Search by Images

Using Auto Encoders

Proposal Recap

Problem Statement

“Searching into a dataset of images by using an image”.

This can be used in

1. Getting the attached news.
2. Detecting rumors.
3. Searching for similar products.

Report on the related work.

Feature Detection

Using algorithms for matching image deformation such as blur, rotation, scale, and illumination change.
Such as **SIFT**, **PCA-SIFT** and **SURF**

Search by Color

"Piximilar Multicolor Search process"

Given archive of images→ create digital signature based colors (color histograms) → store the extracted signatures in database → whenever given a query image, extract its colors → find the most suitable matching image

Visual Similarity search

A set of algorithms is used to analyze image attributes ex (colour, shape, texture, luminosity, complexity, objects and regions). These attributes are stored for indexing and this can be used with keywords to refine searches on extremely large collections.

Auto Encoder

Encoding Images into 1D vector array and then finding the Nearest neighbours to a given query image.

The selected one to expand on.

Feature Detection

Using algorithms for matching image deformation such as blur, rotation, scale, and illumination change.
Such as **SIFT**, **PCA-SIFT** and **SURF**

Search by Color

"Piximilar Multicolor Search process"

Given archive of images→ create digital signature based colors (color histograms) → store the extracted signatures in database → whenever given a query image, extract its colors → find the most suitable matching image

Feature Detection

A set of algorithms is used to analyze image attributes ex (colour, shape, texture, luminosity, complexity, objects and regions). These attributes are stored for indexing and this can be used with keywords to refine searches on extremely large collections.

Auto Encoder

Encoding Images into 1D vector array and then finding the Nearest neighbours to a given query image.

Related work.

Auto Encoder

Most people uses AutoEncoder architecture to achieve this problem by training the data set into two models

Encoder : Encodes the image into 1D vector array

Decoder : Decodes the image back from the array and check the error

After that, the Decoder part is removed (to be used later) and only the encoder part is used to

1. Encode the whole dataset to 1D arrays.
2. Whenever an image query is received, Just encode it too and extract the nearest ‘N’ neighbours to this array.

Then finally, Decode the candidates using the decoder.

Related work.

People Approaches to the Auto encoder.

1. Fully connected layers.

2. Fully convolutional way
   a. Encoder : Conv + pooling
   b. Decoder : Conv + Up sampling

3. Convolution , DeConvolution
   a. Encoder : Accumulative convolutions layers + Pooling.
   b. Decoder : Accumulative transpose convolutional layers.

4. Mix of 2 and 1

Our proposed solution/model and Difference from related work.

We will try different approaches for this problem and compare them all to find the best way to get this task done.

1. First we will try the mix approach with Convolutional Neural Networks (CNNs) and FC layers

   ➢ The **Encoder** consists of the following layers

   - ✓ Convolutional Layers.
   - ✓ Non-Linearity layers.
   - ✓ Rectification Layers (can use just ReLU).
   - ✓ Pooling layers (Average and Max pooling)
   - ✓ Fully connected layers.
   - ✓ Dropout layers.

   ➢ The **Decoder** consists of the following layers

   - ✓ FC layers (undos effect of last layer in encoder).
   - ✓ Transpose Convolution Layers (Reversing the encoder)

Our proposed solution/model and Difference from related work.

2. Second we will try to improve that by using Recurrent Neural Networks (**RNN**) instead of CNN:

   - ➢ LSTM network:
     - ✓ Each layer of LSTM has as many cells as the timesteps and using ReLU as activation function.
     - ✓ It takes 2d as input (each layer).
     - ✓ If the subsequent layer is also LSTM, we will duplicate this vector using “RepeatVector(timeSteps)” to get a 2D array for the next layer.

Our proposed solution/model and Difference from related work.

3. Thirdly we will try to use the **Attention model** alongside RNN Encoder-Decoder model:

A potential issue with this encoder–decoder approach is that a neural network needs to be able to compress all the necessary information of a source information into a fixed-length vector. This may make it difficult for the neural network to cope with long or big images, especially those that are larger than the images in the training corpus.

Attention model is proposed as a solution to this limitation.

0.0151 MSE loss on the training data

0.0174 MSE loss on the validation data

0.0173 MSE loss on the testing data

Implementing and testing AE model using:

- ✓ FC layers.
- ✓ Conv + Up Sampling.
- ✓ Conv + TransposeConv.
- ✓ Conv + FC + Up Sampling.
- ■ Conv + FC + TransposeConv.
- ■ Manipulating params for the above.
- ■ RNN
- ■ Attention Model
- ■ Apply nearest neighbour to be done.

What we have achieved so far.

Approach \ MSE train loss by	Them	Winner	Us
Fully Connected Only	0.0214	→	0.0026
Convolution + Up Sampling	0.0407	→	0.0065
Convolution + DeConvolution	0.0297	→	0.0091
Conv + FC + Up Sampling	0.0173	→	0.0050

Them :
https://towardsdatascience.com/aligning-hand-written-digits-with-convolutional-autoencoders-99128b83af8b
https://towardsdatascience.com/build-a-simple-image-retrieval-system-with-an-autoencoder-673a262b7921
https://medium.com/analytics-vidhya/building-a-convolutional-autoencoder-using-keras-using-conv2dtranspose-ca403c8d144e

- Currently we have a working CNN Encoder-Decoder Model.
- We will try modifications and enhancements on the CNN approach.
- Our next step would be improving that by using RNNs instead of CNNs.
- A future step would be trying to fix the limitations and enhancing the RNNs using Attention model.

Next steps for the rest of the project.

Thanks !