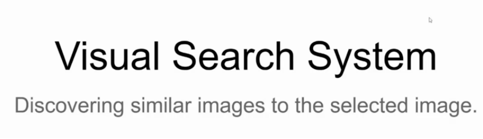
System Design Visual Search System

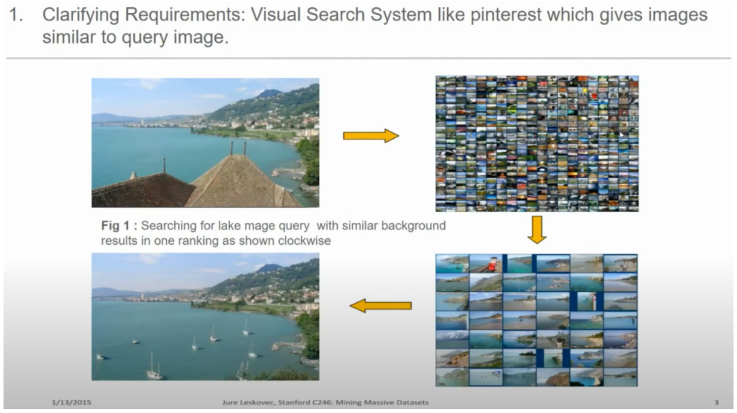
Video Link: https://www.youtube.com/watch?v=5mkegPovTI8&list=PL_b_MRp1BnEj4UuHv1jAGeO1a0VTRXsAk&index=5

1. Objective



Visual Search System

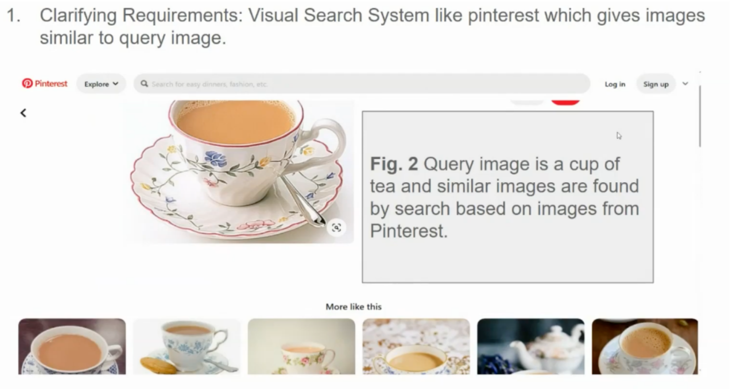
Discovering similar images to the selected image.



1. Clarifying Requirements: Visual Search System like pinterest which gives images similar to query image.

Fig 1 : Searching for lake mage query with similar background results in one ranking as shown clockwise

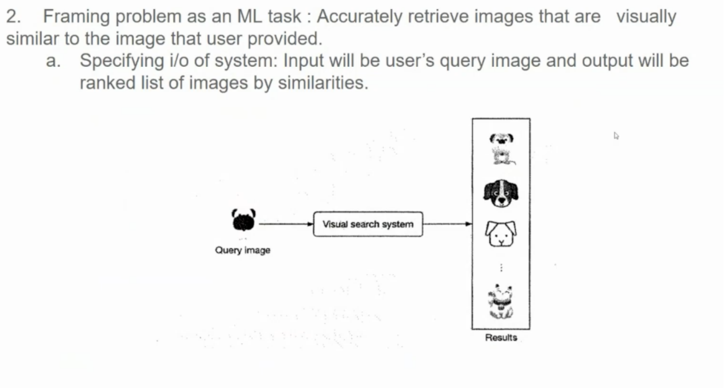
1/13/2015 Jure Leskovec, Stanford C246: Mining Massive Datasets 3



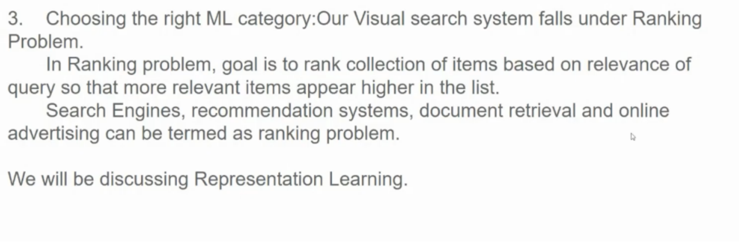
1. Clarifying Requirements: Visual Search System like pinterest which gives images similar to query image.

Fig. 2 Query image is a cup of tea and similar images are found by search based on images from Pinterest.

2. Framing problem as an ML task



2. Framing problem as an ML task : Accurately retrieve images that are visually similar to the image that user provided.
   a. Specifying i/o of system: Input will be user's query image and output will be ranked list of images by similarities.



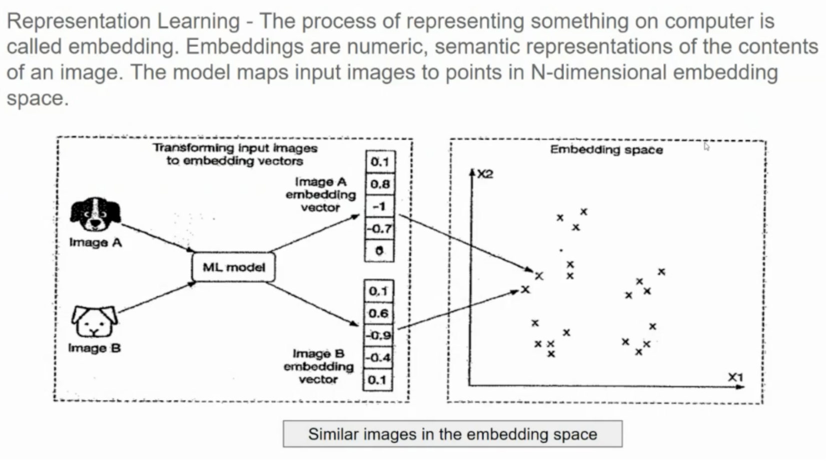
3. Choosing the right ML category:Our Visual search system falls under Ranking Problem.

In Ranking problem, goal is to rank collection of items based on relevance of query so that more relevant items appear higher in the list.

Search Engines, recommendation systems, document retrieval and online advertising can be termed as ranking problem.

We will be discussing Representation Learning.

Representation learning is a machine learning approach focused on automatically discovering and organizing features or representations from raw data. Instead of manually designing features, the model learns how to represent data in ways that make it easier for algorithms to perform tasks like classification, clustering, or prediction. Deep learning, such as neural networks, is commonly used for this as it can learn hierarchical and complex features directly from the data. This approach is widely used in image processing, NLP, and recommendation systems.



Representation Learning - The process of representing something on computer is called embedding. Embeddings are numeric, semantic representations of the contents of an image. The model maps input images to points in N-dimensional embedding space.

Similar images in the embedding space

An embedding is a way of converting items (like words, products, or even people) into numbers, making it easier for computers to understand and work with them. It's like creating a unique "location" for each item in a large "map" of numbers, where similar items are placed closer together. This helps computers recognize relationships, such as how "apple" is similar to "banana" because both are fruits, even though they're represented by different numbers. Embeddings are essential in tasks like recommendation systems, search engines, and language translation because they help algorithms find and understand patterns.

Embeddings are numerical representations of items like words, images, or products, mapped into a multi-dimensional "embedding space." In this space, each embedding (a vector of numbers) represents an item, and the distance between these embeddings reflects how similar or different the items are.

For example, suppose we have an image embedding model. When we process images of, say, dogs and cats, each image is transformed into an embedding in this space. If two images show similar things—like two different photos of a golden retriever—their embeddings will be close to each other in the space. Images of cats, on the other hand, will form a separate cluster, closer to other cats.

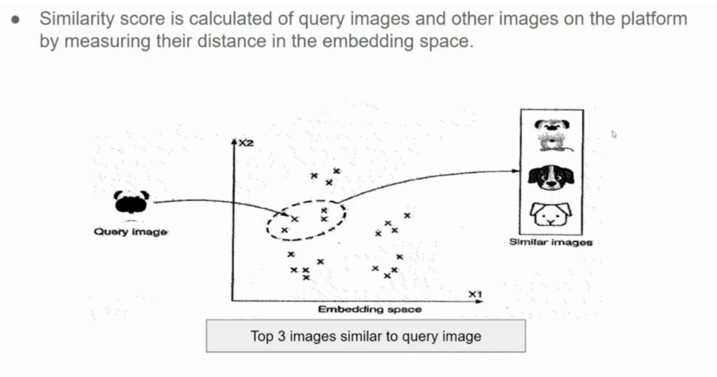
Finding Similar Images Using Embedding Space

To find images similar to a given one, we can calculate the "distance" between the embedding of the query image and the embeddings of other images in the dataset. A common way to do this is by measuring cosine similarity or Euclidean distance.

For example:

1. **Query Image**: You input a photo of a golden retriever.
2. **Embedding Search**: The model transforms this image into an embedding vector.
3. **Compare Distances**: The model finds other embeddings in the dataset that are closest to this vector.
4. **Results**: Images of other dogs, especially golden retrievers, appear as the most similar matches.

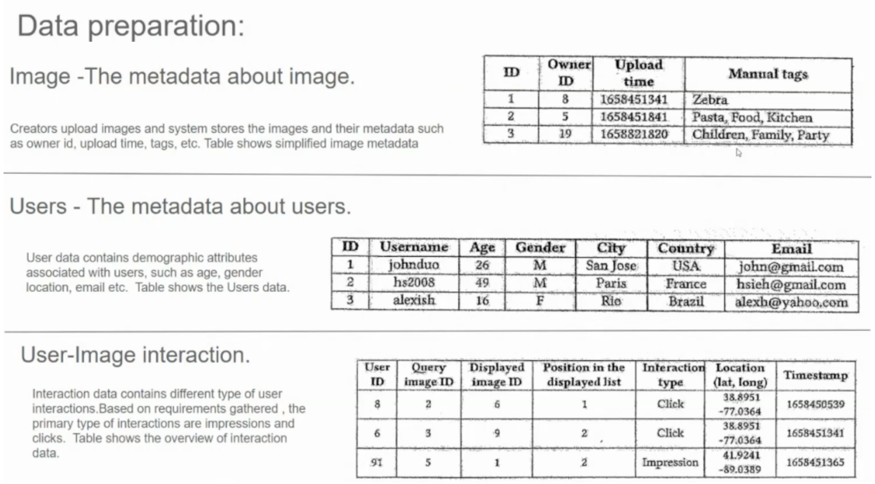
This method is used in applications like Google Images, where you can search by image, and it finds visually similar images.



- Similarity score is calculated of query images and other images on the platform by measuring their distance in the embedding space.

Top 3 images similar to query image

3. Data



Data preparation:

Image -The metadata about image.

Creators upload images and system stores the images and their metadata such as owner id, upload time, tags, etc. Table shows simplified image metadata

ID	Owner ID	Upload time	Manual tags
1	8	1658451341	Zebra
2	5	1658451841	Pasta, Food, Kitchen
3	19	1658821820	Children, Family, Party

Users - The metadata about users.

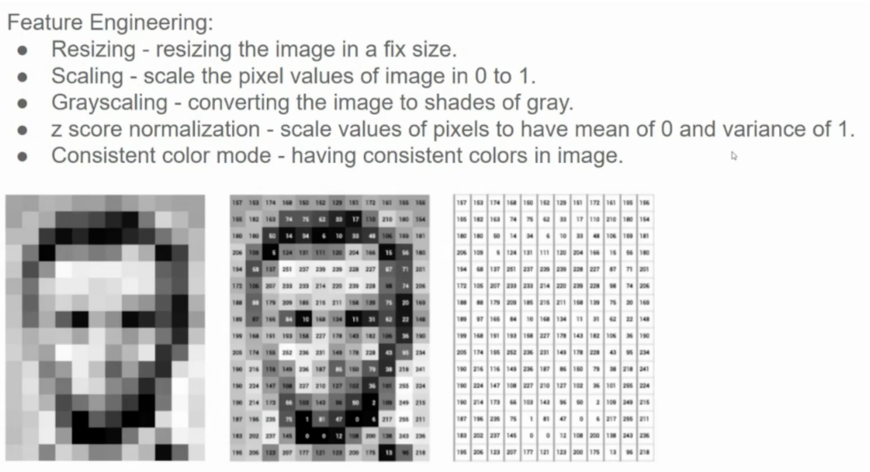
User data contains demographic attributes associated with users, such as age, gender location, email etc. Table shows the Users data.

ID	Username	Age	Gender	City	Country	Email
1	johnduo	26	M	San Jose	USA	john@gmail.com
2	hs2008	49	M	Paris	France	hsieh@gmail.com
3	alexish	16	F	Rio	Brazil	alexh@yahoo.com

User-Image interaction.

Interaction data contains different type of user interactions.Based on requirements gathered , the primary type of interactions are impressions and clicks. Table shows the overview of interaction data.

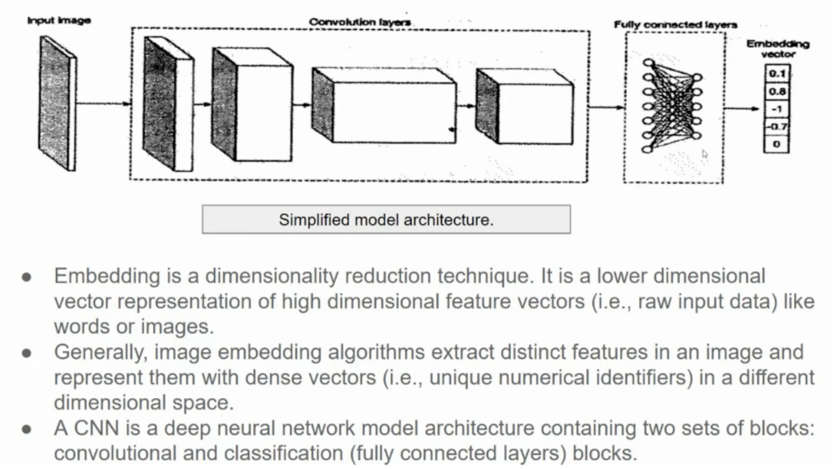
User ID	Query image ID	Displayed image ID	Position in the displayed list	Interaction type	Location (lat, long)	Timestamp
8	2	6	1	Click	38.8951 -77.0364	1658450539
6	3	9	2	Click	38.8951 -77.0364	1658451341
91	5	1	2	Impression	41.9241 -89.0389	1658451365



Feature Engineering:

- Resizing - resizing the image in a fix size.
- Scaling - scale the pixel values of image in 0 to 1.
- Grayscaling - converting the image to shades of gray.
- z score normalization - scale values of pixels to have mean of 0 and variance of 1.
- Consistent color mode - having consistent colors in image.

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218



Simplified model architecture.

- Embedding is a dimensionality reduction technique. It is a lower dimensional vector representation of high dimensional feature vectors (i.e., raw input data) like words or images.
- Generally, image embedding algorithms extract distinct features in an image and represent them with dense vectors (i.e., unique numerical identifiers) in a different dimensional space.
- A CNN is a deep neural network model architecture containing two sets of blocks: convolutional and classification (fully connected layers) blocks.

Diagram Explanation

1. **Input Image**: The process starts with an input image, which is a high-dimensional array of pixel values.
2. **Convolutional Layers**:
   - The CNN applies multiple convolutional layers to the image. These layers detect various features like edges, textures, shapes, and complex patterns.
   - Convolutional layers progressively transform the raw pixel data into more abstract feature representations. Each layer builds on the features learned by the previous one.
3. **Fully Connected Layers**:
   - After the convolutional layers, the features extracted from the image are flattened and fed into fully connected (dense) layers.
   - These layers further combine and process the features to capture more complex relationships and representations, ultimately reducing the data to a lower-dimensional vector.
4. **Embedding Vector**:
   - The final output from the fully connected layers is an **embedding vector**, a dense, lower-dimensional representation of the original image.
   - This vector is a series of numbers (as shown, e.g., [0.1, 0.8, -1, 0.7, 0]) that uniquely represents the image in a more compact form, which is useful for various downstream tasks like image retrieval or similarity comparison.

Bullet Point Explanation

- **Embedding as Dimensionality Reduction**:
  - Embeddings are a **dimensionality reduction technique**. They transform high-dimensional data (such as raw images) into a lower-dimensional vector representation.
  - This is beneficial because it allows for the storage and processing of complex data in a compressed form without losing essential information.
- **Feature Extraction for Dense Vectors**:
  - The process of creating embeddings involves **extracting distinct features** from the input data, whether it's images or text.
  - For images, these features could include patterns, colors, shapes, or textures that uniquely define the content of the image.
  - The extracted features are then represented as a **dense vector** (embedding) in a multi-dimensional space, where similar images (in terms of content or visual features) are closer together.
- **CNN Architecture**:
  - A **CNN (Convolutional Neural Network)** is used here as the model architecture.
  - CNNs typically consist of two main types of layers:
    - **Convolutional layers** for feature extraction (these detect patterns within the data).
    - **Fully connected layers** (classification layers) to interpret and summarize these features into a lower-dimensional output.

Applications of Embedding Vectors

- **Similarity Search**: Embedding vectors allow similar images to be easily found. Since images with similar features have embeddings that are close in the embedding space, we can calculate distances (e.g., using cosine similarity) to find images that are visually similar to a query image.
- **Image Classification and Clustering**: The embedding vector can also be used to classify or cluster images. For example, images of cats and dogs will have distinct clusters in the embedding space, making it easy to identify which category a new image belongs to based on its embedding.