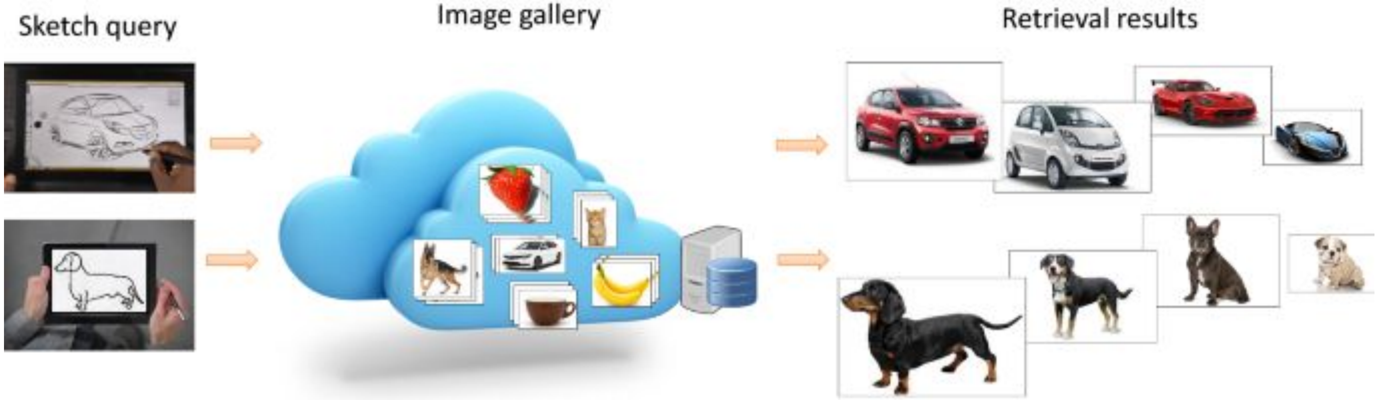
Drawing Based Image Retrieval

Nisarg Vora
2018A7PS0254P

Problem Introduction: Drawing based Image Retrieval

- Drawing based image retrieval is a task that has been explored a lot recently as an alternative method for image retrieval.
- A **drawing is given as a Query** based on which the relevant results are r**etrieved from an Image Gallery**
- **Drawing-based image retrieval is challenging because it requires comparison across two domains drawings and photos that are have distinct appearance with human imperfections**



Related Work

Important Paper: FaceNet: A Unified Embedding for Face Recognition and Clustering

Paper	Method	Type
SYM-FISH: A Symmetry-aware Flip Invariant Sketch Histogram Shape Descriptor	1.Computing the FISH descriptor 2.Discovering the symmetry character 3.Constructing a symmetry table combined with FISH descriptor.	Improved Shape Context Descriptor
Sketch Based Image Retrieval	1.ResNet / GoogleNet 2.Siamese Network 3.Triplet Network	CNN
Image-to-Image Demo	**cGAN** using pix2pix library tensorflow	Conditional Generative Adversarial Networks
Sketch to photo	1.Queries based on sketch + item tags. 2.Amalgamation of various techniques	Use of item labelings Amalgamation of various techniques
Deep Sketch Hashing	1.Convert images to sketches. 2.Use CNN to convert input sketch and sketches of all the images. 3.Use different loss metrics	Variations of Convolutional Neural Networks

Sketchy Dataset

Image Specifications: 256x256 RGB for the animal images

Drawing Specifications: 256x256 grey-scale images.

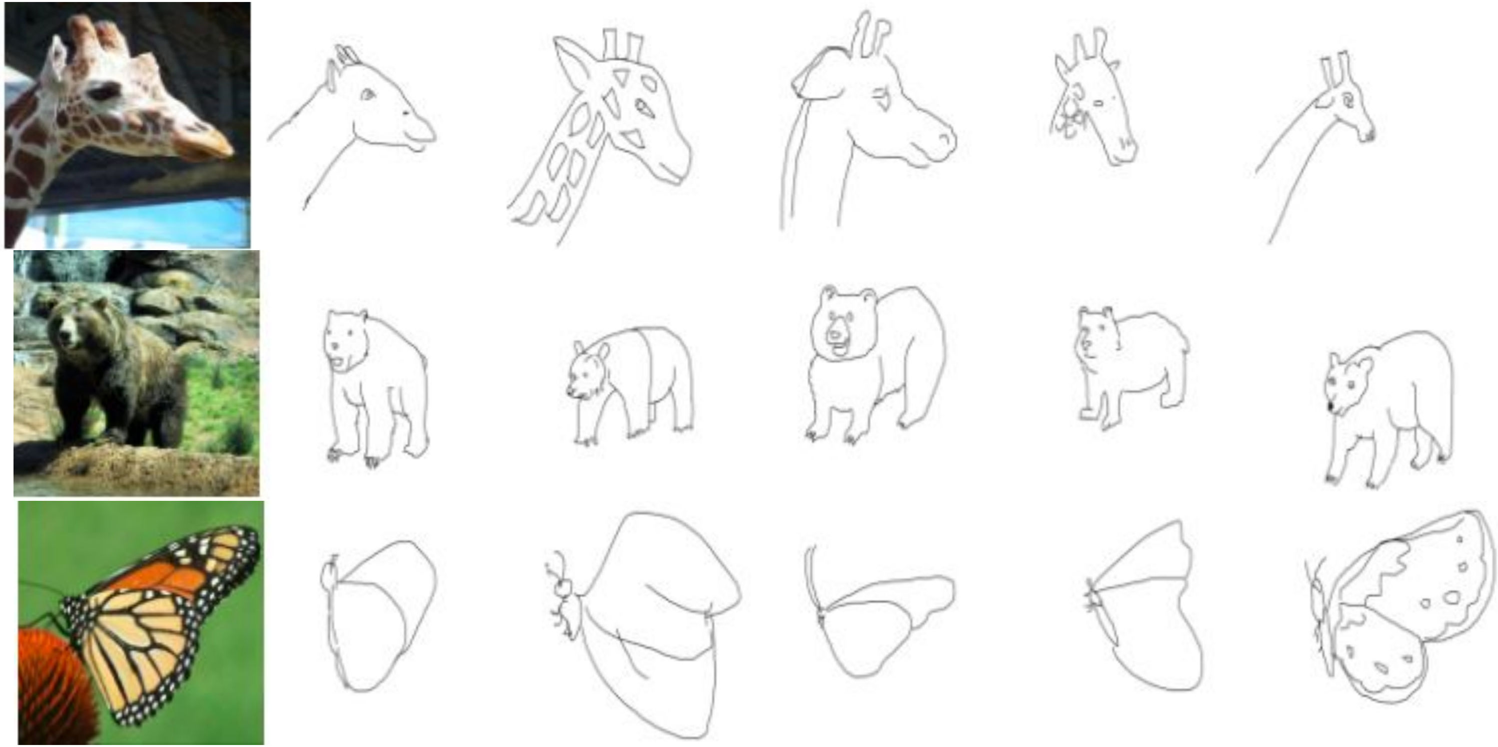
Size : 6 GB

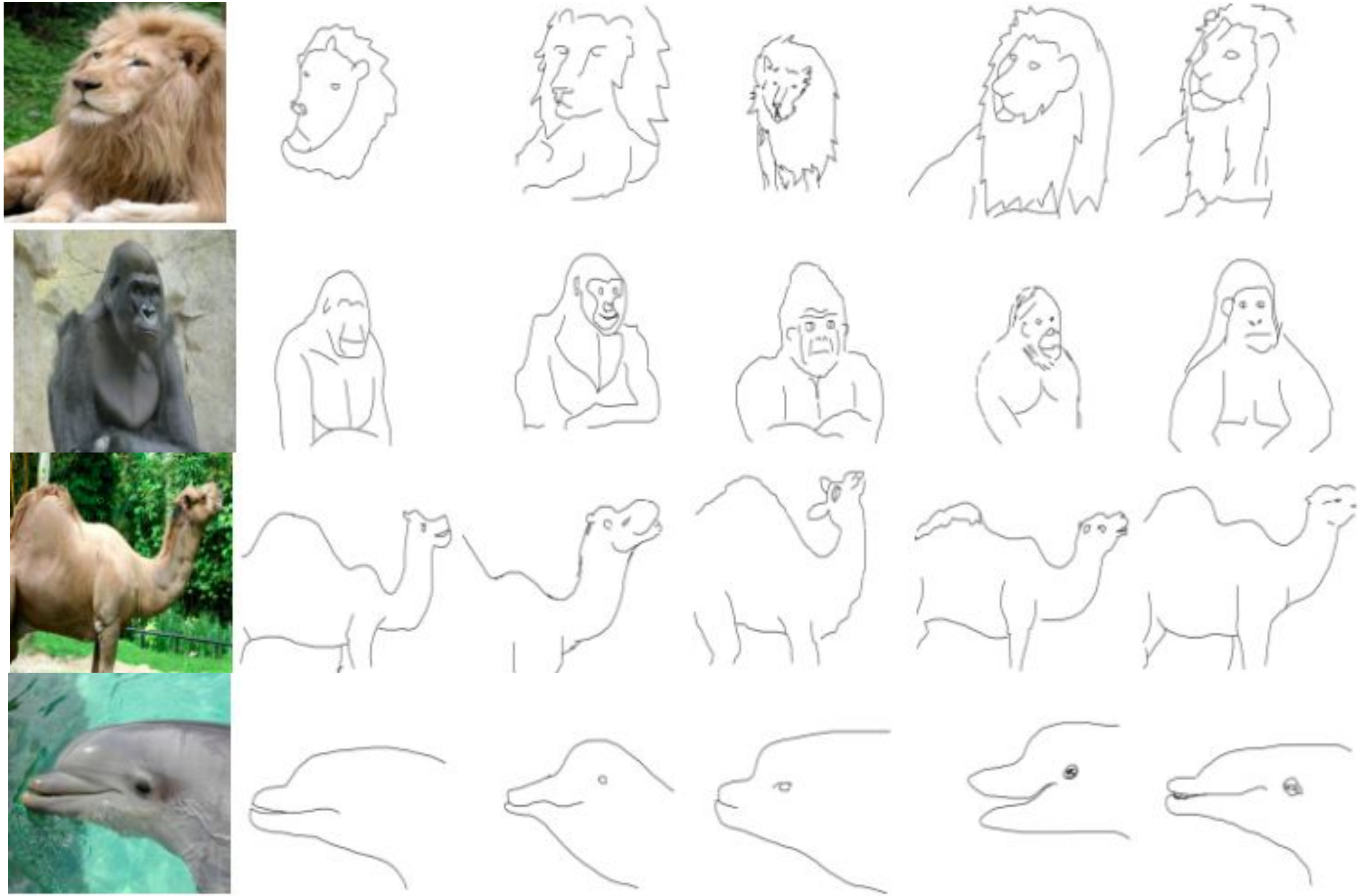
Dataset collected from a large crowd asked to select and draw a particular animal and/or object.

The Sketchy database contains 75,471 drawings of 12,500 objects spanning 125 categories.

The crowd consisted of people from varying age groups and backgrounds to properly incorporate randomness in the drawings. The participants were first shown a few images and were then asked to make the drawings representing those images based on their memory. This was done in order to mimic the real scenario in which the users make a particular drawing based on some old memory.

Preview:





Figuring-out Methodology:

- Image classification/identification/retrieval is a standard task in computer vision. In general, the image classification problem involves assigning one label out of a given fixed set of discrete labels to the input image on the basis of its visual content.
- While this is a trivial task for humans, robust image classification is a big challenge for a machine.
- To the computer, the image is just a grid of numbers which entirely change in unreliable ways with variations in viewpoint, illumination, occlusion, etc.
- As a result, there is no obvious algorithm which solves this problem. However, it has been discovered that a data driven approach, where the machine utilizes information regarding each class and machine learning techniques, is efficient.

Method used

Metric Learning with Siamese Network and Triplet Loss

Convolutional Neural Networks vs Neural Networks

Drawing Image size : 256x256

Assume a single layer neural network with 100 neurons.

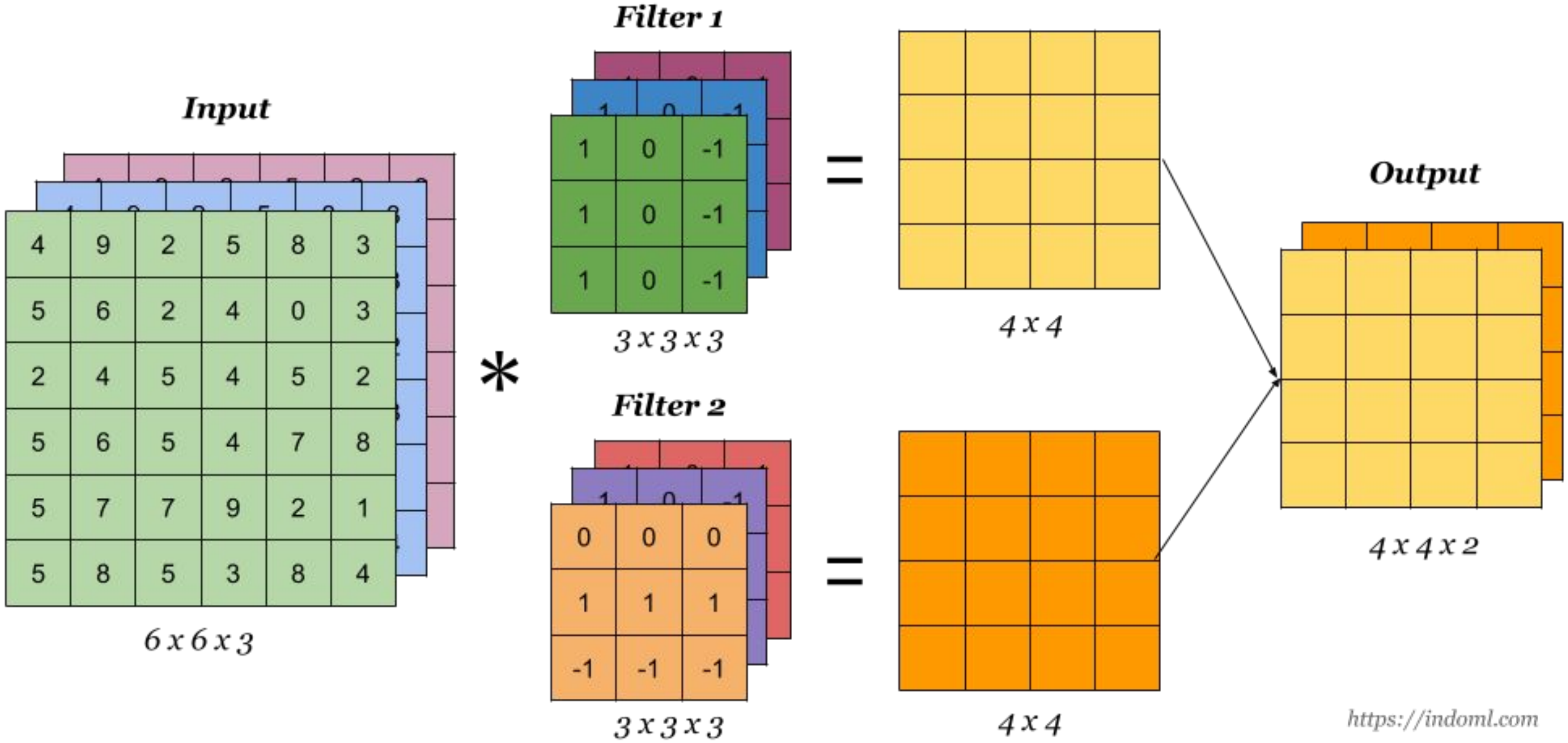
For a FCN,

Number of params = 256*256*101

= **66,19,136 params for a single layer**

Tends to overfit

Convolutional Neural Networks vs Neural Networks

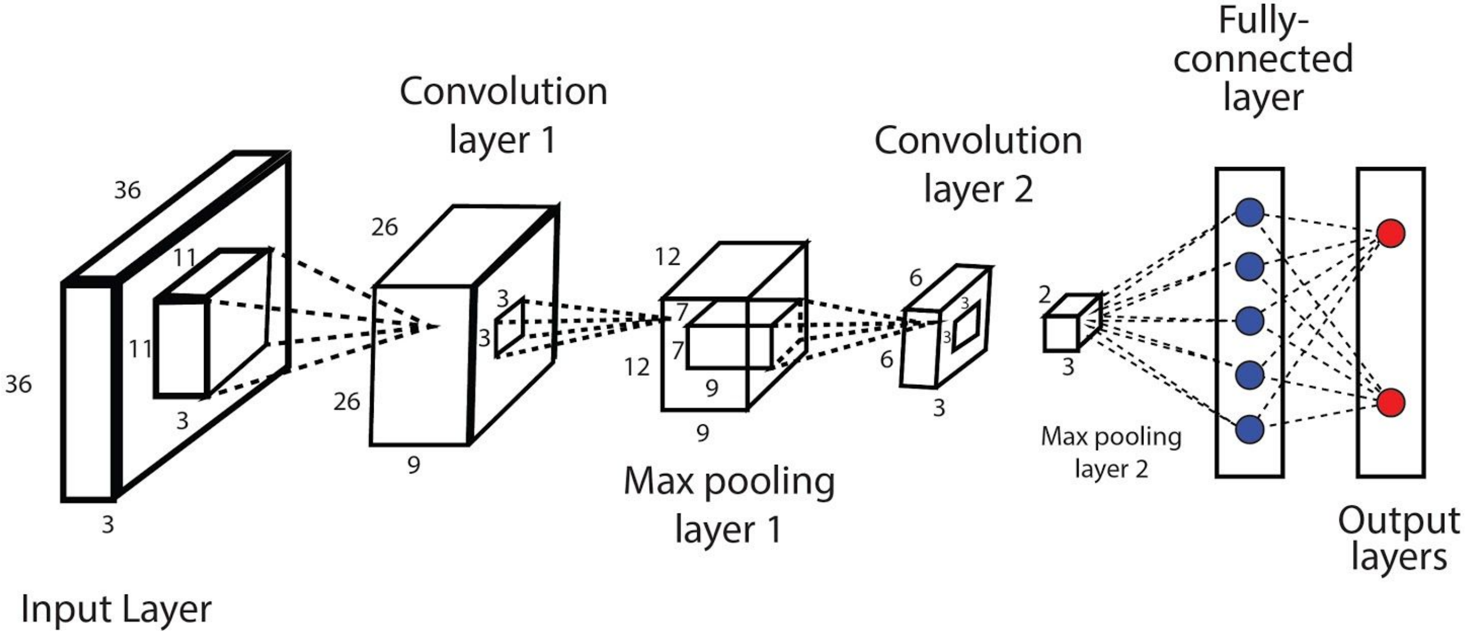


4	9	2	5	8	3
5	6	2	4	0	3
2	4	5	4	5	2
5	6	5	4	7	8
5	7	7	9	2	1
5	8	5	3	8	4

1	0	-1
1	0	-1
1	0	-1

0	0	0
1	1	1
-1	-1	-1

https://indoml.com


Fully-
connected
layer
Convolution
layer 1
Convolution
layer 2
36
11
36
11
3
3
Input Layer
26
26
9
3
3
12
7
7
12
9
9
Max pooling
layer 1
6
6
3
3
3
2
3
Max pooling
layer 2
Output
layers

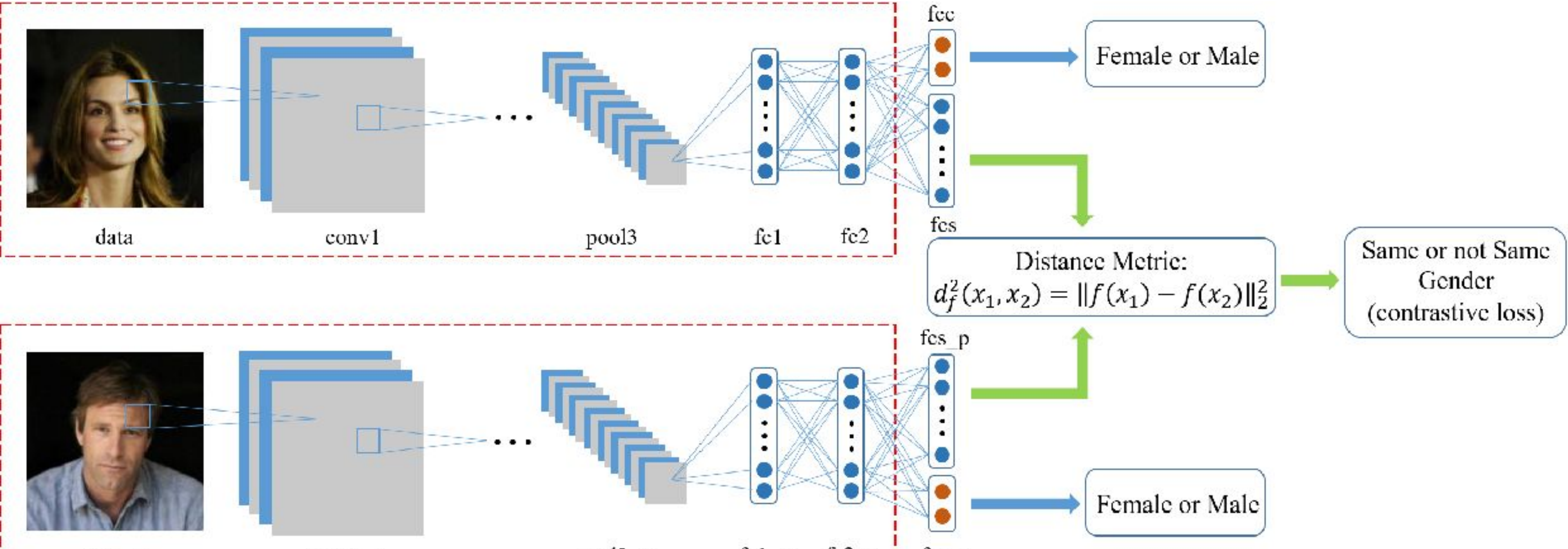
Drawbacks and Issues

1. Deep neural nets usually require vast amounts of data to train on to excel at a particular task

2. Everytime a new class is added the entire network needs to be changed

2.Metric Learning with Siamese and Triplet Convolutional Neural Networks

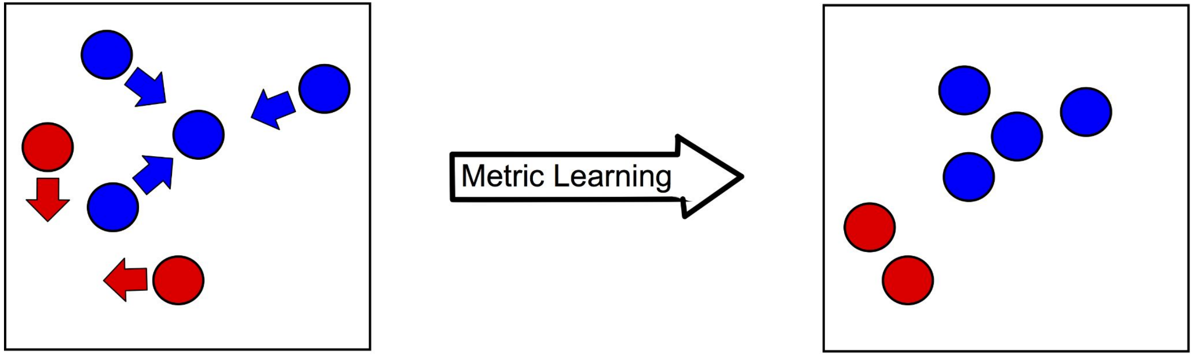
FaceNet: A Unified Embedding for Face Recognition and Clustering


fcc
Female or Male
data
conv1
pool3
fc1
fc2
fcs
Distance Metric:
$d_f^2(x_1, x_2) = \|f(x_1) - f(x_2)\|_2^2$
Same or not Same Gender (contrastive loss)
fcs_p
Female or Male

Metric Learning

The aim is to establish similarity or dissimilarity between objects, images or data points which is measured in terms of a distance metric.

Distance metric is such that the distance between closely related objects is minimised and that between different objects is maximized

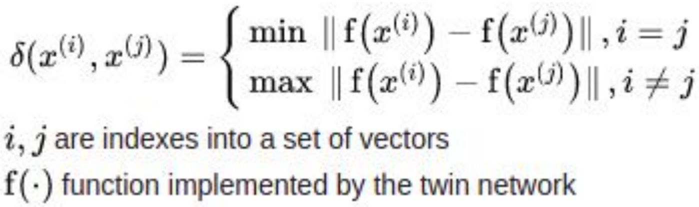


Siamese Network

Siamese networks train a similarity measure between the input points

The two inputs run through the same neural network and a feature vector is received as output from the neural network.

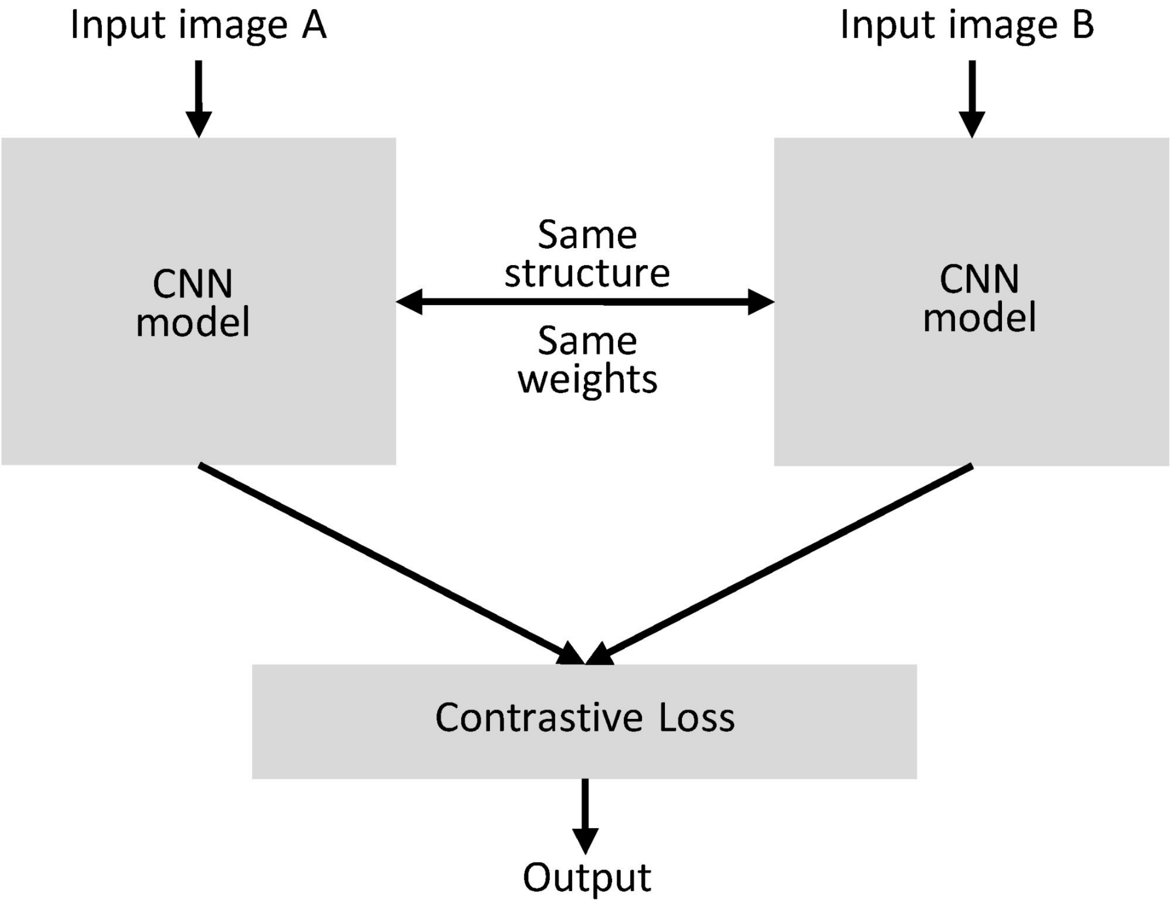
Based on the feature vector we compute a similarity metric and minimize the distance between similar items.



$$\delta(x^{(i)}, x^{(j)}) = \begin{cases} \min \ \| \mathrm{f}(x^{(i)}) - \mathrm{f}(x^{(j)}) \| \, , i = j \\ \max \ \| \mathrm{f}(x^{(i)}) - \mathrm{f}(x^{(j)}) \| \, , i \neq j \end{cases}$$

i, j are indexes into a set of vectors

$\mathrm{f}(\cdot)$ function implemented by the twin network


Input image A
Input image B
CNN model
Same structure
Same weights
CNN model
Contrastive Loss
Output

Triplet Loss

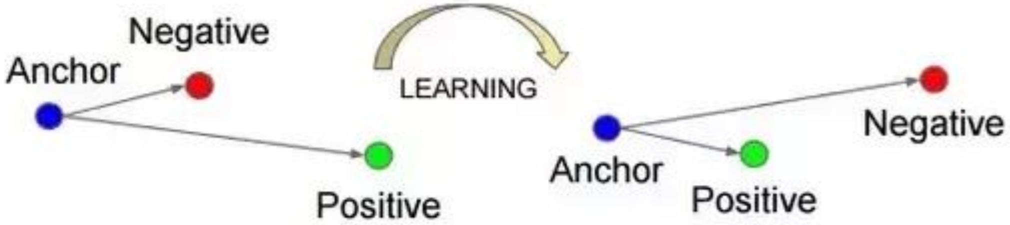


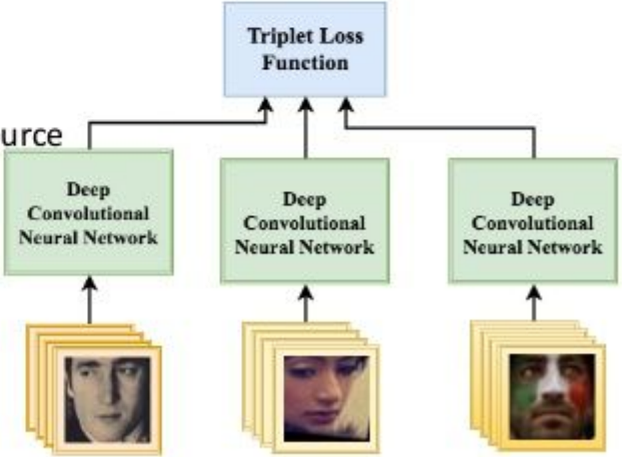
Figure 3. The **Triplet Loss** minimizes the distance between an *anchor* and a *positive*, both of which have the same identity, and maximizes the distance between the *anchor* and a *negative* of a different identity.

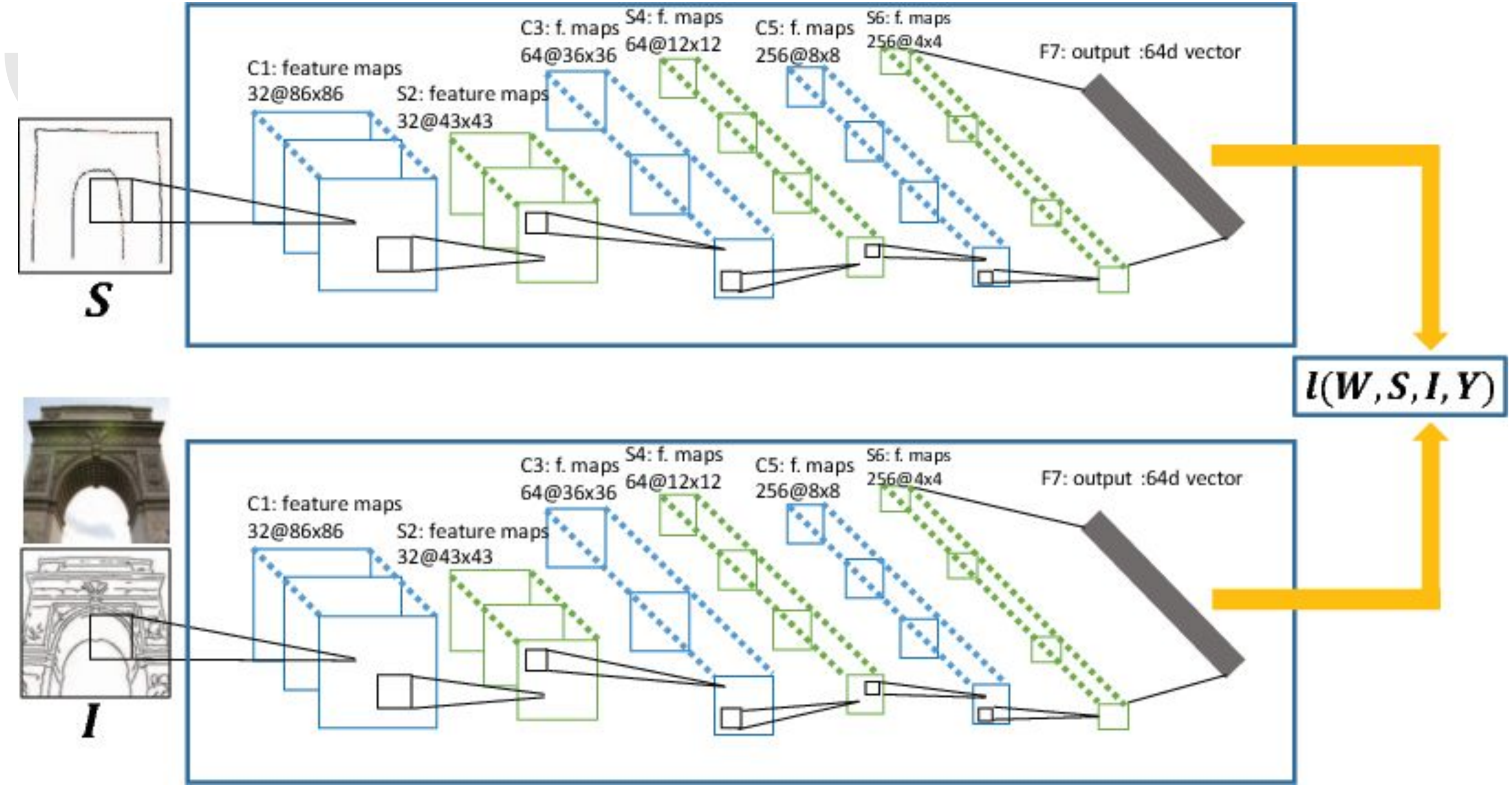
$$Loss = \sum_{i=1}^{N} \left[\|f_i^a - f_i^p\|_2^2 - \|f_i^a - f_i^n\|_2^2 + \alpha \right]_+$$

Triplet Loss Functions

- Construct triplets from input/embedding values
- Designed to make the pairwise loss function faster and less resource intensive
- Positive, Negative pairs

(x_1, x_2, x_3): $$I_s(x_1, x_2) = \ positive,$$ $$I_s(x_1, x_3) = \ negative$$


Triplet Loss Function
Deep Convolutional Neural Network
Deep Convolutional Neural Network
Deep Convolutional Neural Network


S
C1: feature maps
32@86x86
S2: feature maps
32@43x43
C3: f. maps
64@36x36
S4: f. maps
64@12x12
C5: f. maps
256@8x8
S6: f. maps
256@4x4
F7: output :64d vector
$l(W, S, I, Y)$
I
C1: feature maps
32@86x86
S2: feature maps
32@43x43
C3: f. maps
64@36x36
S4: f. maps
64@12x12
C5: f. maps
256@8x8
S6: f. maps
256@4x4
F7: output :64d vector

Preprocessing

The drawing images are grayscale images made with a pencil and do not have any color attribute.

Since the colors can not be incorporated into pencil based drawing or most of the drawings on touch screen devices it is very unlikely to affect the output images.

Therefore we convert the photos also to grayscale images and normalize both photos and drawings by dividing each by 255.

Model

Architecture

The network consists of 3 2D convolutional layers with filters of size 32 x 32, with ReLu activation function.

Each layer is followed by a max pooling layer and batch normalization. The output of these layers is then flattened and fed as input into a fully connected layer with 128 neurons from which the final feature vector is received as output.

The architecture was selected after a bunch of trial and error methods based on the RAM and GPU limitations of Google Colab which prevented us from selecting a better and more complex model.

Results

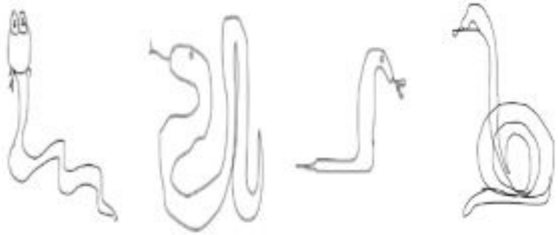
Given a drawing, the model retrieves the top k most relevant images to the drawing in the database. Retrieval @3 and @5 was used for measuring the performance of the model or all the relevant images that have a distance metric less than a threshold value.

Even though a very naive convolutional neural network model was used in the Siamese network for computing the feature vector, the results were promising.

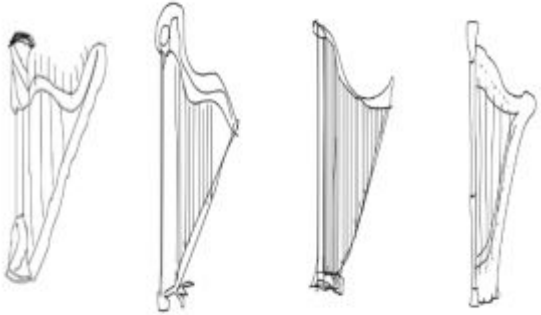
At a threshold of similarity distance metric equal to 0.03 the model retrieved the relevant images for 43.22% of the drawings. In retrieval @3, for 20.4% drawing queries, the relevant image was retrieved as one of the three images whereas in retrieval @5, for 27.3% drawing queries the relevant image was retrieved as one of the five images. The percentage of queries for which the relevant result is retrieved increases with increase in the number of images retrieved for each query simply due to the randomness and therefore retrieval at larger numbers has been avoided.

Limitations

The mode fails for a few closely related objects Like the example.



A few drawings are improper and match multiple images.



Further Work:

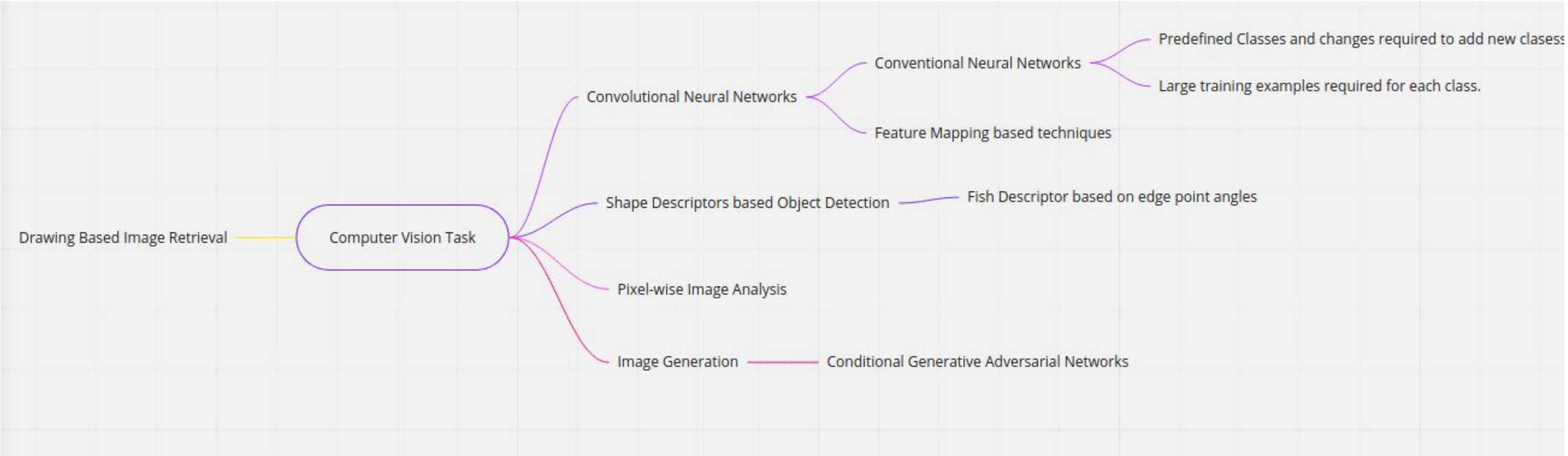
- Due to RAM and GPU based limitations on Google Colab only a very simple convolutional neural network in the Siamese Network on a partial dataset

- More complex neural networks like the GoogleNet could be implemented in the Siamese Network

- Using the Network on the entire dataset would definitely yield better results.

Thank You

Possible Approaches



Drawing Based Image Retrieval

Computer Vision Task

Convolutional Neural Networks

Conventional Neural Networks

Predefined Classes and changes required to add new clasess

Large training examples required for each class.

Feature Mapping based techniques

Shape Descriptors based Object Detection

Fish Descriptor based on edge point angles

Pixel-wise Image Analysis

Image Generation

Conditional Generative Adversarial Networks