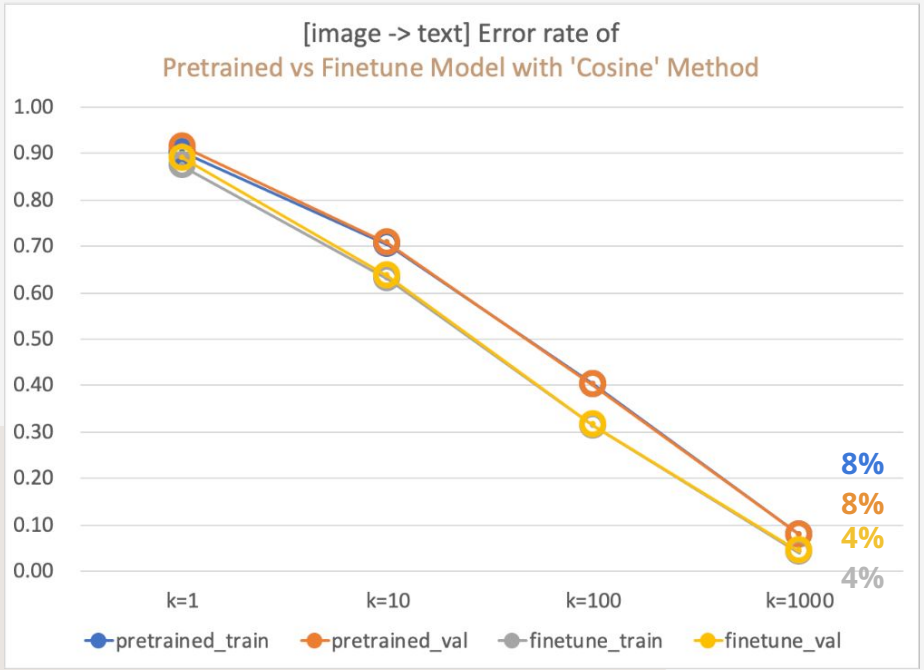


Loss Function: CrossEntropyLoss

Small Learning Rate: e-8

Evaluation

For both [image to text] and [text to image] function, we got better results in finetune models.



User Interface 04

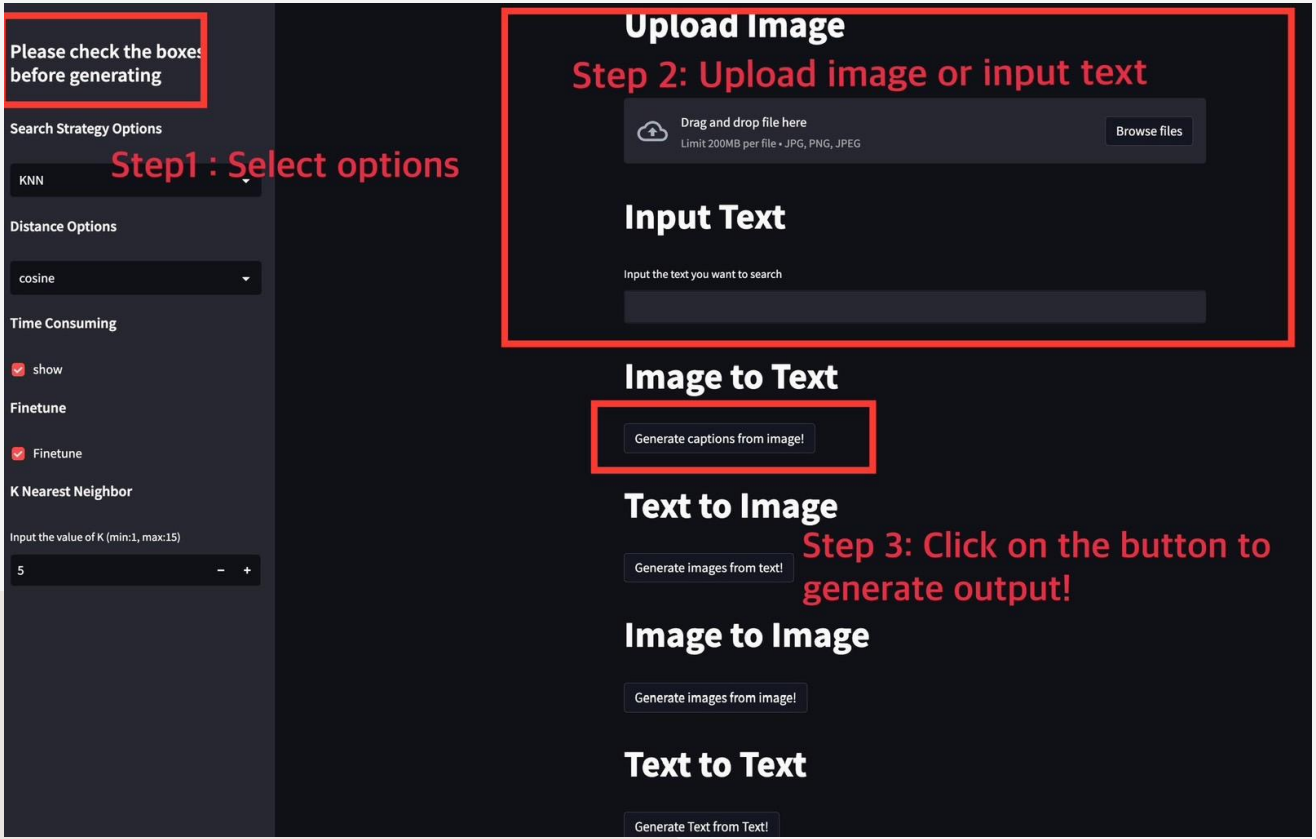
Try it yourself!

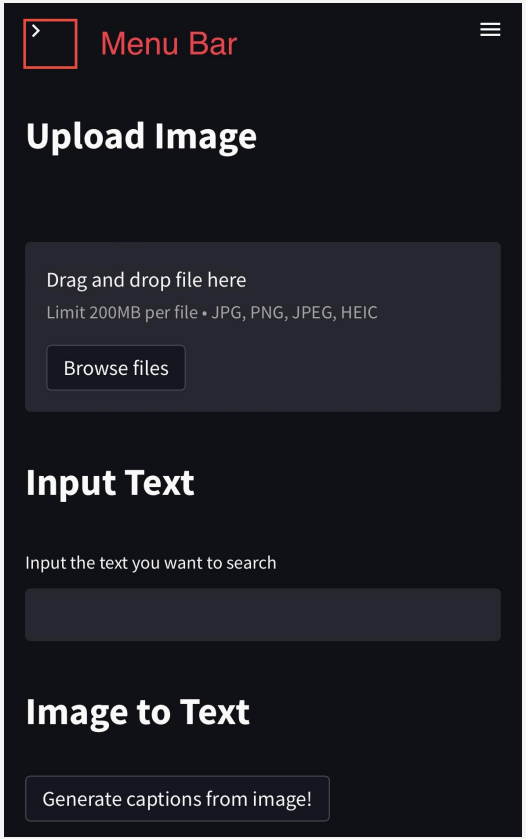
Scan the QR code above!

Or http://18.237.16.206:8501/

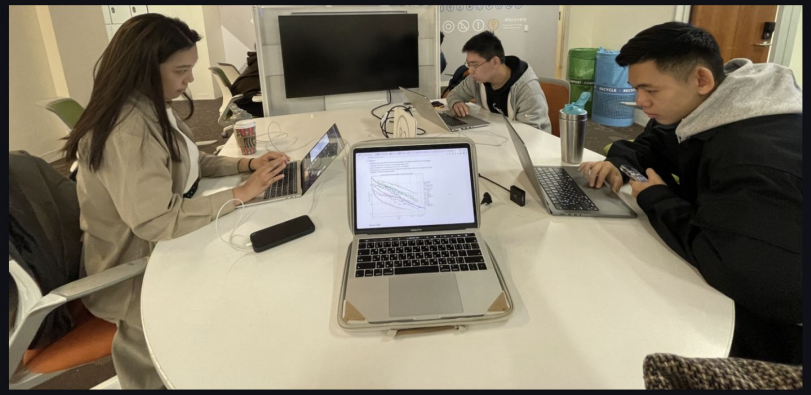

05
DEMO

User Interface - Streamlit





Example Output- Image to Text



1. Similarity: 1.0

a group of people busy at work studying.

2. Similarity: 0.9910773

Three students do work in the crowded library.

3. Similarity: 0.9549173

A group of people gather together on their laptops in a room.

4. Similarity: 0.9511437

An white Apple computer is featured in this photo.

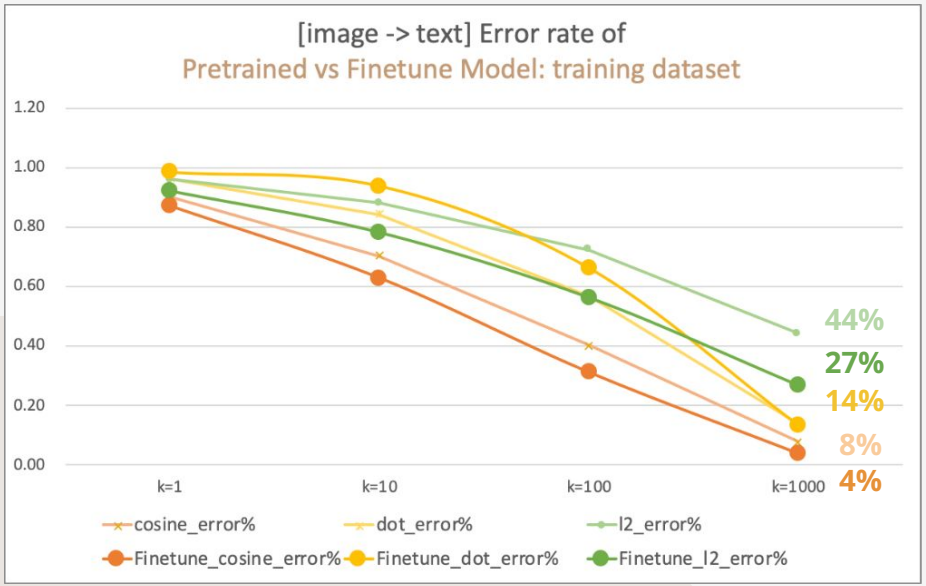
5. Similarity: 0.94546354

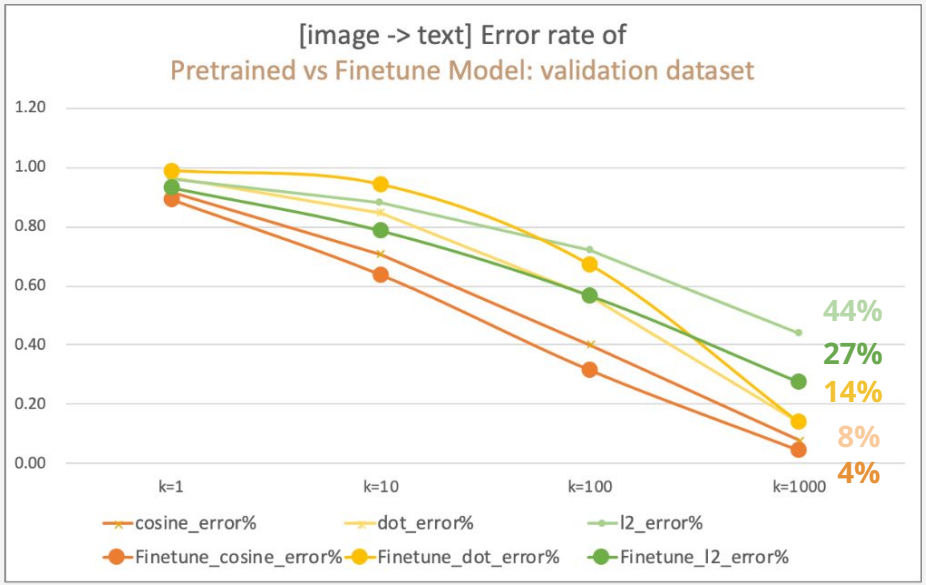
A group of people sitting in a room with laptops.

816 ms for executing

Evaluation

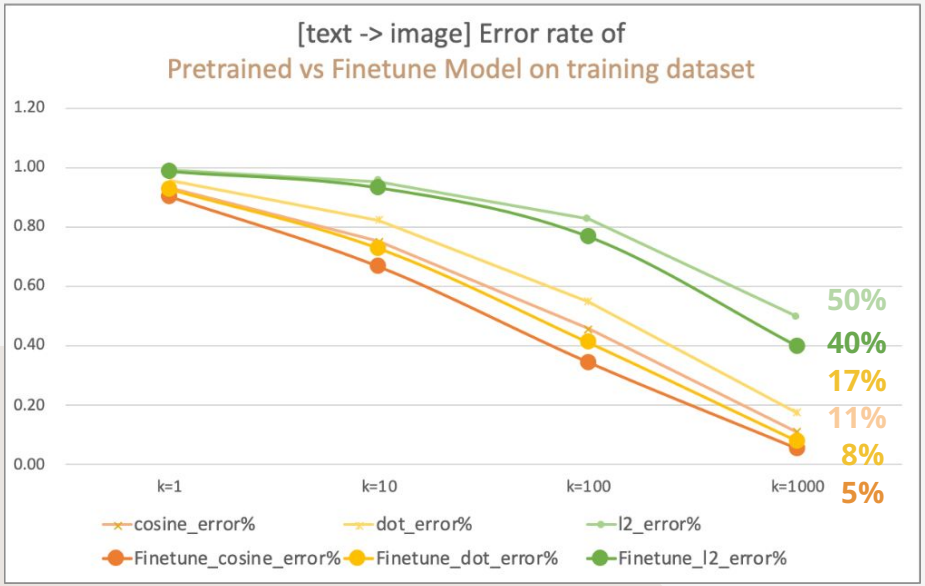
For [image to text] function,
distance method = **'Cosine'** leads to lowest error rate,
distance method = **'L2'** has better result than distance method = **'Dot'** in both training and validation datasets.

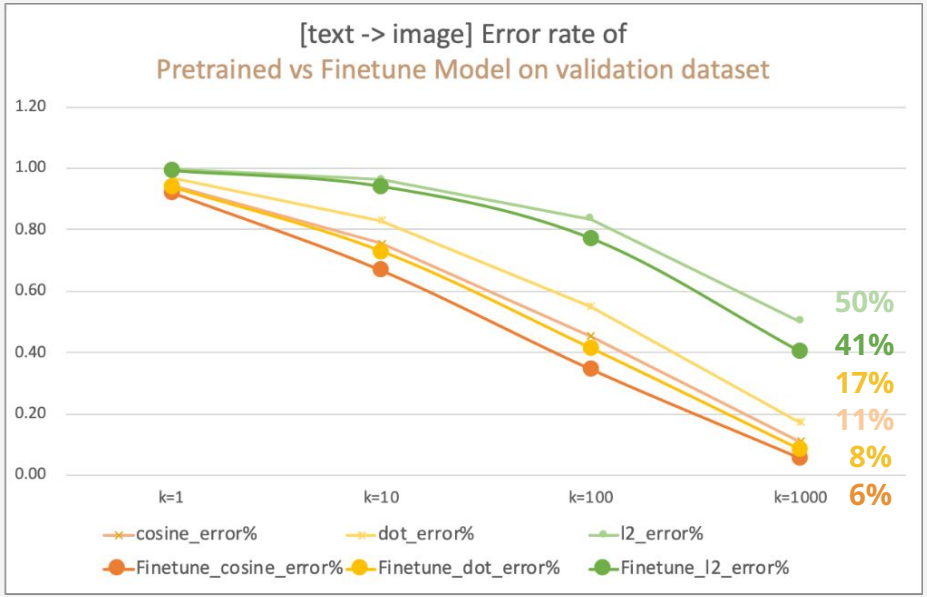




Evaluation

For [text to image] function,
distance method = **'Cosine'** leads to lowest error rate,
distance method = **'Dot'** has better result than distance method = **'L2'**





CLIP Finetune