Image Retrieval

Jiayuan Gu

gujy1@shanghaitech.edu.cn

Modified based on Prof. Shenghua Gao’s slides

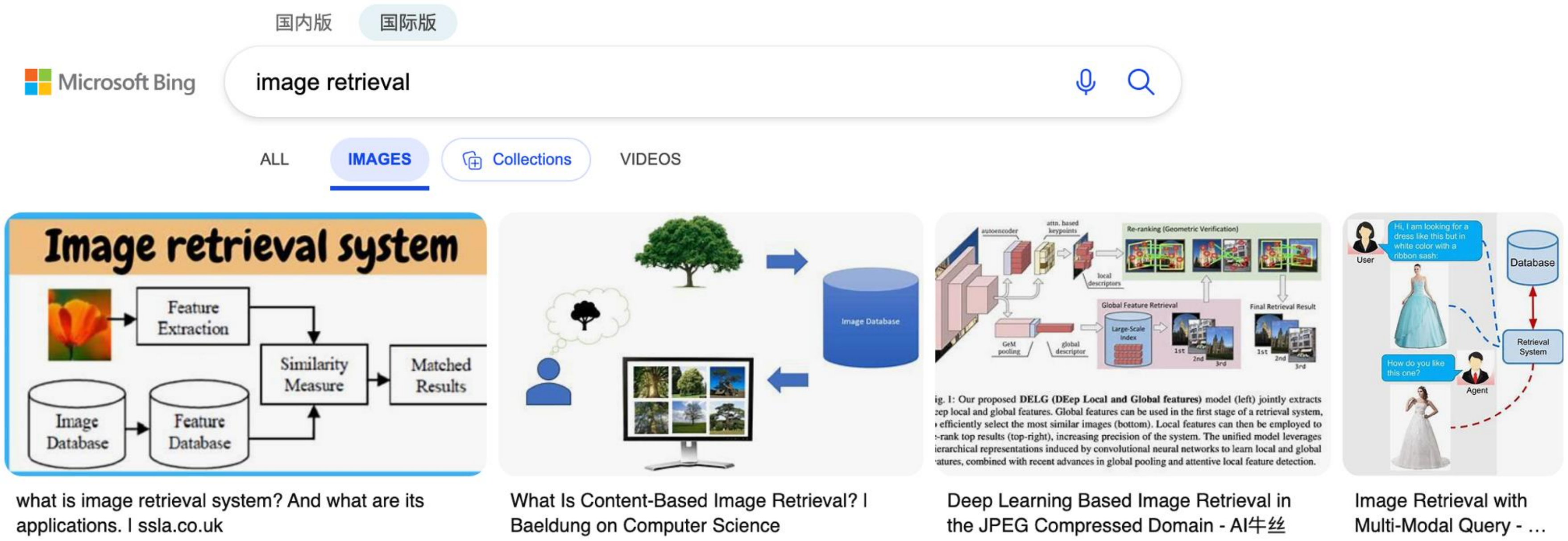
What is Image Retrieval?

- Wiki: *An image retrieval system is a computer system used for browsing, searching and retrieving **images** from a large database of digital images.*

Example of an image database and image retrieval:
iPhone (or Google photos) can help group your photos

Different Methods of Image Retrieval

- Image meta search: search of images based on associated metadata such as keywords, text, etc.
  - Labor is needed for manual annotation
  - Annotation can be inaccurate



Different Methods of Image Retrieval

- Description-based image retrieval (DBIR) : search of images based on associated metadata such as keywords, text, etc.

- Content-based image retrieval (CBIR): the application of computer vision to the image retrieval.
  - CBIR aims at **avoiding the use of textual descriptions** and instead retrieves images based on **similarities in their contents** (textures, colors, shapes etc.) to a user-supplied query image or user-specified image features.

Overview of general CBIR model

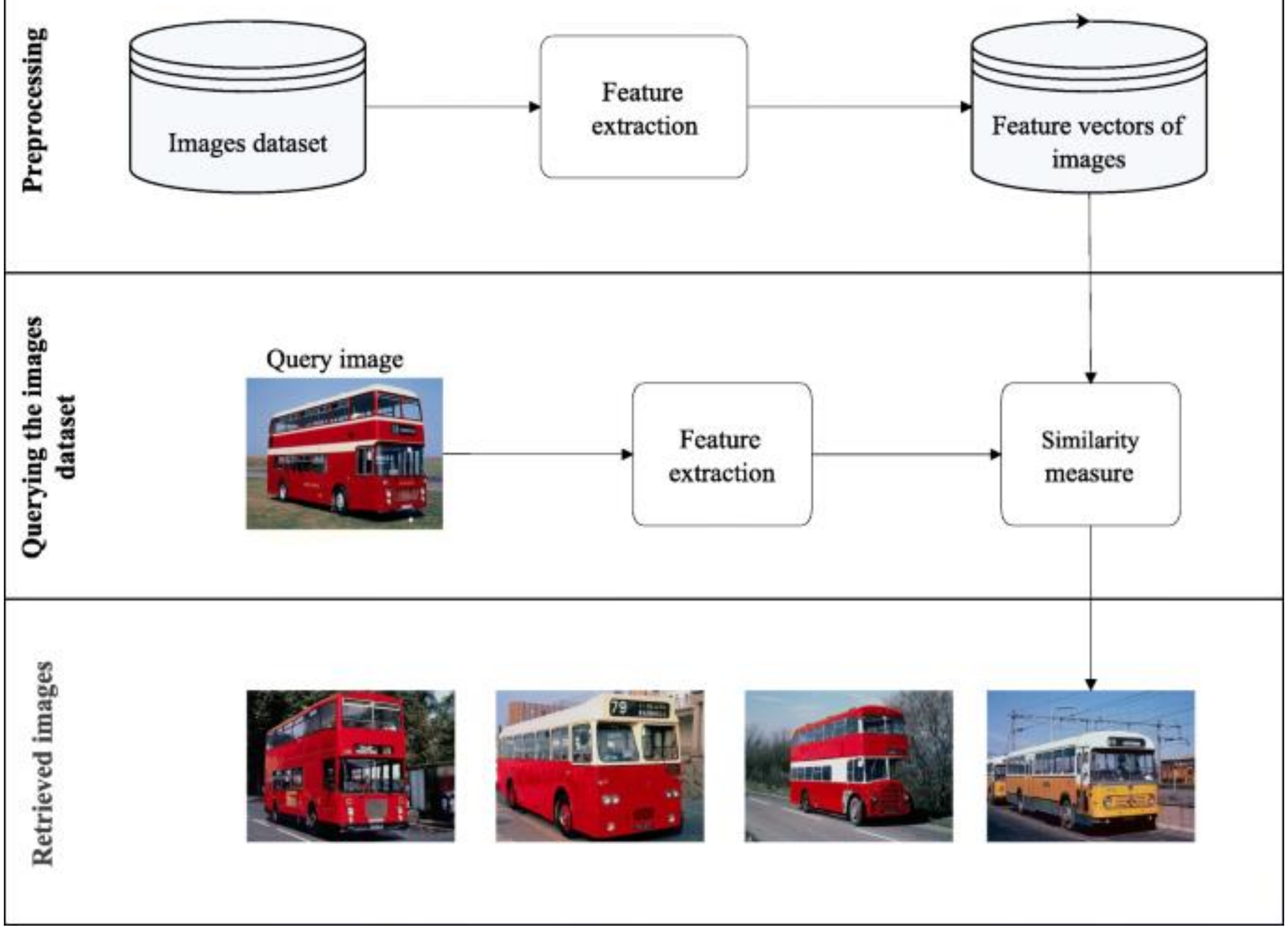
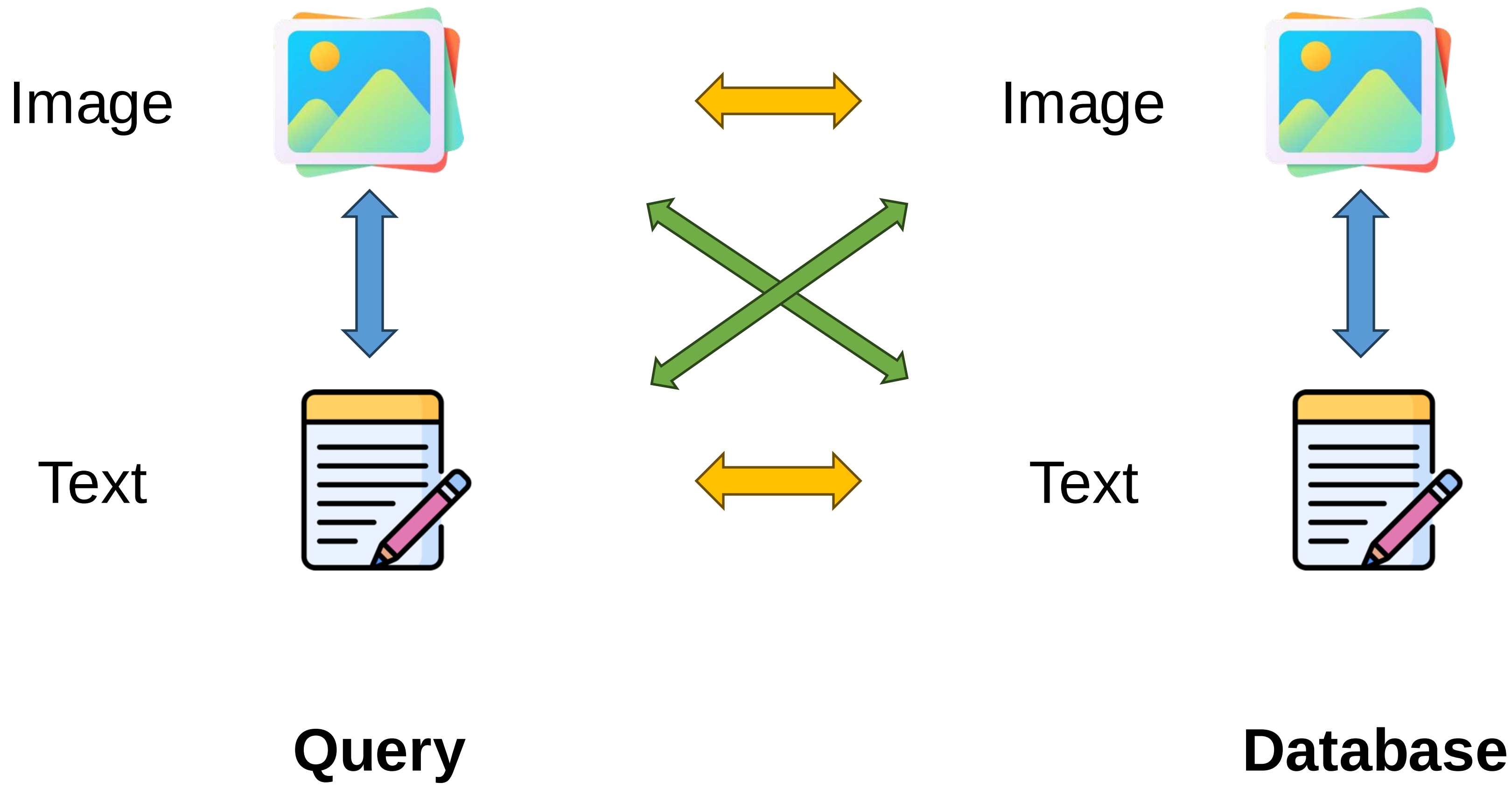


Figure from “An efficient bi-layer content based image retrieval system”

General Information Retrieval



Features for Retrieval

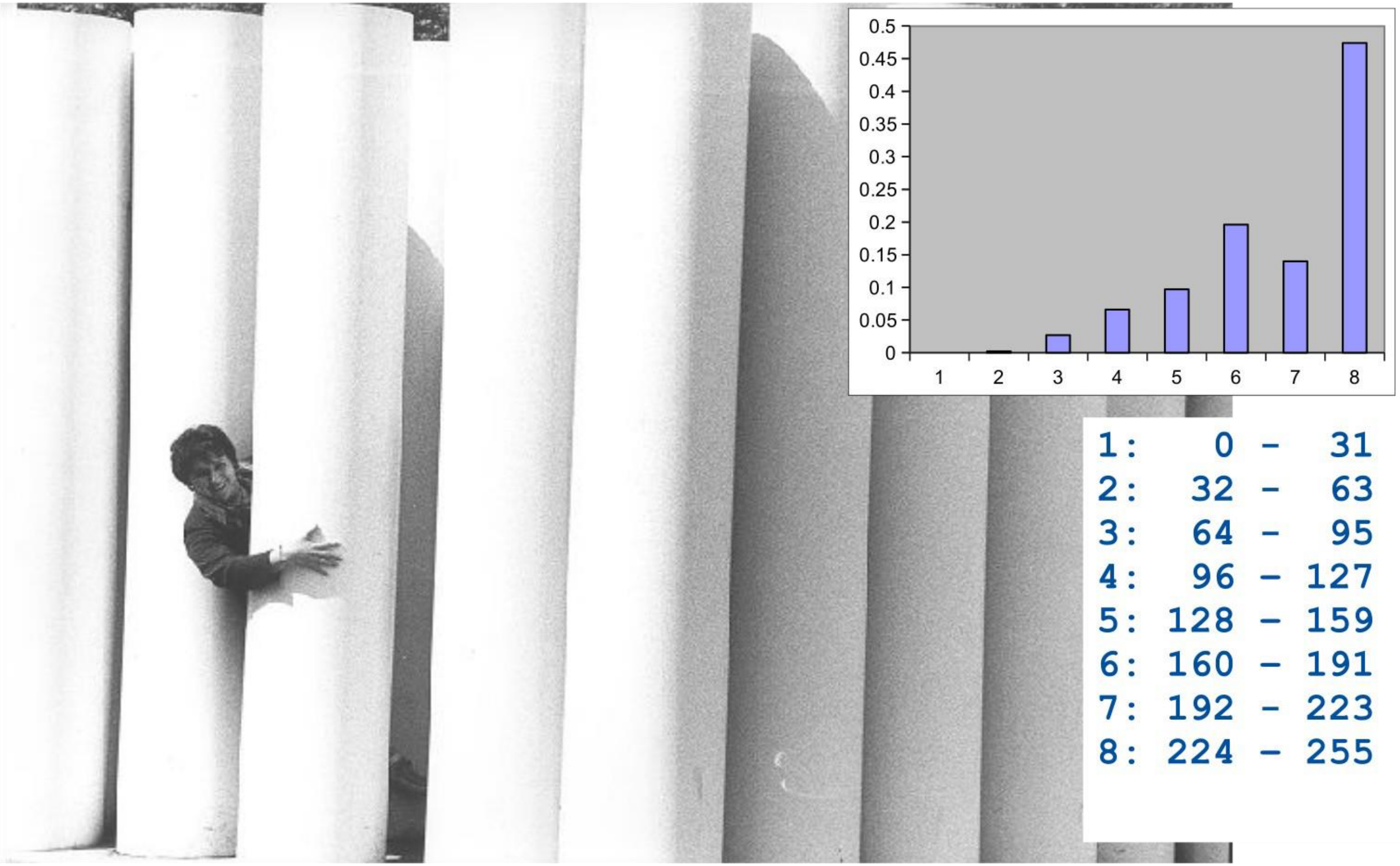
- What features are used for retrieving images?
- Color: mean, distribution, relative locations
- Shape: segmented objects, sketches
- Others, like texture

Feature: Color Histograms

- An early similarity measurement uses color histograms
  - The RGB (or other color space) is discretized into bins
  - For each bin, a count is maintained on the number of pixels that fall into the bin

- Once constructed, the histograms can be compared using several metrics

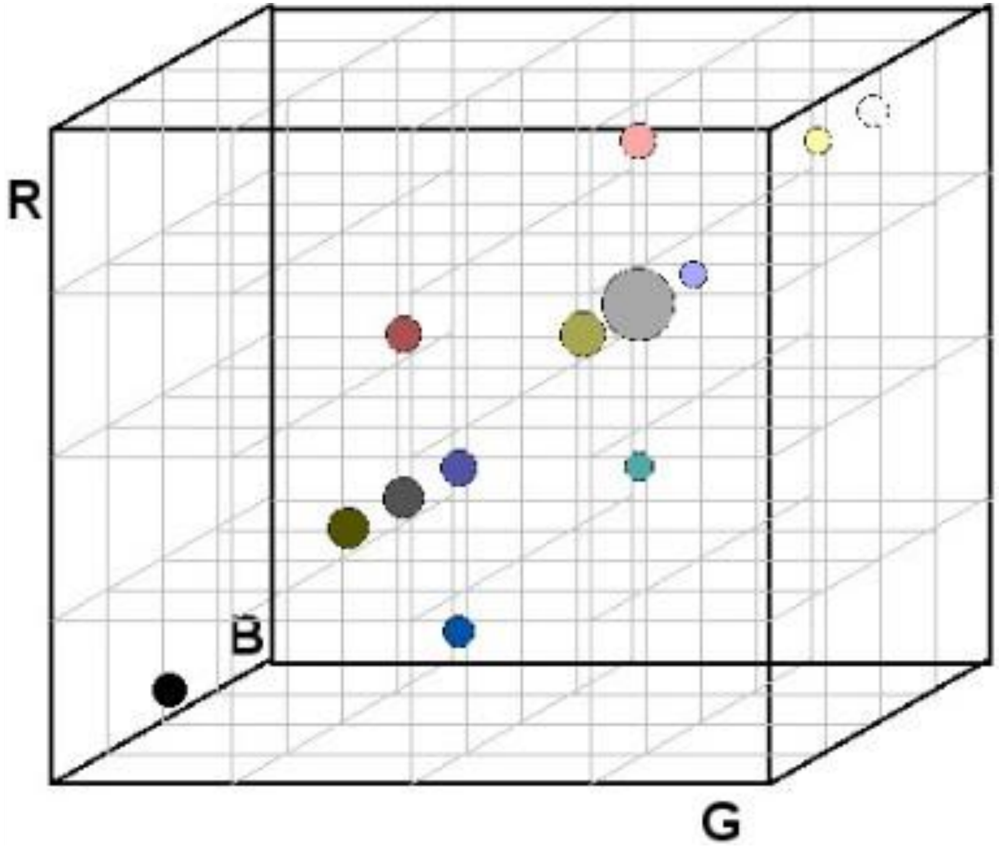
Grayscale Histograms



Rueger, “Multimedia Information Retrieval” Lecture 5
www.nii.ac.jp/userimg/lectures/20120319/Lecture5.pdf

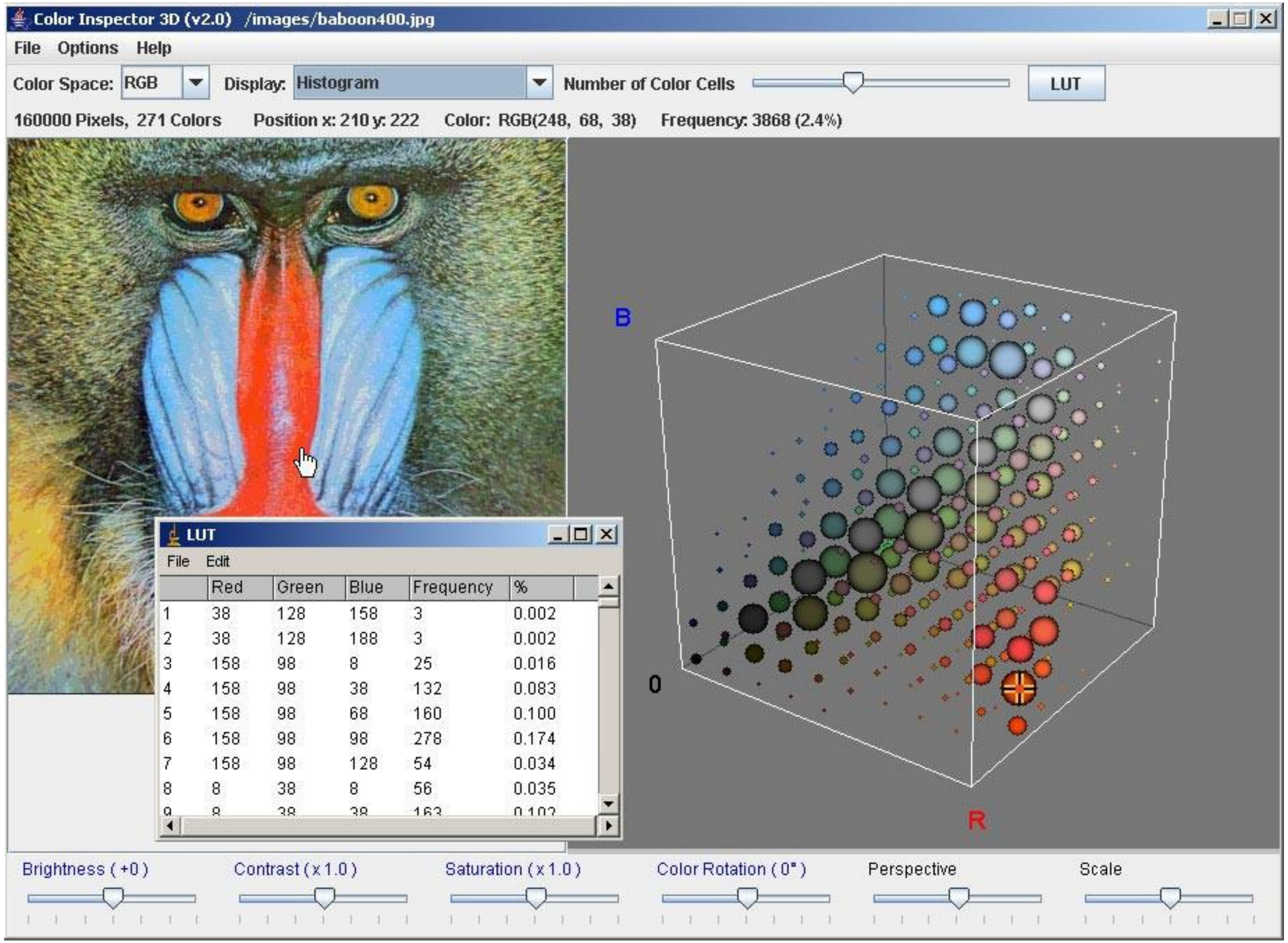
RGB Histograms

→



Rueger, “Multimedia Information Retrieval” Lecture 5
www.nii.ac.jp/userimg/lectures/20120319/Lecture5.pdf

RGB Histograms



https://imagej.net/

Query by image and video content: the QBIC system

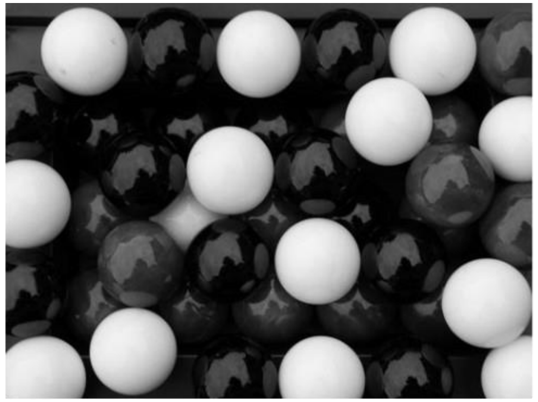
- The QBIC system developed by IBM was the first commercial system for image-based content retrieval
- It uses color, texture, shape, location, and keywords

Example: QBIC Search by Color

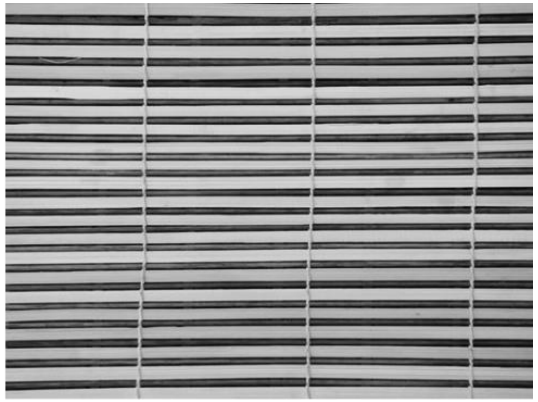
Tamura Texture Features

- Texture is a property of image regions, not pixels
- Perceptual experiments yielded a small set of descriptors that capture how people see texture

coarseness



contrast

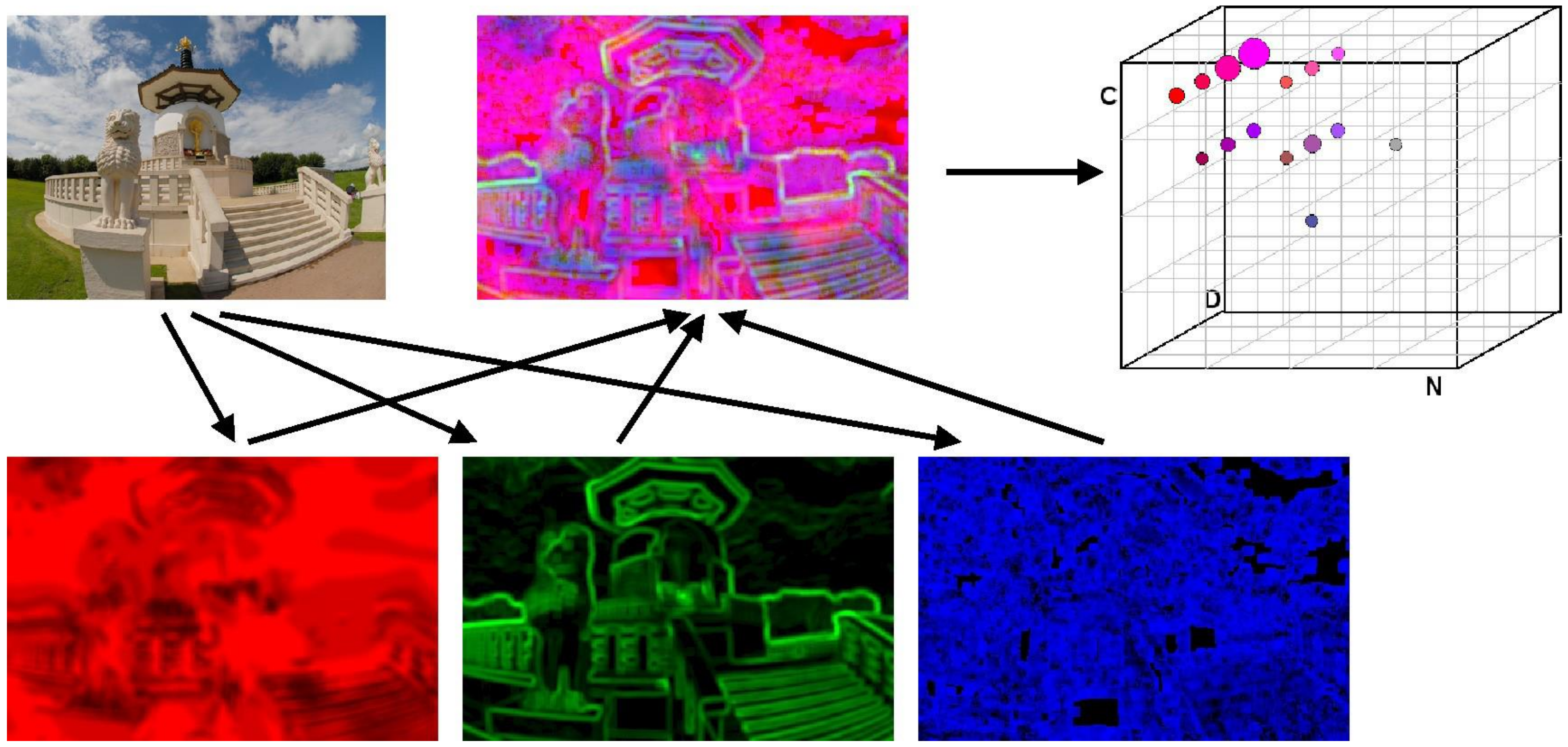


directionality

Rueger, “Multimedia Information Retrieval” Lecture 5
www.nii.ac.jp/userimg/lectures/20120319/Lecture5.pdf

Search by Texture

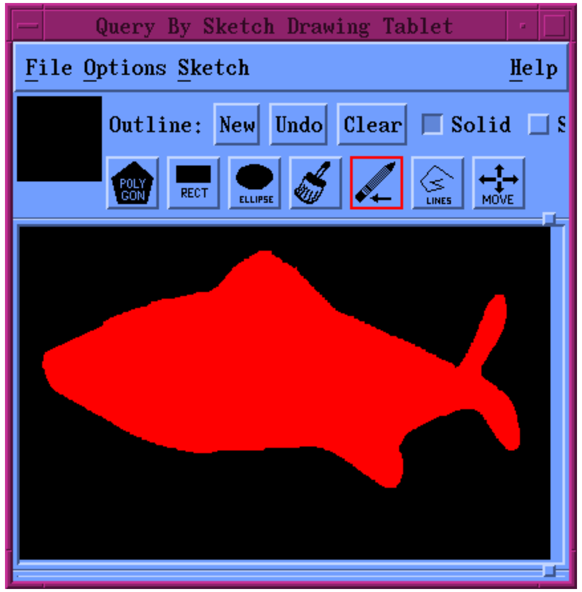
Create 3D histogram like color histogram



Coarseness Contrast Directionaity

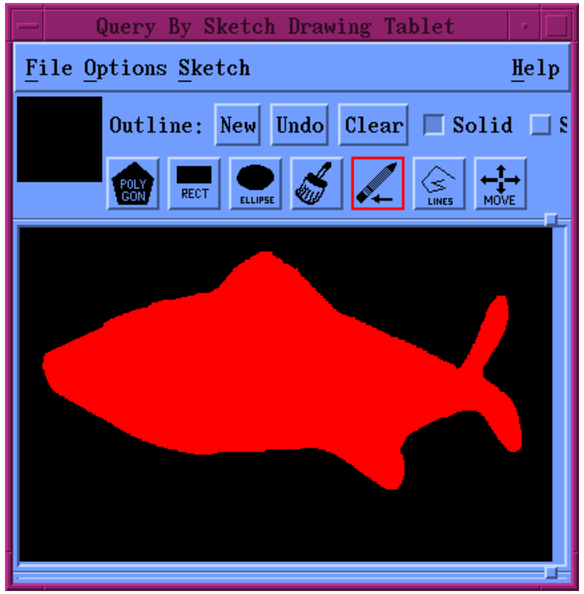
Rueger, “Multimedia Information Retrieval” Lecture 5
www.nii.ac.jp/userimg/lectures/20120319/Lecture5.pdf

Example: QBIC Search by Shape

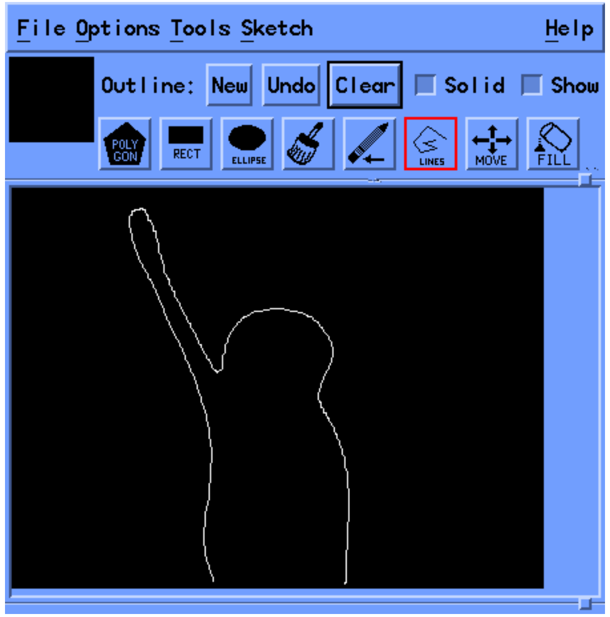


Shape Features

- Boundary length
- Area enclosed
- Boundary curvature (overall or histogram)
- Moments
- Projections onto axes
- Tangent angle histogram



Example: OBIC Search by Sketch



Canny edeg operator is used to compute features

Region-Based Image Retrieval

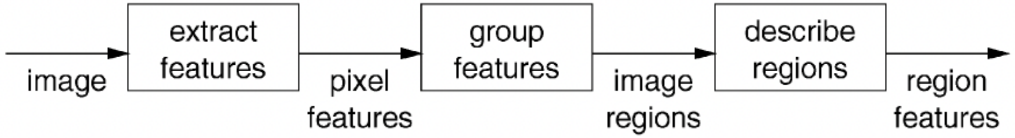
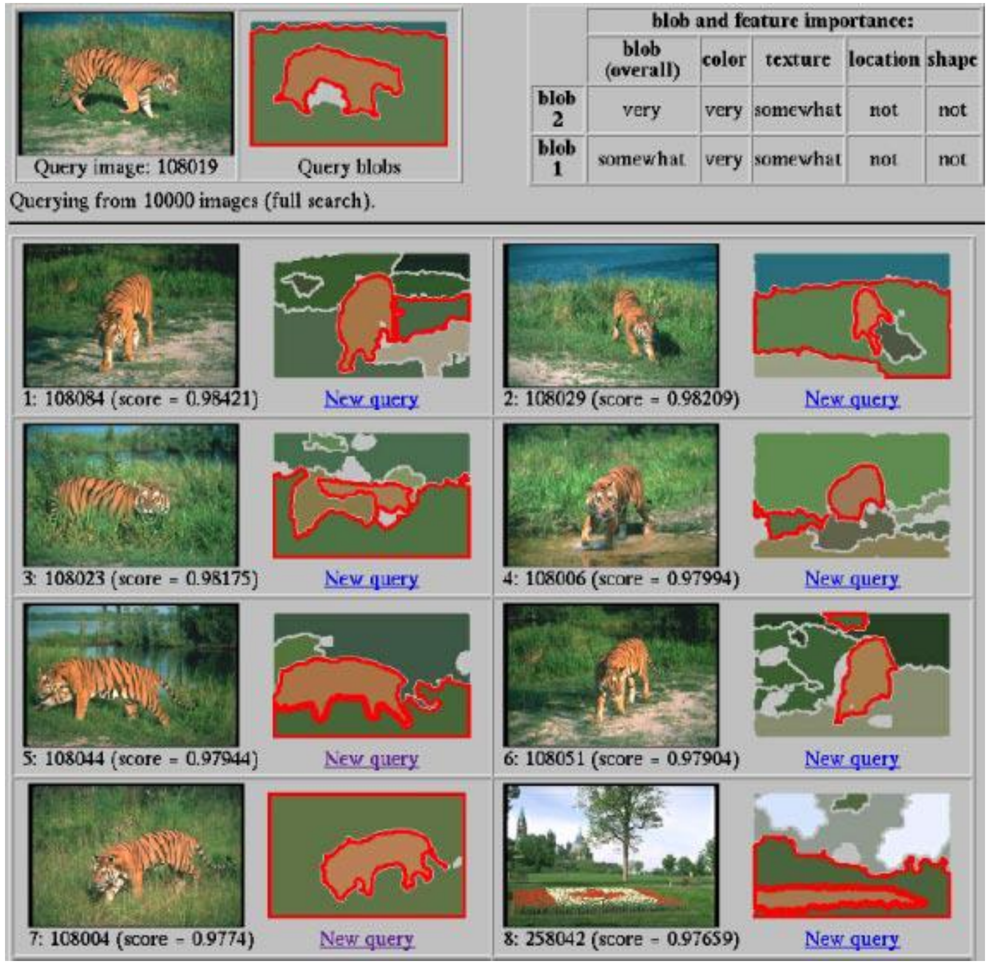


Image segmentation first

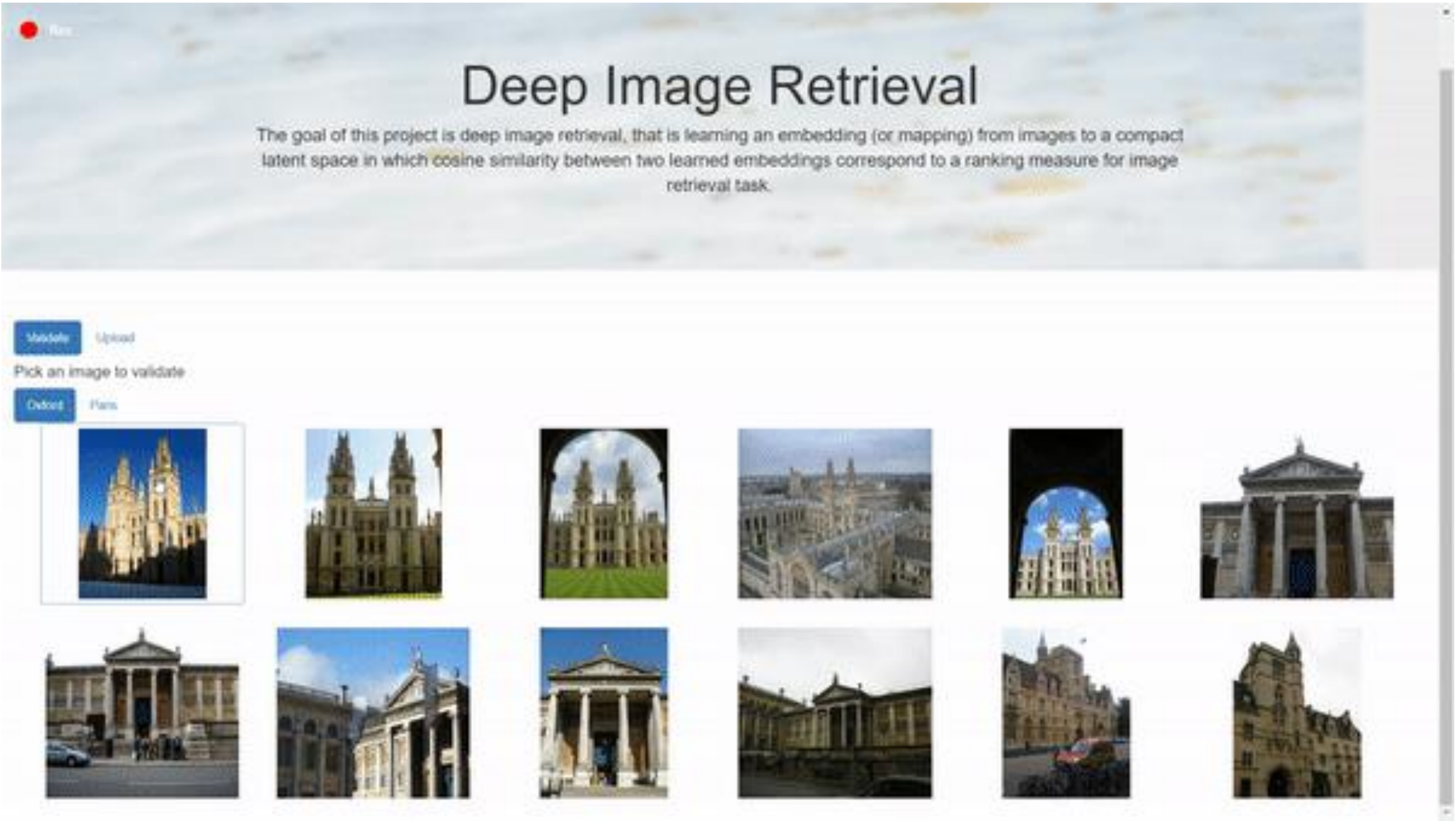
The feature similarity is computed over regions

Support logic operations, e.g., “like-blob-1 and (like blob-2 or like-blob-3)”



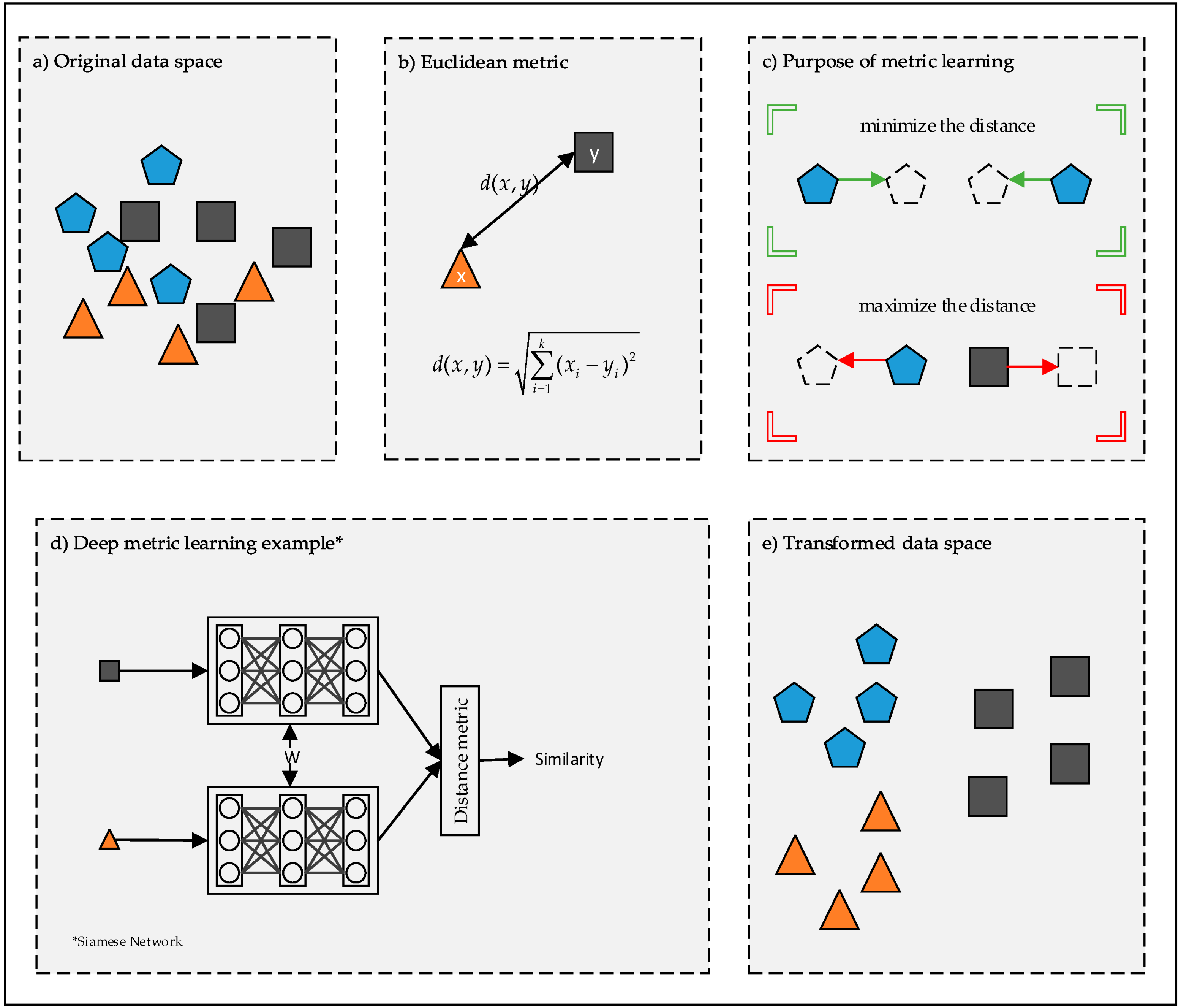
Blobworld: A System for Region-Based Image Indexing and Retrieval

Deep-Feature-Based Retrieval



learning an embedding (or mapping) from images to a compact latent space in which (cosine) similarity between two learned embeddings correspond to a ranking measure for image retrieval task

https://github.com/keshik6/deep-image-retrieval



KAYA M, BİLGE HŞ. Deep Metric Learning: A Survey. Symmetry. 2019; 11(9):1066.

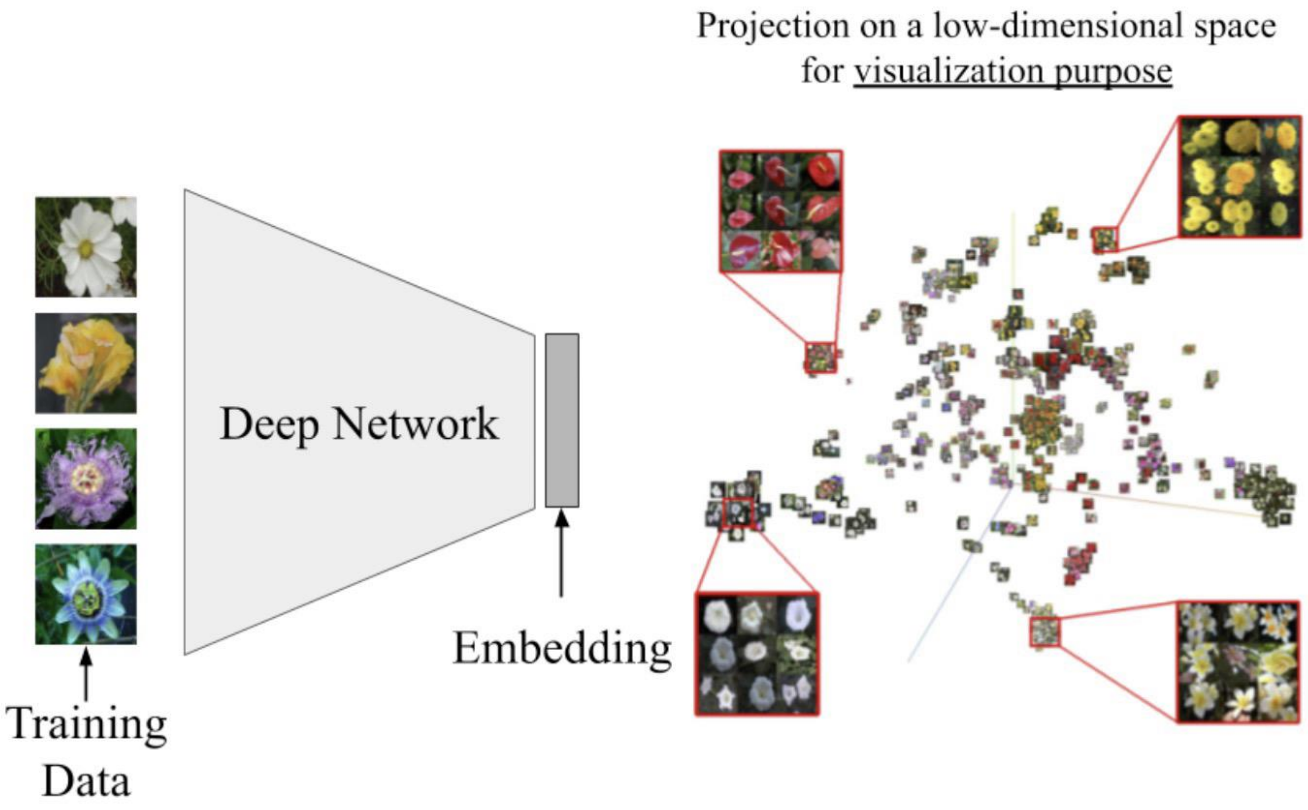
Metric Learning

- Distance metric learning (or simply, metric learning) aims at automatically constructing task-specific **distance metrics** from (weakly) supervised data, in a machine learning manner.

- The metric learning problem is generally formulated as an optimization problem where one seeks to **find the parameters of a distance function** that optimize some objective function measuring the agreement with the training data.

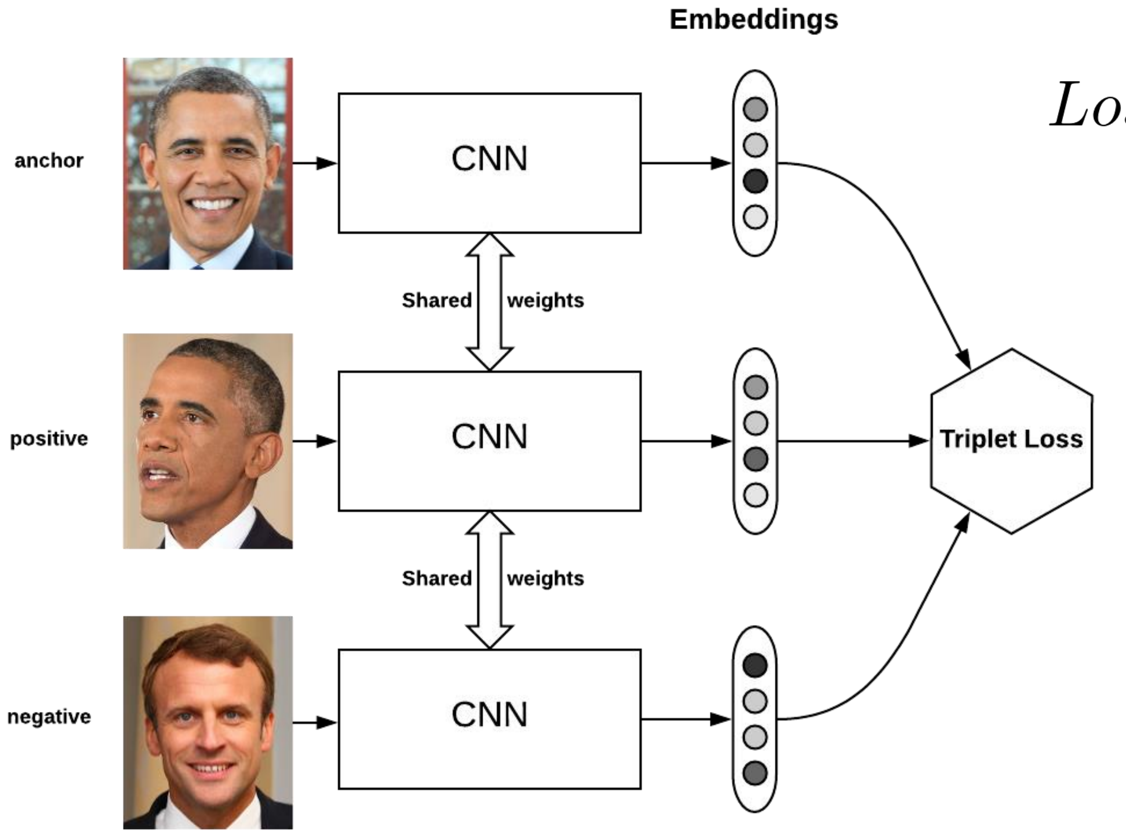
https://contrib.scikit-learn.org/metric-learn/introduction.html

Learning a Good Embedding Space

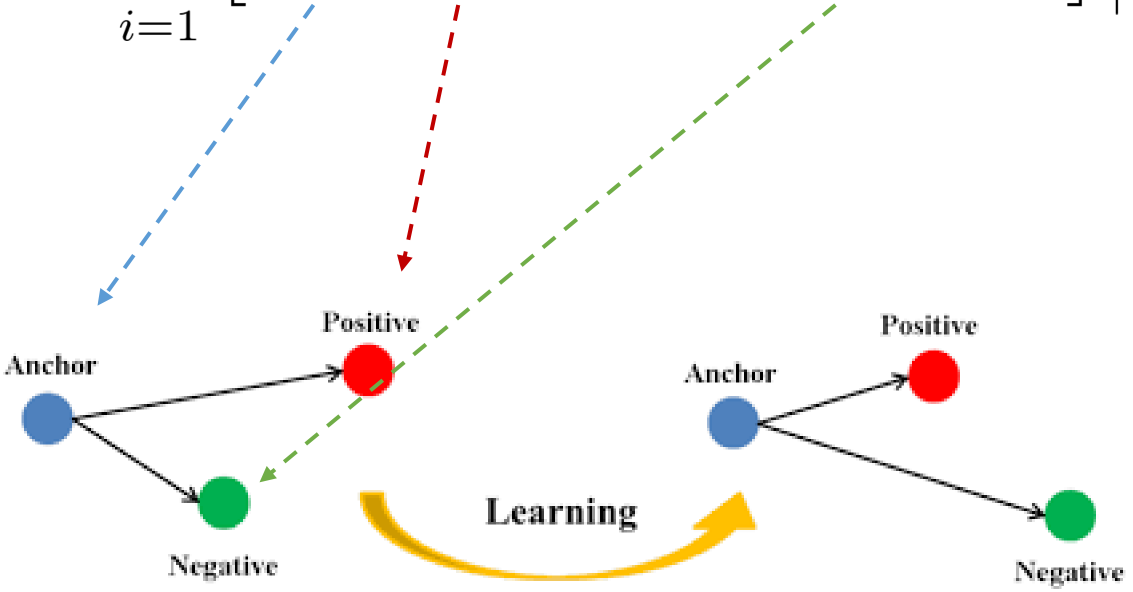


A deep network trained with a ranking loss to enable searching and indexing.

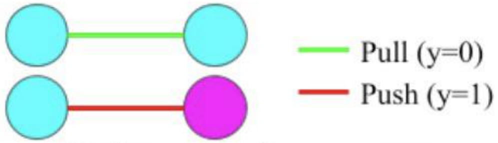
Triplet Loss



$$Loss = \sum_{i=1}^{N} \left[\| f_i^a - f_i^p \|_2^2 - \| f_i^a - f_i^n \|_2^2 + \alpha \right]_+$$



Contrastive Loss



$$L_{\text{contrastive}} = (1-y) \times D(x_i, x_j) + y \times \max(0, m - D(x_i, x_j))$$

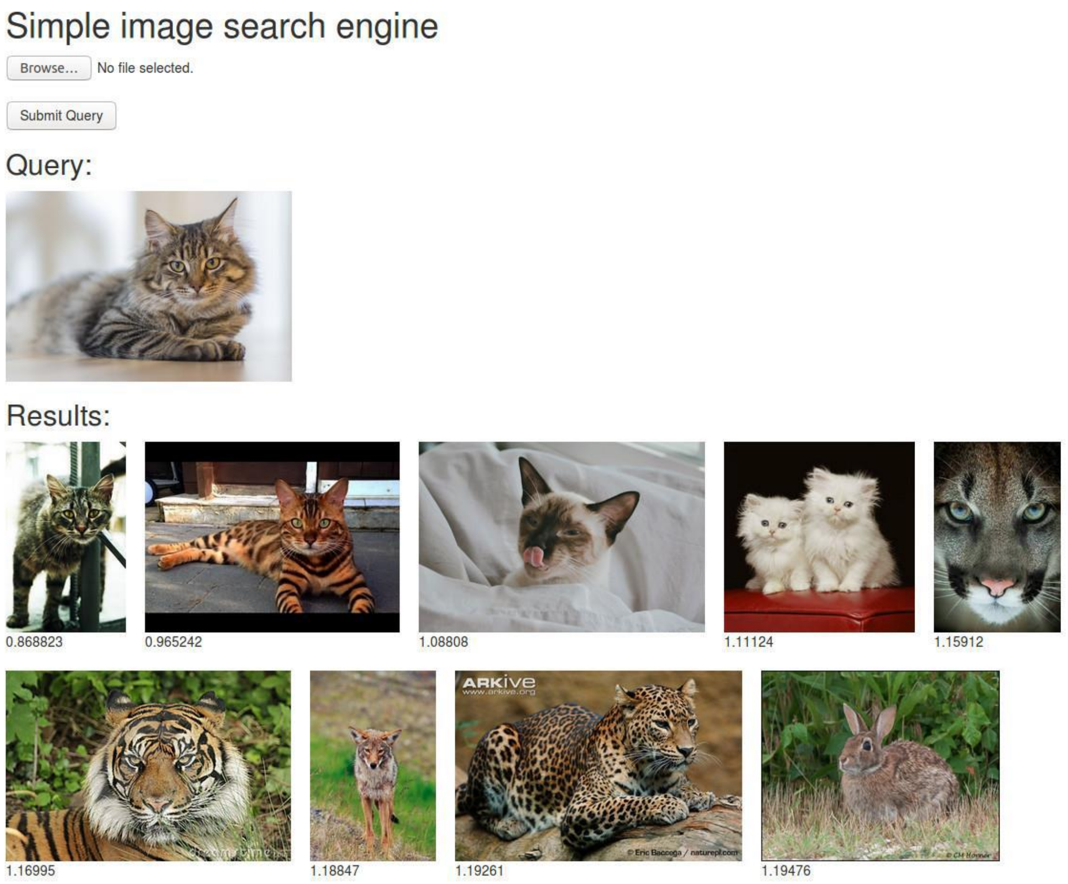
Contrastive Loss formulation.

$$L_{\text{triplet}} = max(0, D(a,p) - D(a,n) + m)$$

Triplet Loss formulation

Example: Simple-Image-Search

- Demo: http://www.simple-image-search.xyz/
  - Based on deep feature
  - Web interface
  - Pure python



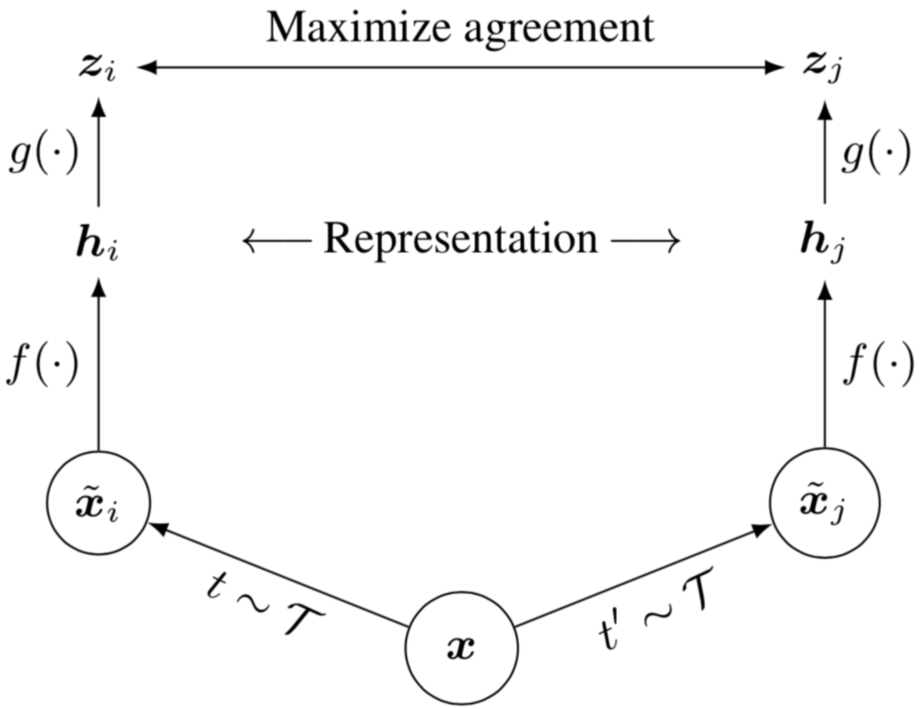
Case Study: CVPR 2020 Tutorial Image Retrieval in the Wild

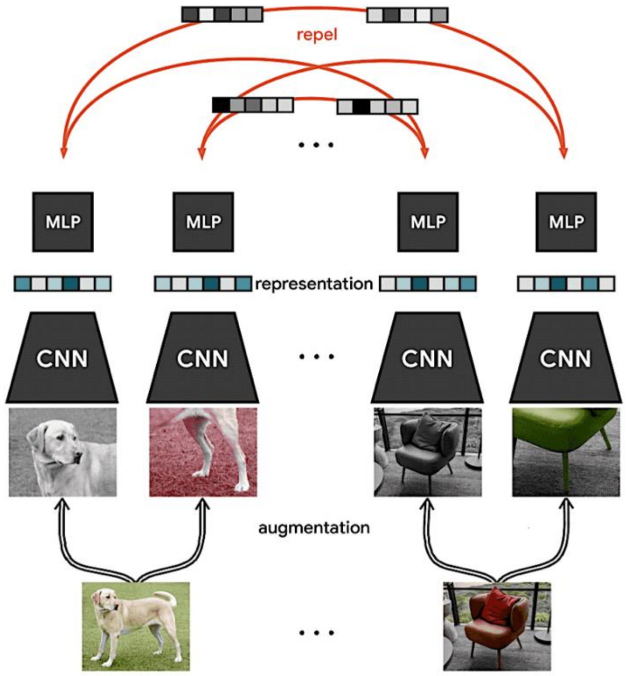
Advanced Topics

Self-supervised learning for feature representation

Visual-language model (VLM)

Self-Supervised Feature Learning





A Simple Framework for Contrastive Learning of Visual Representations

Case Study: SimCLR

1. Self-supervised Formulation [Data Augmentation]

First, we generate batches of size N from the raw images. Let's take a batch of size N = 2 for simplicity. In the paper, they use a large batch size of 8192.

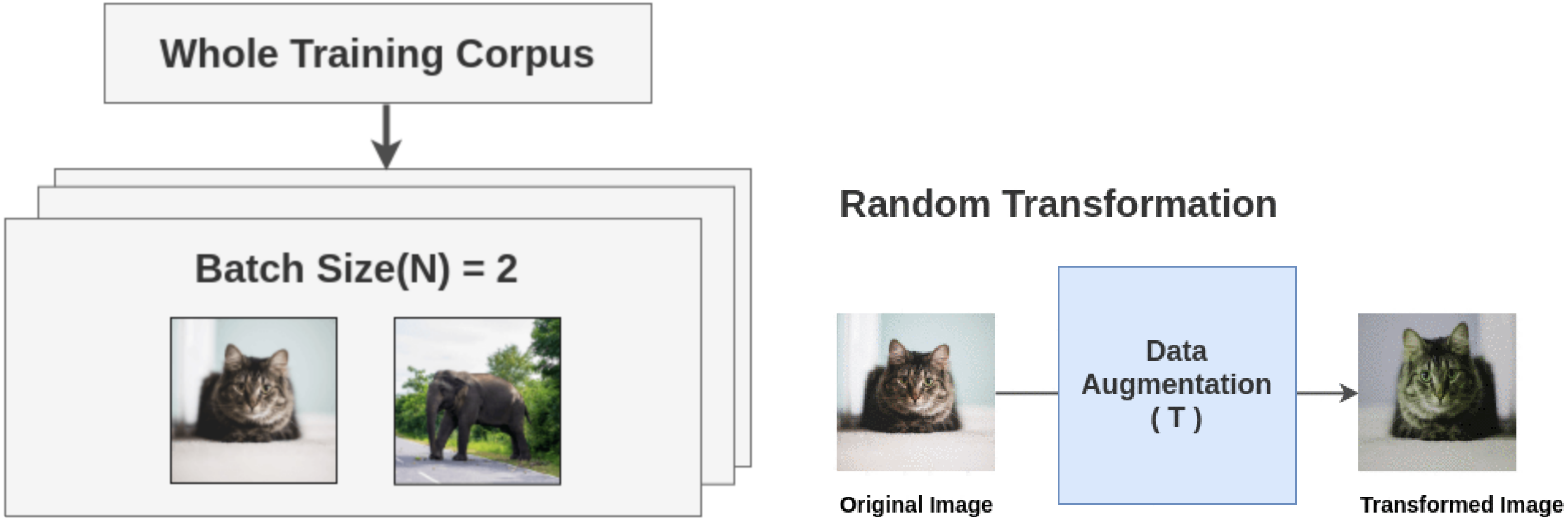


Figure from blog

Case Study: SimCLR

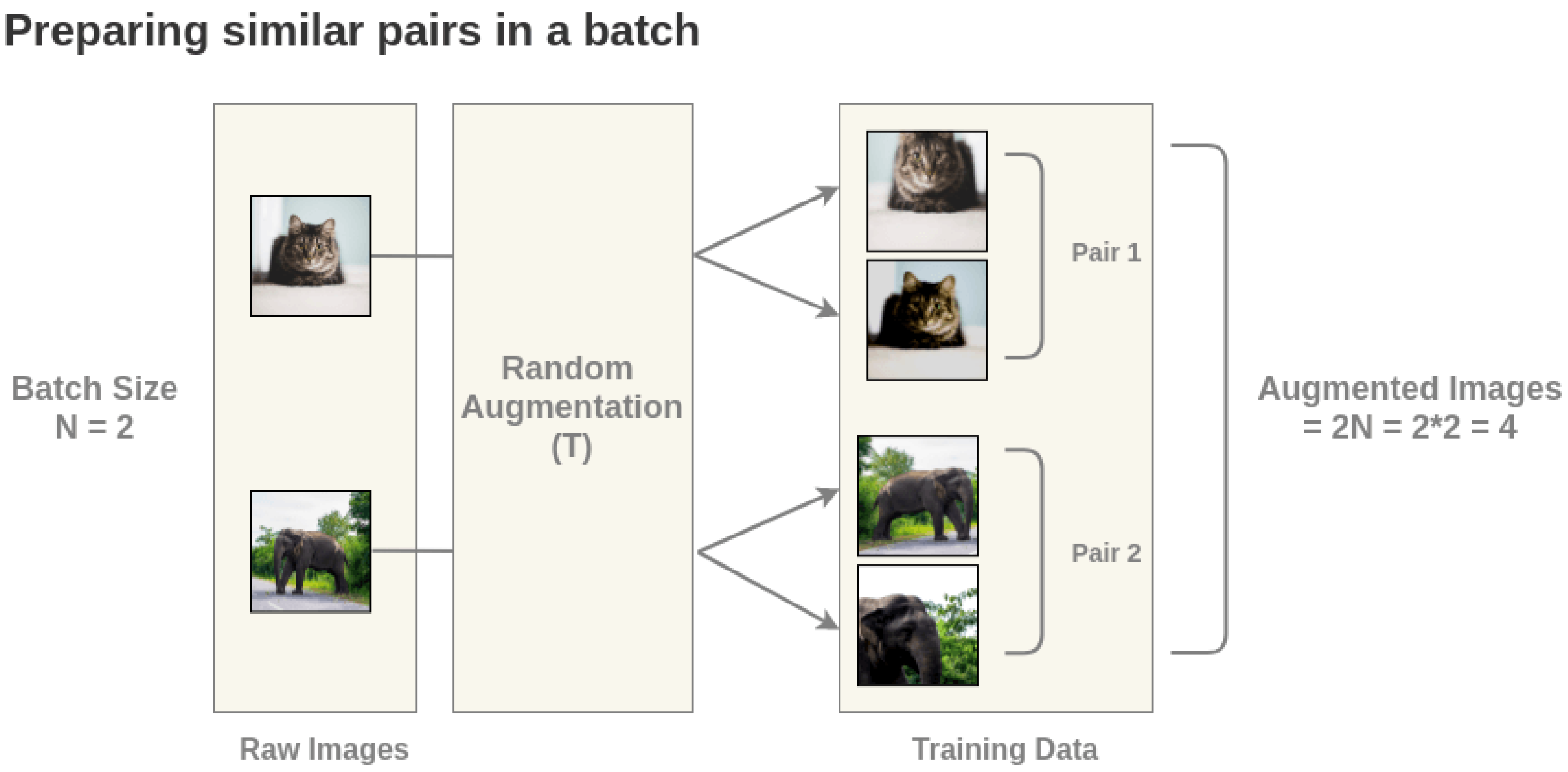


Figure from blog

Case Study: SimCLR

2. Getting Representations [Base Encoder]

Each augmented image in a pair is passed through an encoder to get image representations. The encoder used is generic and replaceable with other architectures. The two encoders shown below have shared weights and we get vectors h_i and h_j.

Encoder Component of Framework

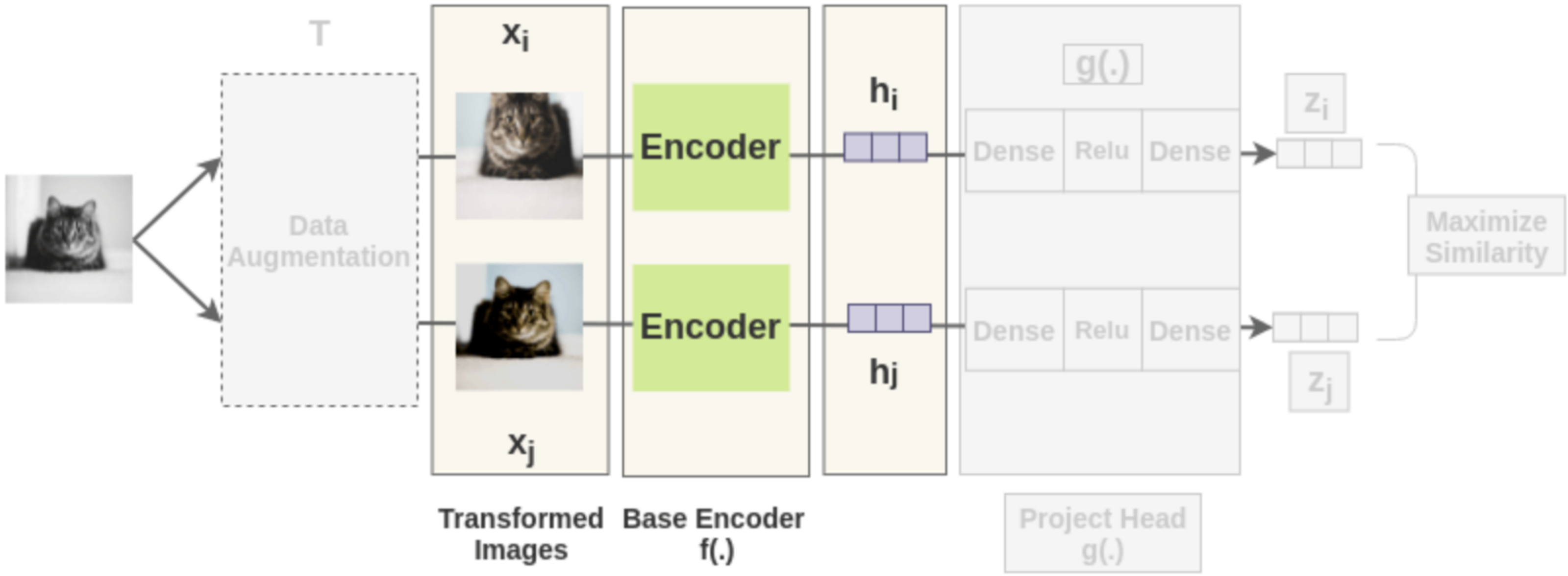


Figure from blog

Case Study: SimCLR

In the paper, the authors used ResNet-50 architecture as the ConvNet encoder. The output is a 2048-dimensional vector h.

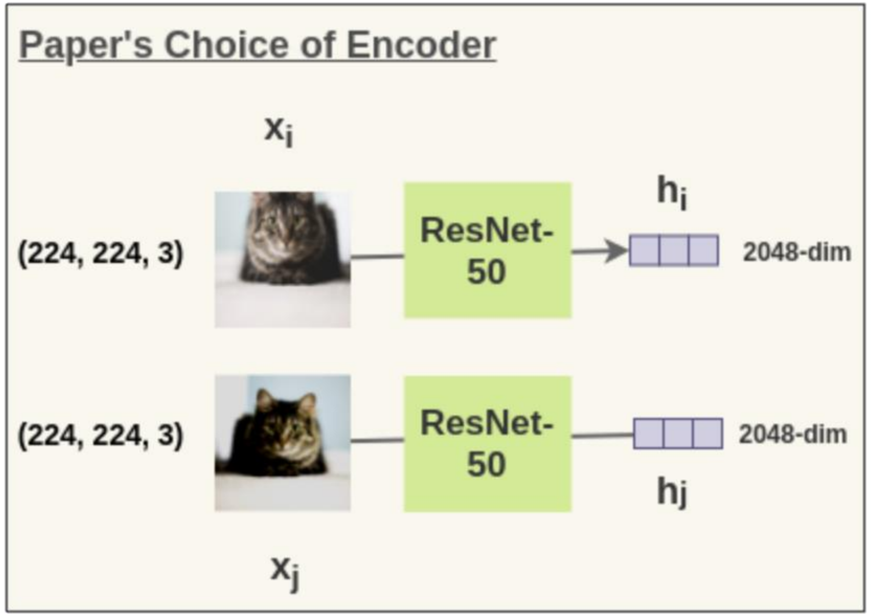


Figure from blog

Case Study: SimCLR

3. Projection Head

The representations h_i and h_j of the two augmented images are then passed through a series of non-linear **Dense → Relu → Dense** layers to apply non-linear transformation and project it into a representation z_i and z_j. This is denoted by $g(.)$ in the paper and called projection head.

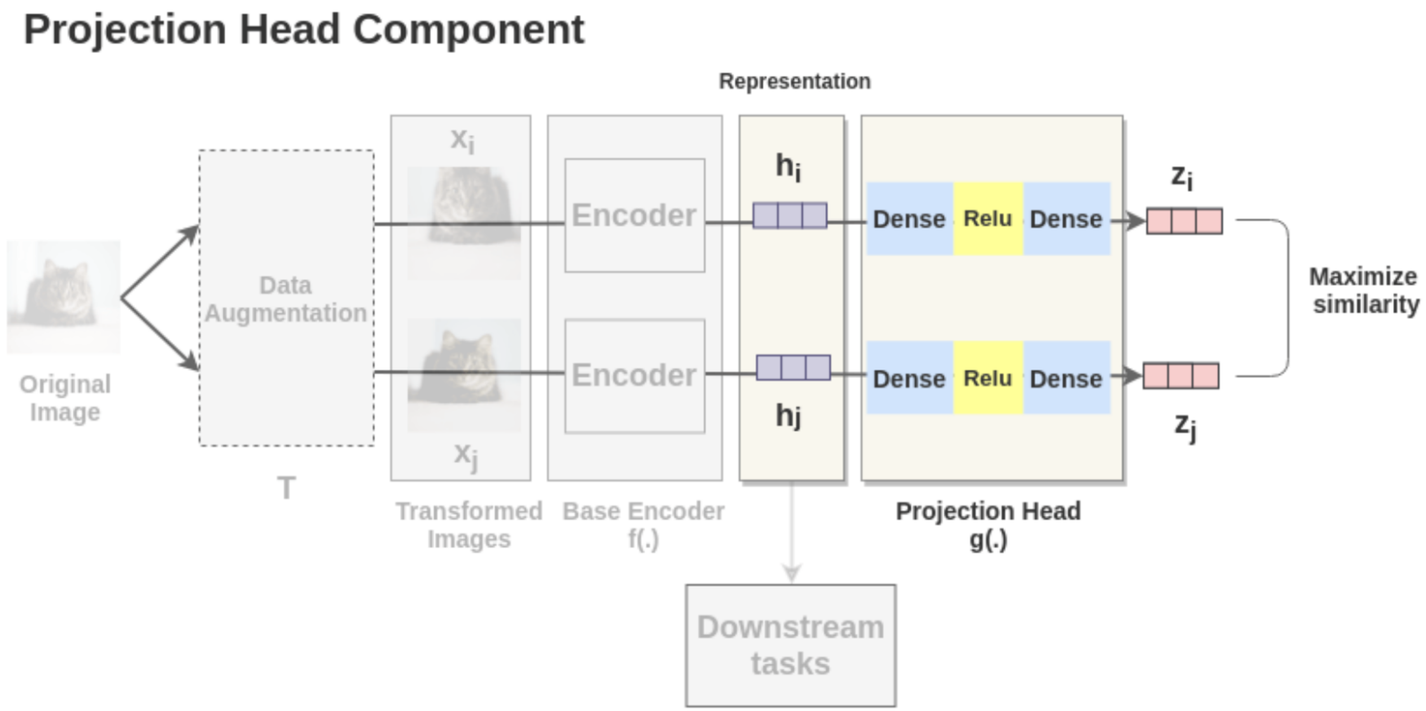


Figure from blog

Case Study: SimCLR

Training model: Bring similar closer

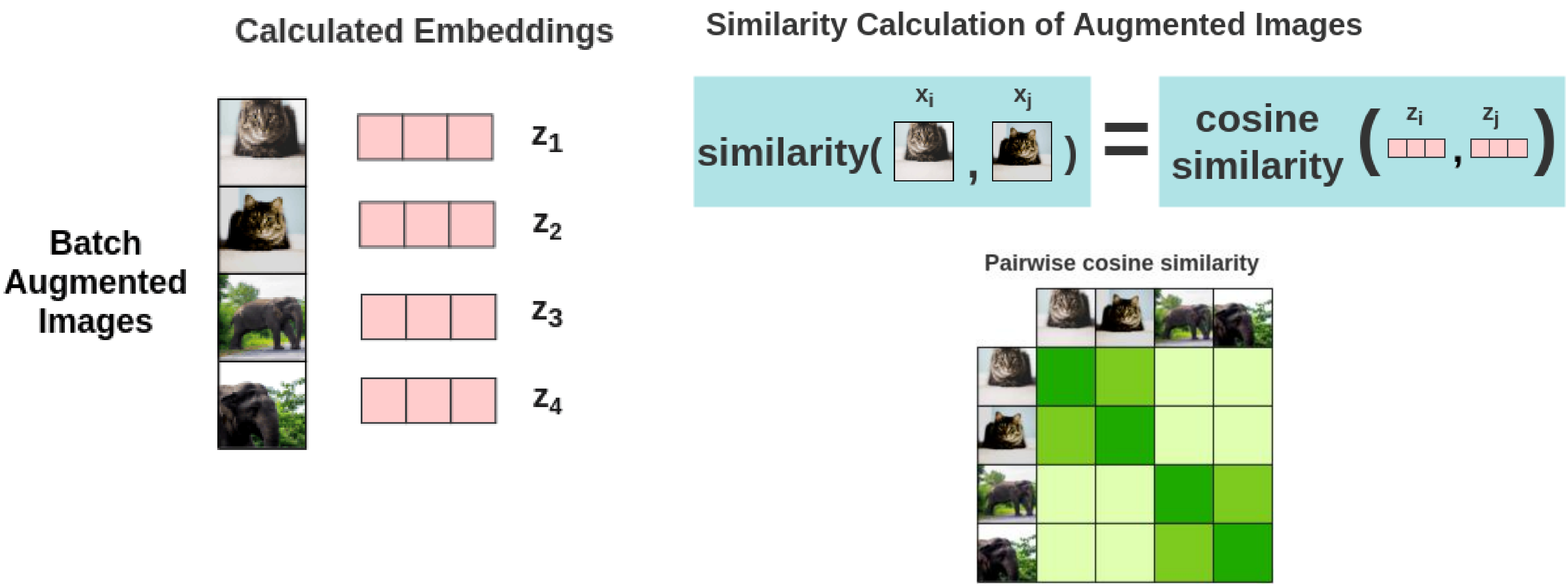


Figure from blog

Case Study: SimCLR

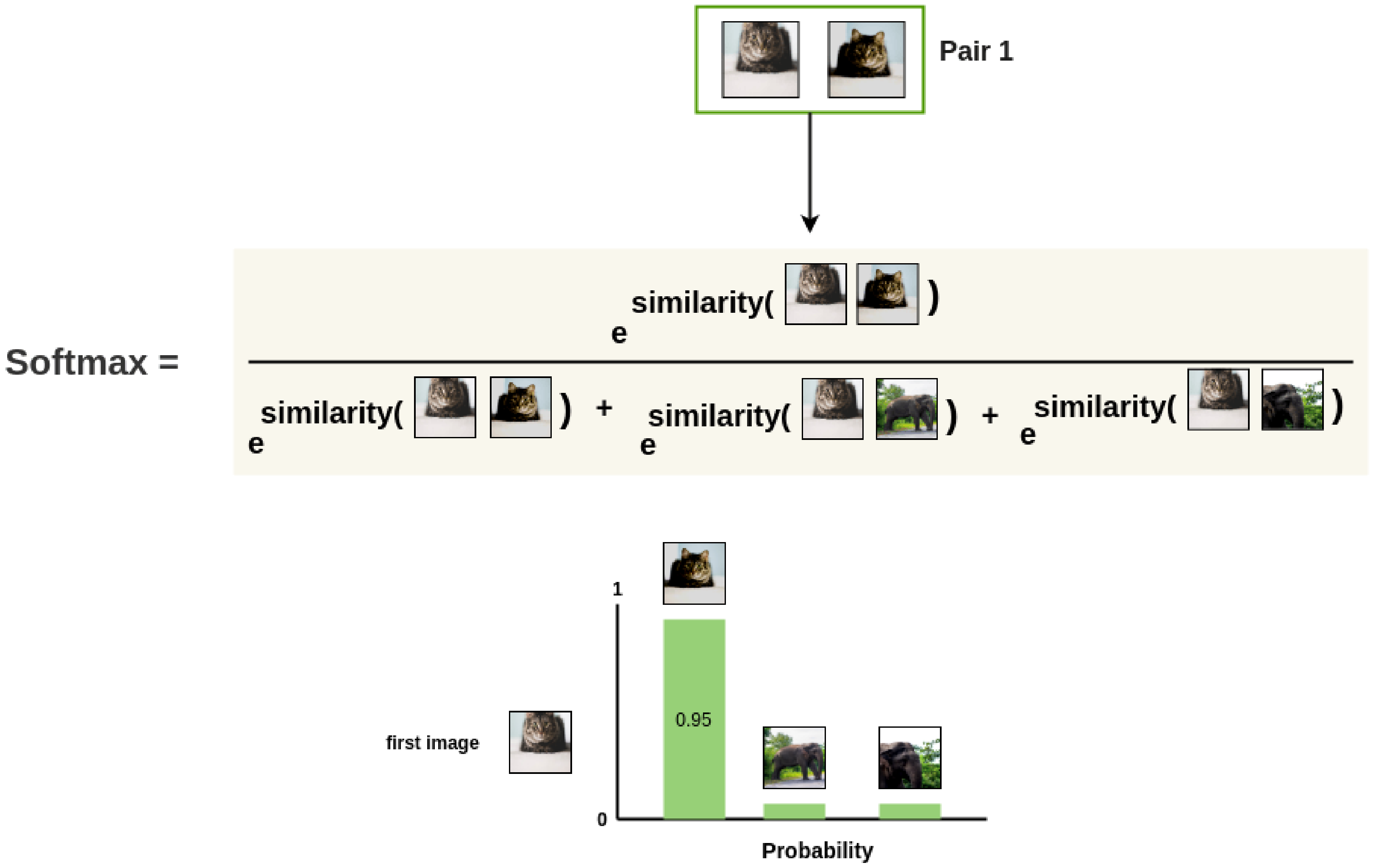


Figure from [blog]()

Noise Contrastive Estimation (NCE)

Let $\mathbf{x}$ be the target sample $\sim P(\mathbf{x}|C=1;\theta) = p_\theta(\mathbf{x})$ and $\tilde{\mathbf{x}}$ be the noise sample $\sim P(\tilde{\mathbf{x}}|C=0) = q(\tilde{\mathbf{x}})$. Note that the logistic regression models the logit (i.e. log-odds) and in this case we would like to model the logit of a sample u from the target data distribution instead of the noise distribution:

$$\ell_\theta(\mathbf{u}) = \log \frac{p_\theta(\mathbf{u})}{q(\mathbf{u})} = \log p_\theta(\mathbf{u}) - \log q(\mathbf{u})$$

After converting logits into probabilities with sigmoid $\sigma(.)$, we can apply cross entropy loss:

$$\mathcal{L}_{\text{NCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[\log\sigma(\ell_\theta(\mathbf{x}_i)) + \log(1-\sigma(\ell_\theta(\tilde{\mathbf{x}}_i)))\right]$$
$$\text{where } \sigma(\ell) = \frac{1}{1+\exp(-\ell)} = \frac{p_\theta}{p_\theta + q}$$

The idea is to run logistic regression to tell apart the target data from noise.

Binary classification

https://lilianweng.github.io/posts/2021-05-31-contrastive/

Noise Contrastive Estimation (NCE)

Given a context vector $\mathbf{c}$, the positive sample should be drawn from the conditional distribution $p(\mathbf{x}|\mathbf{c})$, while $N-1$ negative samples are drawn from the proposal distribution $p(\mathbf{x})$, independent from the context $\mathbf{c}$. For brevity, let us label all the samples as $X = \{\mathbf{x}_i\}_{i=1}^N$ among which only one of them $\mathbf{x}_{\text{pos}}$ is a positive sample. The probability of we detecting the positive sample correctly is:

$$p(C = \texttt{pos}|X, \mathbf{c}) = \frac{p(x_{\text{pos}}|\mathbf{c}) \prod_{i=1,\dots,N; i\neq \text{pos}} p(\mathbf{x}_i)}{\sum_{j=1}^N \left[p(\mathbf{x}_j|\mathbf{c}) \prod_{i=1,\dots,N; i\neq j} p(\mathbf{x}_i)\right]} = \frac{\frac{p(\mathbf{x}_{\text{pos}}|c)}{p(\mathbf{x}_{\text{pos}})}}{\sum_{j=1}^N \frac{p(\mathbf{x}_j|\mathbf{c})}{p(\mathbf{x}_j)}} = \frac{f(\mathbf{x}_{\text{pos}}, \mathbf{c})}{\sum_{j=1}^N f(\mathbf{x}_j, \mathbf{c})}$$

where the scoring function is $f(\mathbf{x}, \mathbf{c}) \propto \frac{p(\mathbf{x}|\mathbf{c})}{p(\mathbf{x})}$.

The scoring function f is related to mutual information optimization

The InfoNCE loss optimizes the negative log probability of classifying the positive sample correctly:

$$\mathcal{L}_{\text{InfoNCE}} = -\mathbb{E}\left[\log \frac{f(\mathbf{x}, \mathbf{c})}{\sum_{\mathbf{x}' \in X} f(\mathbf{x}', \mathbf{c})}\right]$$

Multi-class classification

https://lilianweng.github.io/posts/2021-05-31-contrastive/

SimCLR: A Simple Framework for Contrastive Learning of Visual Representations

1. Randomly sample a minibatch of N samples and each sample is applied with two different data augmentation operations, resulting in $2N$ augmented samples in total.

$$\tilde{\mathbf{x}}_i = t(\mathbf{x}), \quad \tilde{\mathbf{x}}_j = t'(\mathbf{x}), \quad t, t' \sim \mathcal{T}$$

where two separate data augmentation operators, t and t', are sampled from the same family of augmentations $\mathcal{T}$. Data augmentation includes random crop, resize with random flip, color distortions, and Gaussian blur.

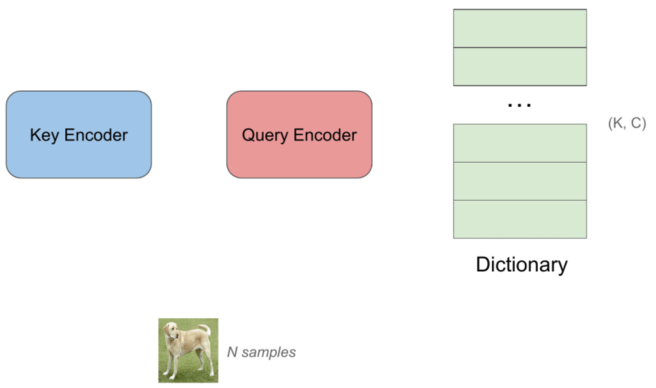
2. Given one positive pair, other $2(N-1)$ data points are treated as negative samples. The representation is produced by a base encoder $f(.)$:

$$\mathbf{h}_i = f(\tilde{\mathbf{x}}_i), \quad \mathbf{h}_j = f(\tilde{\mathbf{x}}_j)$$

3. The contrastive learning loss is defined using cosine similarity $\text{sim}(.,.)$. Note that the loss operates on an extra projection layer of the representation $g(.)$ rather than on the representation space directly. But only the representation $\mathbf{h}$ is used for downstream tasks.

$$\mathbf{z}_i = g(\mathbf{h}_i), \quad \mathbf{z}_j = g(\mathbf{h}_j)$$
$$\mathcal{L}_{\text{SimCLR}}^{(i,j)} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}$$

where $\mathbb{1}_{[k \neq i]}$ is an indicator function: 1 if $k \neq i$ 0 otherwise.



Text credit to https://lilianweng.github.io/posts/2021-05-31-contrastive/

Animation from blog

SimCLR for Downstream Tasks

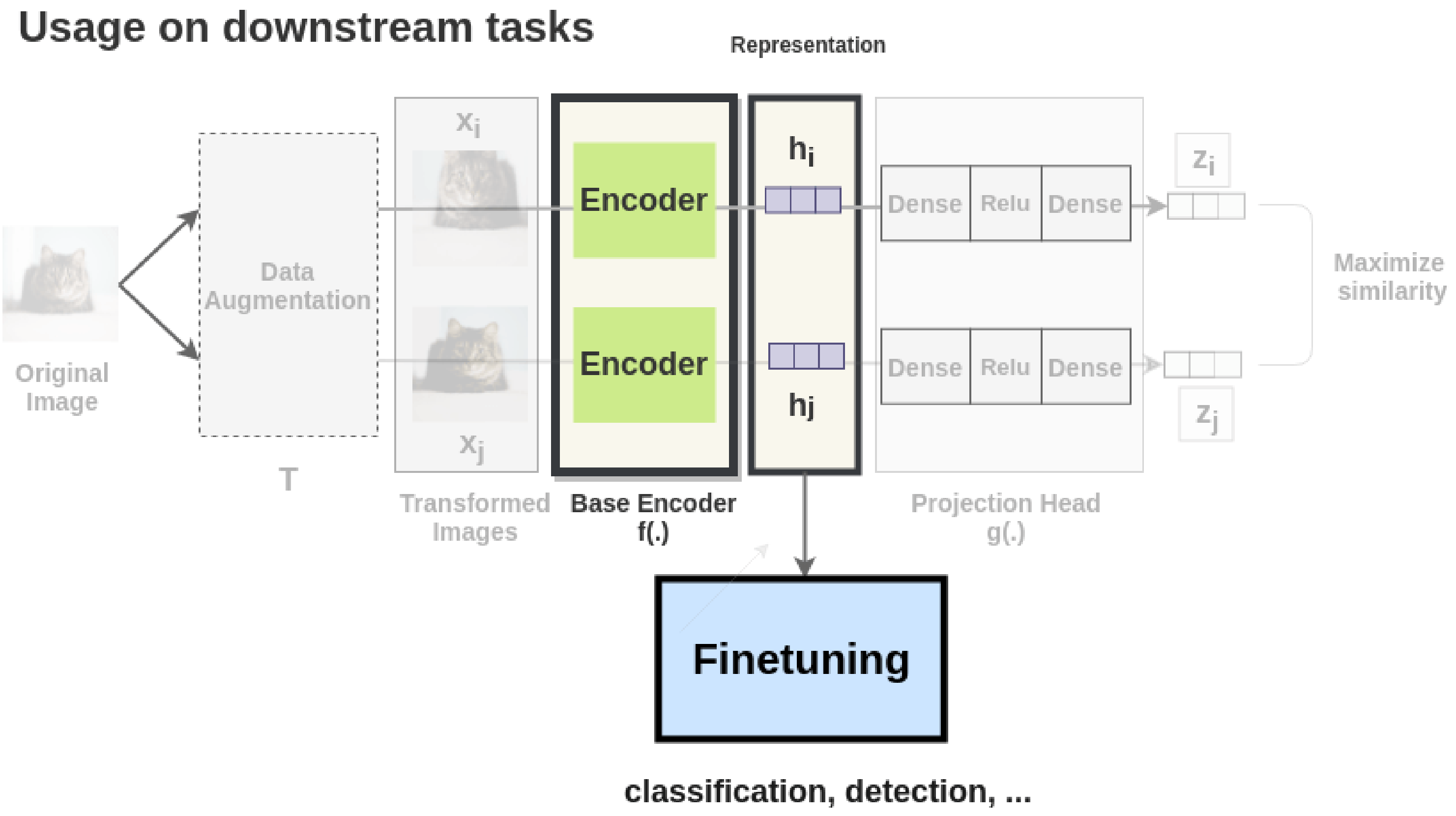


Figure from blog

Can we Align Text and Image?

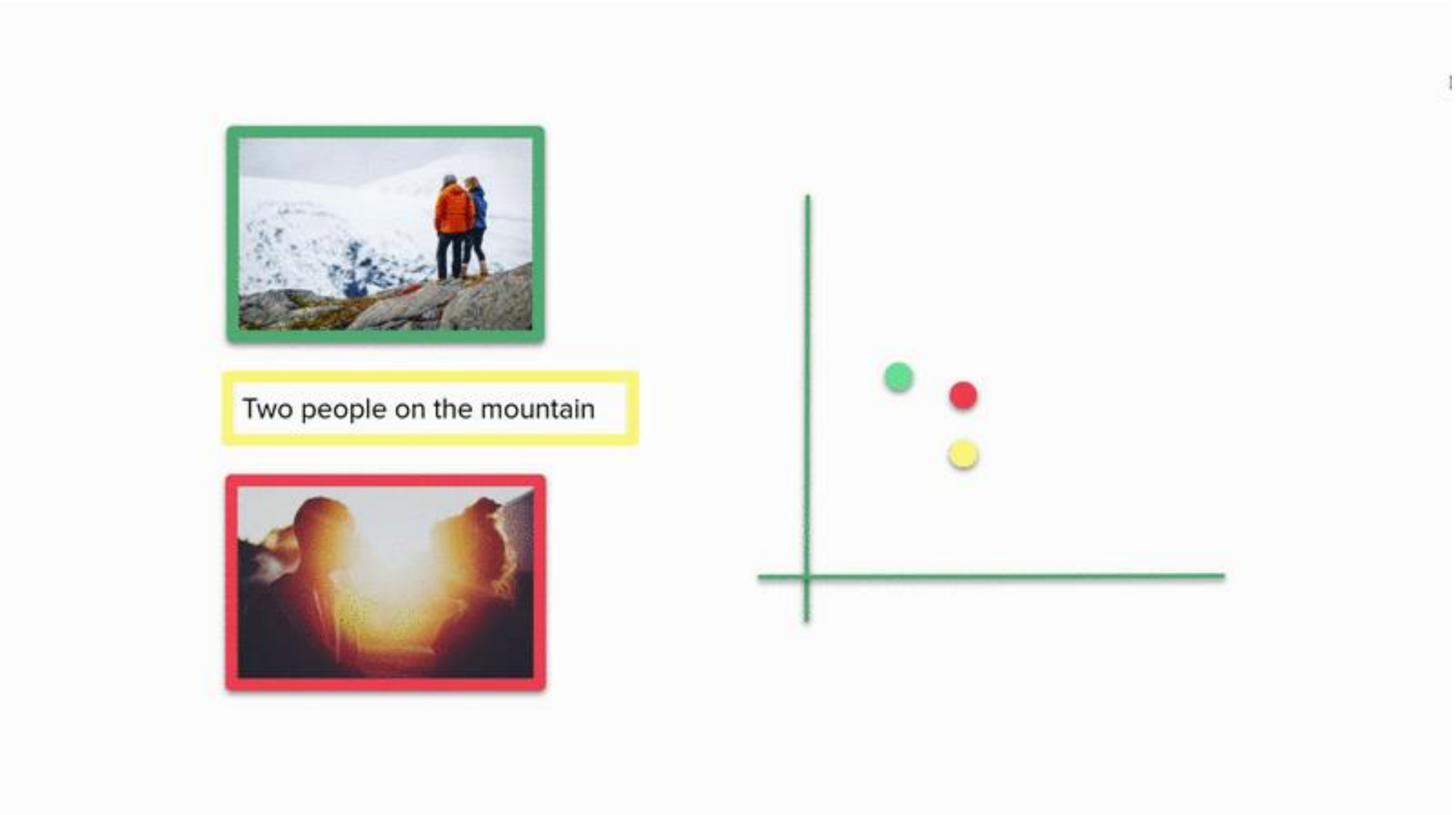


Figure from blog

Yes, Contrastive Learning!

Figure from [blog]

A Unified Space for Text and Image

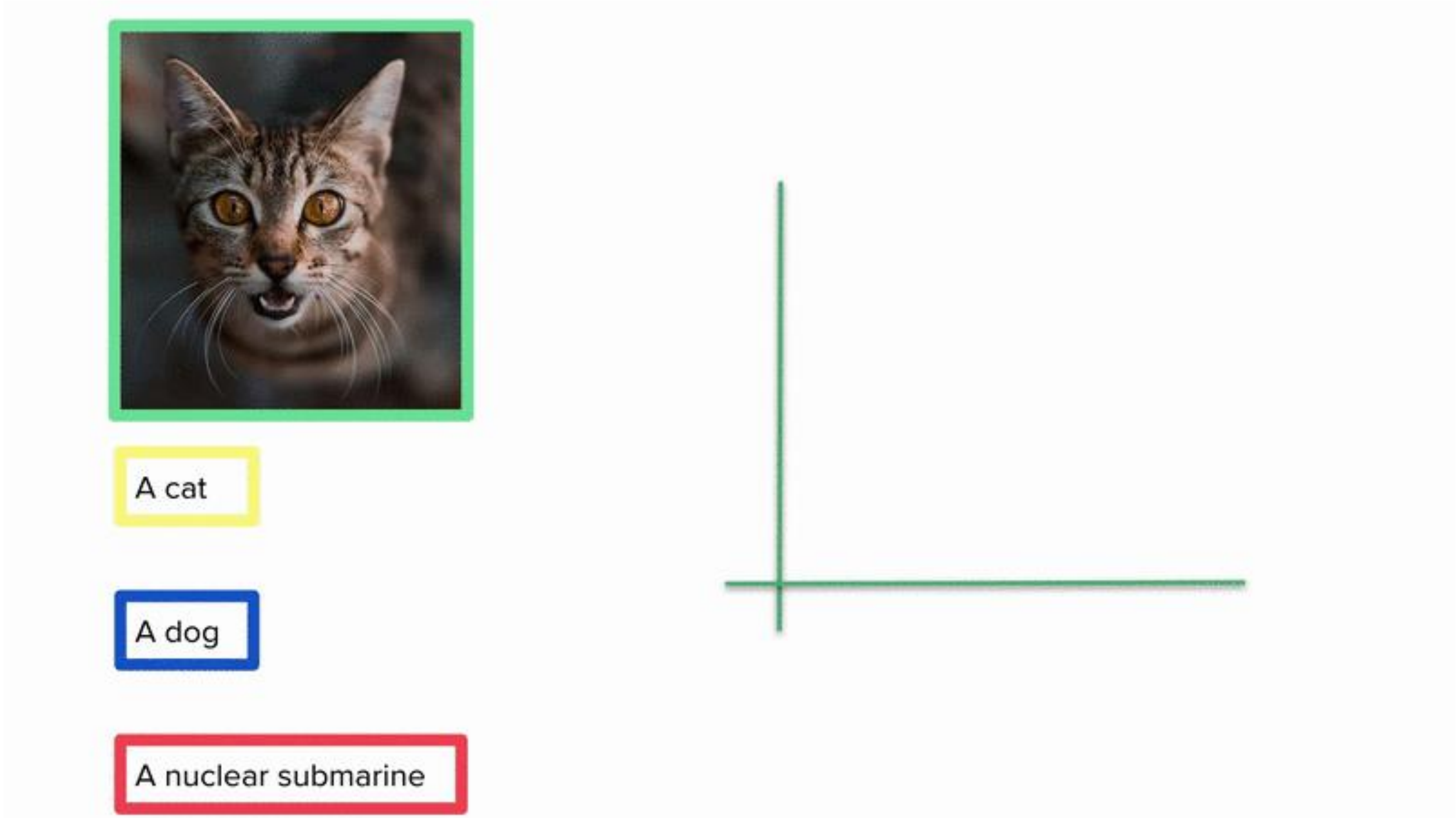
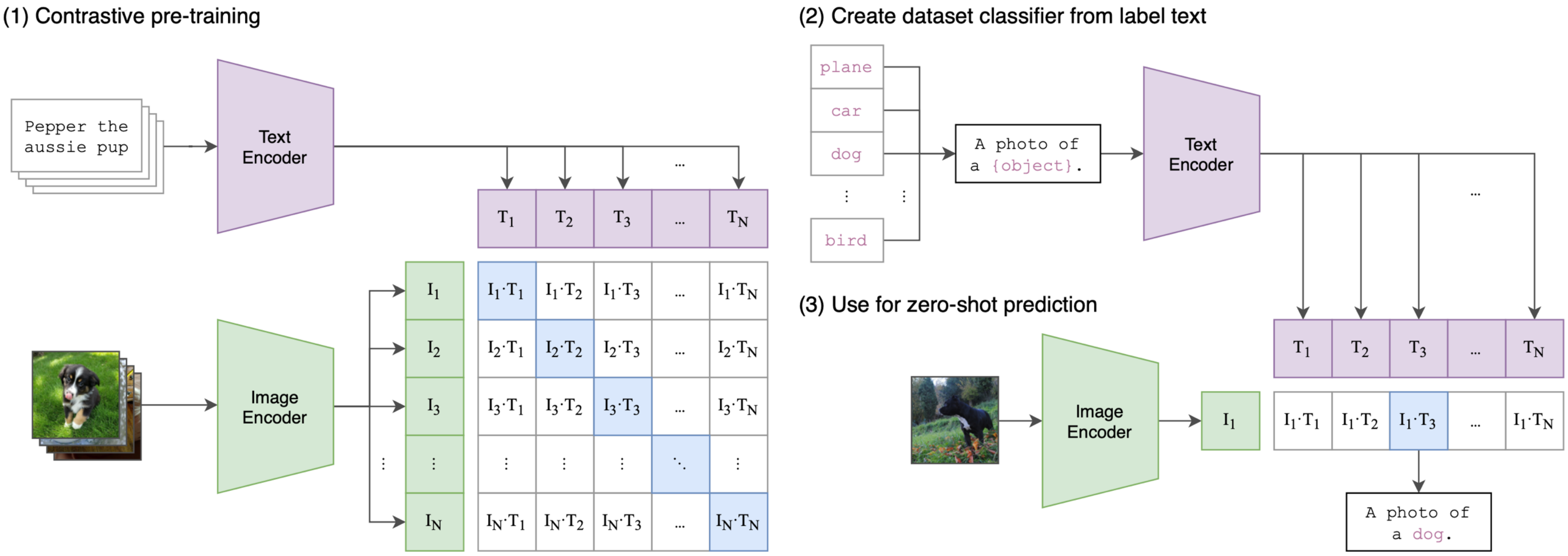


Figure from blog

CLIP: Connecting Text and Image



Learning Transferable Visual Models From Natural Language Supervision

Open-Vocabulary Classification

Food101

guacamole (90.1%) Ranked 1 out of 101 labels

- ✓ a photo of **guacamole**, a type of food.
- × a photo of **ceviche**, a type of food.
- × a photo of **edamame**, a type of food.
- × a photo of **tuna tartare**, a type of food.
- × a photo of **hummus**, a type of food.

SUN397

television studio (90.2%) Ranked 1 out of 397 labels

- ✓ a photo of a **television studio**.
- × a photo of a **podium indoor**.
- × a photo of a **conference room**.
- × a photo of a **lecture room**.
- × a photo of a **control room**.

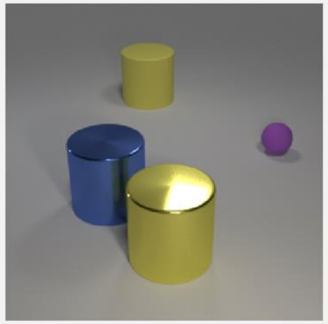
CIFAR-10

bird (40.9%) Ranked 1 out of 10 labels

- ✓ a photo of a **bird**.
- × a photo of a **cat**.
- × a photo of a **deer**.
- × a photo of a **frog**.
- × a photo of a **dog**.

CLEVR Count

4 (75.0%) Ranked 2 out of 8 labels



- × a photo of **3** objects.
- ✓ a photo of **4** objects.
- × a photo of **5** objects.
- × a photo of **6** objects.
- × a photo of **10** objects.

https://openai.com/index/clip/