Image Retrieval and Audio Summarization for Blinds

Team 11: Abhishek Acharya, Arha Samanta, Lakshay Wadhwa, Pijush Bhuyan

Mid Project Review- Report

Project Github Link:
https://github.com/lakshay22037/CSE508_Winter2023_Project_11

Updated Problem Statement

Image Retrieval and Audio Summarization for Blinds. The project aims to address the challenge encountered by individuals with impaired vision when browsing the internet, particularly in the context of visual media.

Updated Literature Review

At present, there are a few platforms that help us to retrieve similar images, or get an audio description of the image. But we plan to build a user-friendly platform, especially for blinds, which does both together.

There are no such live platforms available that can help a blind person understand the given image and also check out some similar images if the idea is not clear with the given image.

Combining such features and also working on building a user experience design specific to blinds, will help us to make a product for solving a social problem.

Some References of papers that are based on this similar topics are:

- A Decade Survey of Content Based Image Retrieval using Deep Learning
  This Paper summarizes a decade-long CBIR (Content Based Image retrieval) method used in IR systems.
- Content-Based Image Retrieval by Clustering
  This paper also reviews the development of content-based image retrieval (CBIR) techniques from the early years to the present and discusses the current state-of-the-art in the field.
- Deep Learning for Image Retrieval: What Works and What Doesn't
  This paper presents a comprehensive study of deep learning methods for content-based image retrieval, including a detailed review of various deep learning architectures and their performance on standard benchmark datasets.
- Teena Varma, Stephen S Madari, Lenita L Montheiro, Rachna S Pooojary, 2021, Text Extraction From Image and Text to Speech Conversion, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NTASU – 2020 (Volume 09 – Issue 03)

Our Approach

From our Literature Review, we found that Autoencoders and Deep CNN Classifiers are commonly used to extract embeddings from image data. Autoencoders use a bottleneck layer to create compressed representations of the original input, while Deep CNN Classifiers use neural codes containing activation of convolutional layers to extract powerful features.

For our project, we will combine both the latent space and neural code features to create a hybrid vector representation of the input image. We will generate these hybrid embeddings for all images in our database and use PCA to reduce the dimensionality of the embeddings.

Overall, this approach will enable us to efficiently and accurately represent images in our project, which will be used further to do similarity matching with the query image. This will help people who are blind or visually impaired access the audio descriptions of those related images and gain insight into what the image wants to say.

In conclusion, our system stands out from others in two ways.

A. It offers a more accurate and scalable image retrieval system with audio summarization.
B. It provides a user-friendly web interface that seamlessly interacts with the information retrieval system.

Methodology

1. Firstly, we will extract the embeddings from our image dataset using various methods and store them as a CSV file. This will enable us to represent the images in a compressed format that captures their important features.
2. We will then perform clustering on these embeddings to group similar images together. This will help us to organize our dataset and identify images that are similar to each other.
3. Next, we will build a unigram inverted index using the cluster centers as terms and the corresponding embeddings of each cluster as postings. This index will enable us to efficiently search for images that are similar to a given query image.
4. Finally, we will evaluate the search performance of our system with test queries which will be evaluated using metrics such as mean average precision and also try to optimize the time and space required for the retrieval system. These metrics will help us to understand how well our system is able to retrieve relevant images in response to user queries.

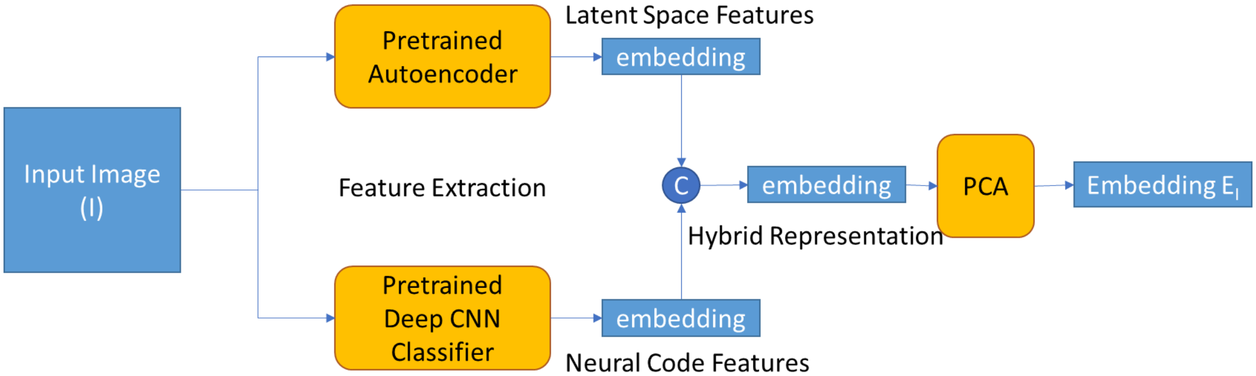


Fig 1 - Embedding extraction

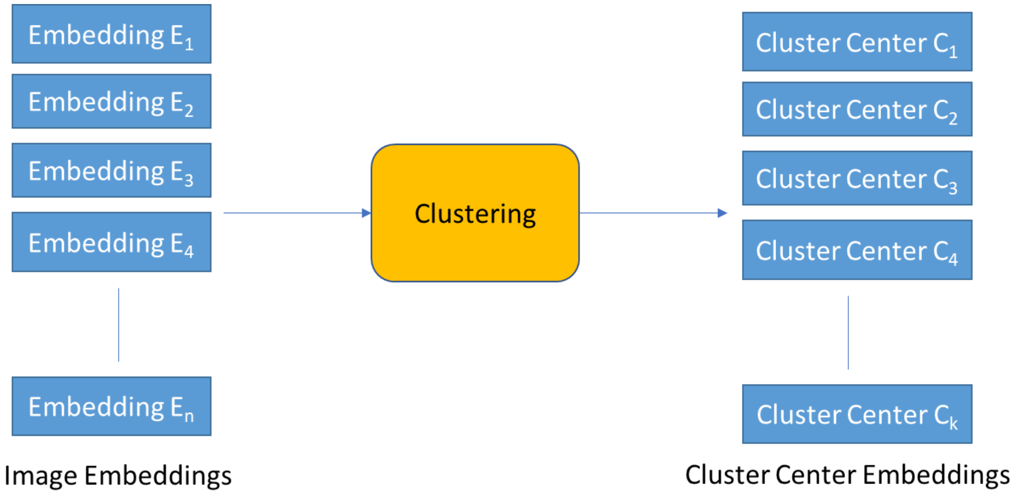


Fig 2 - Clustering the embeddings

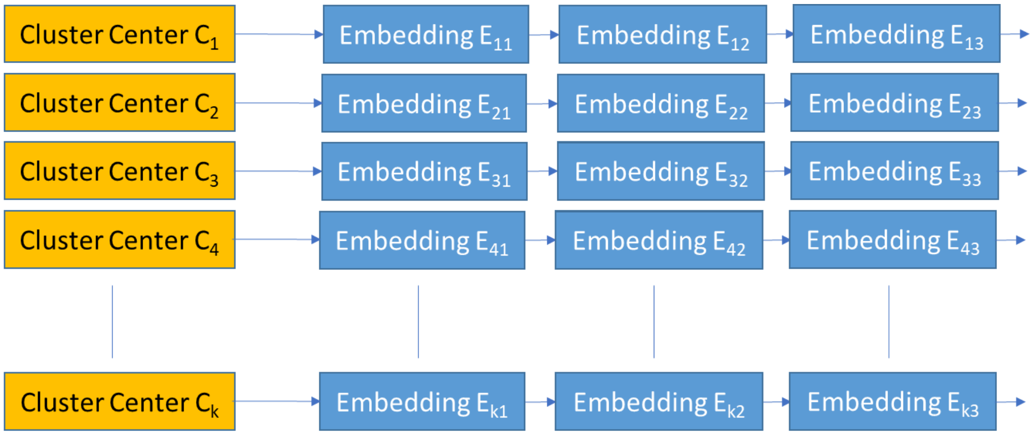


Fig 3 - The inverted embedding index

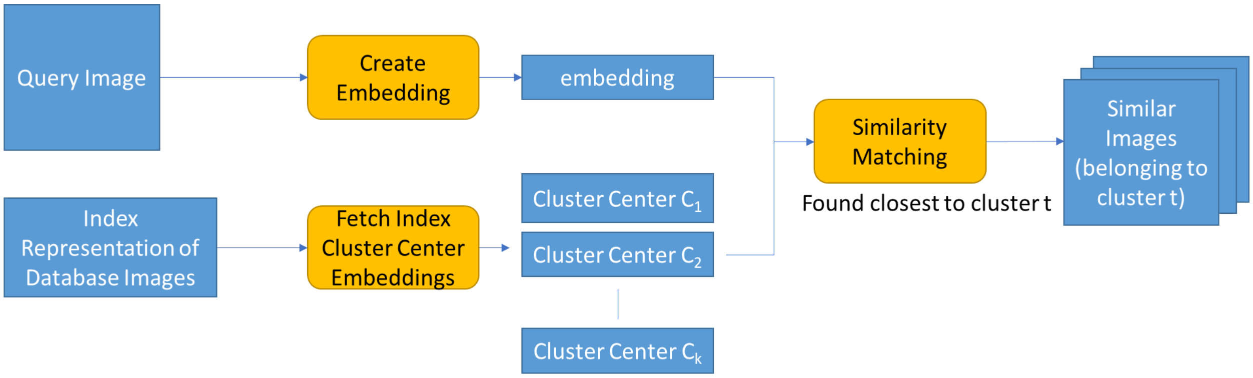

Query Image
Create Embedding
embedding
Index Representation of Database Images
Fetch Index Cluster Center Embeddings
Cluster Center C_1
Cluster Center C_2
Cluster Center C_k
Similarity Matching
Found closest to cluster t
Similar Images (belonging to cluster t)

Fig 4 - Processing a test query image

Baseline Results

A baseline inverted index was built using the embeddings extracted from the CIFAR 100 dataset. A pre-trained EfficientNet-B0 architecture was used for the same. The pretrained weights are available at this link.

Note -

a) K-Means clustering was used for finding the cluster-centers and corresponding labels on the train and test image datasets

b) For similarity score, cosine similarity was used to map a given query embedding to the closest term in the inverted index dictionary.

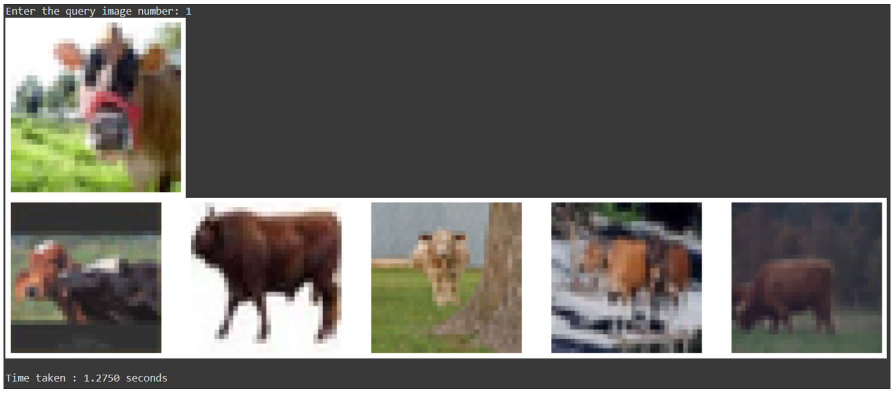
Specifications -

Train Images	: 50000
Image Size	: 32x32x3
Embedding Size	: 1280 (compressed from 3072 features)
Clusters found	: 100
Size of Inverted Index	: 491 Megabytes

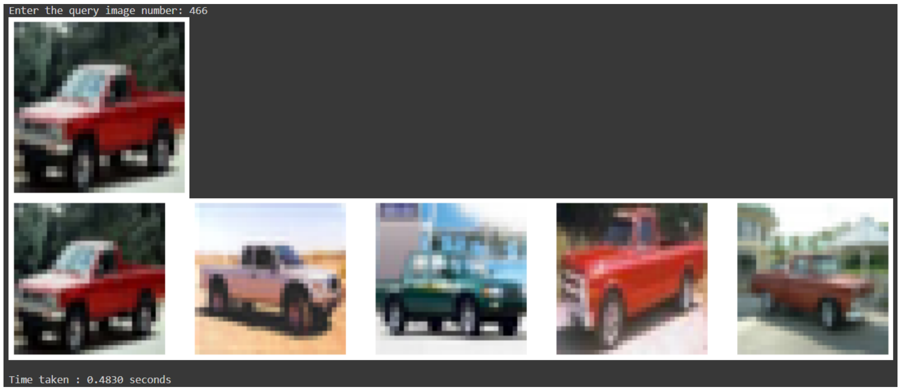
Results -

Test Images	: 10000
Mean Average Precision	**: 92.09%**

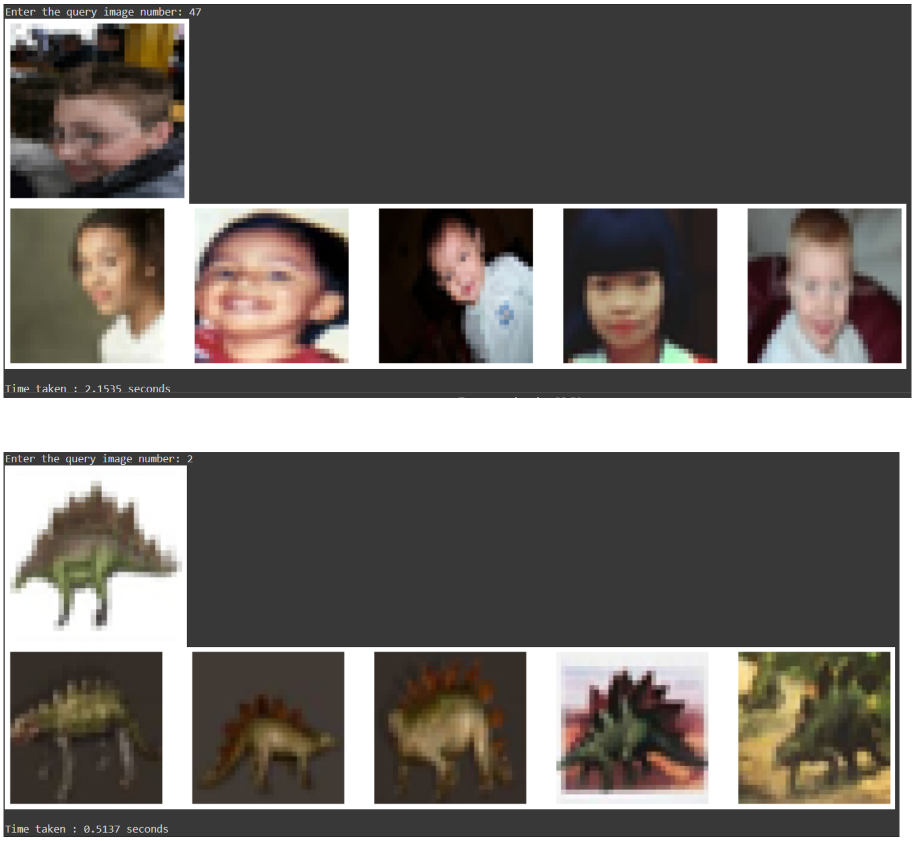
Sample Queries -



```
Enter the query image number: 1
Time taken : 1.2750 seconds
```



```
Enter the query image number: 466
Time taken : 0.4830 seconds
```



```
Enter the query image number: 47
Time taken : 2.1535 seconds
```

```
Enter the query image number: 2
Time taken : 0.5137 seconds
```

Fig 5 - Retrieved Results on the baseline for sample queries

Proposed Method

We tried to extract the latent feature representation of the images using a pre-trained autoencoder on the cifar-100 dataset. The implemented autoencoder model was taken from here.

The latent embeddings extracted (of size 512) were again clustered using k-means clustering technique to see if the can classify the output classes in n-dimensional feature space.

However it was found that autoencoders alone were not good enough to learn differentiating latent vectors which could well separate the output classes of the inputs learnt by the model. They were only good at capturing information that was sufficient enough to reconstruct the input back from the latent features.

This can be easily understood by the 2D T-SNE plot of the embeddings extracted by both the baseline and autoencoder models as shown below -

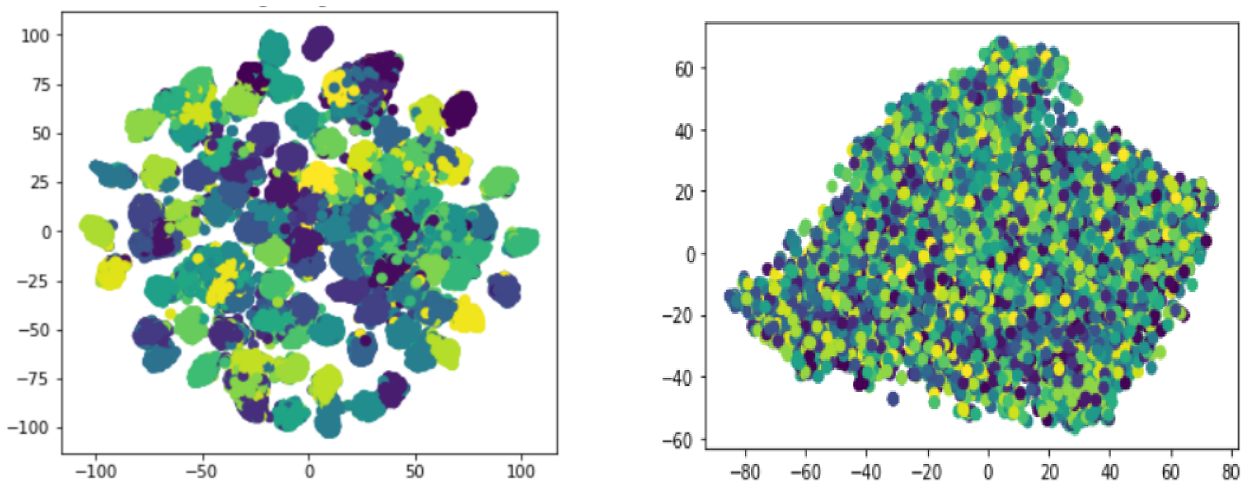


Fig 6 - 2D T-SNE visualization of the embeddings extracted from baseline(left) and autoencoders(right)

Therefore, we tried to combine both the embeddings from the baseline and the autoencoder by simple concatenation of both the embeddings for each image in the dataset.

It was found that the hybrid embeddings so created were well separable among the target classes as shown in the T-SNE plot below. The autoencoder embeddings provided additional information about the images on top of the baseline embeddings which results in a well formed hybrid embedding space.

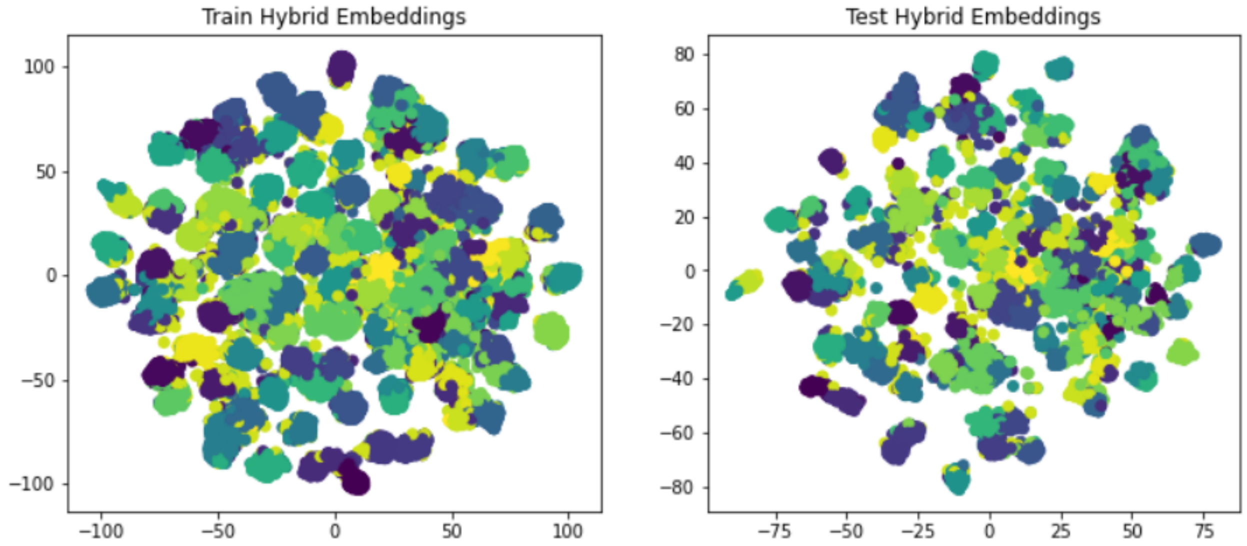


Fig 7 - 2D T-SNE visualization of the hybrid embeddings of the train and test set

Now an inverted index was built with the hybrid embeddings which has the following details

Specifications -

Train Images	: 50000
Image Size	: 32x32x3
Embedding Size	: 1792 (1280+512 compressed from 3072 features)
Clusters found	: 100
Size of Inverted Index	: 688 Megabytes

Results -

Test Images	: 10000
Mean Average Precision	**: 95.43%** (significant improvement over the baseline)

Future Work

We will try to reduce the embedding space using linear dimensionality reduction techniques like PCA to compare the tradeoff between performance and space/time for processing each query. We will also like to try out non-linear combination techniques for creating the hybrid embeddings to compare the results. Finally we will also integrate image summarization and audio output using text to speech services.