Multiple Instance Recognition Using Deep Neural Network

CS783:Visual Recognition

1

Submitted By :
Sachin Sardiwal(150608)
Naveen Chandra(160432)

Contents

1 Approach

We have used VGG16 architechure to get the feature descriptors of the images. These feature descriptors are stored in an index file. Using Cosine Similarity, similarity ranking is obtained between query image and images present in the database. These ranking are then used to retrieve images from the dataset.

2 Dataset

The orginal dataset contains 3456 images with 16 different classes. To train such a large network as VGG16 we require a bigger dataset.We have applied data augmentation techniques such as translation, rotation and affine transformations on the original dataset to get a bigger dataset of size 13792. This dataset is then divied into train(70%) and test(30%) dataset.

3 Training a Classification Network

The task is to train a classificatin network with 16 classes. Since the initial layers of VGG16 captures more generic features, we have included those layers in our network. The fully connected layers at the top are removed as they are not useful. Now we have tried many models by varying the number of layers and the size of each layer to get better results. The model is trained on the augmented dataset with following values of hyperparameters.
loss = binary cross entropy
optimizer = sgd
number of epochs = 10
Batch Size = 32
Learning rate = 0.05
Decay = 1e-5
momentum = 0.9

Three fully connected layers are added on the top of VGG16 with sizes 4096,4096 and 1024. All of them have relu activation. A classification layer of size 16 is added then with 'sigmoid' activation.
This network have an accuracy of 99% on test dataset.

4 Creating Index File

4.1 Intermediate model

To be able to match images we need a feature description of each image. This is done by taking one of the fully connected layer as feature descriptor of the image. An intermediate model is defined which uses the pretrained weights and outputs one specific fully connected layer.
This intermediate model is run on every image on our original dataset to get feature descriptors. These features are then L2 normalised and stored in a '.hdf5' file.
Various layers are used as output to intermediate model. We have explored some of the possiblities.
model 1 : output: 'fc9', size 1024
model 2 : output: 'fc8', size 4096
model 3 : output: 'fc7', size 4096

5 Retrieval

Query image is fade into the intermediate network to get the feature descriptor. The feature descriptor is then normalised. Index file is stored in a matrix. By taking the dot product of query feature vector and index matrix, we get a similarity score. By sorting we get the required ranking of images.

6 Results

6.1 Comparison of different models

Model Performance				
Model	mAP at k =10	mAP at k = 30	mAP at k = 100	mAP at k = 200
Model 1	56.00	55.77	50.2	43.10
Model 2	66.00	64.00	58.66	28.96
Model 3	80.00	78.44	67.40	49.56

7 Hashes

Model 1
af1b16c8109640d0c486f51bb38047be CNN.py
1150088ca9a26eb211d197f65750de81 index.py

3a9f39d620a123ae5ea3a3552d205cdd retrieve.py
696ae677ff2e69b77bedf7cec413fd5d evaluate.py
b7a480ef141978be75c2cab127a6d4a4 pretrained/vgg16.h5
903bb244de6a551a8212be48fd44de36 output/index.hdf5

Model 2
af1b16c8109640d0c486f51bb38047be CNN.py
3e0e82e8328d4ec17d1668dbea44152e index.py
0cd1bc9c6bd7a60609fdf839d33817a8 retrieve.py
696ae677ff2e69b77bedf7cec413fd5d evaluate.py
b7a480ef141978be75c2cab127a6d4a4 pretrained/vgg16.h5
89f6236ca3942b70e1470bc071c19e43 output/index.hdf5

Model 3
af1b16c8109640d0c486f51bb38047be CNN.py
2db8a08abc5b04c37aeece3a015a9caa index.py
7984032af970155e20b2e823dacf40ee retrieve.py
696ae677ff2e69b77bedf7cec413fd5d evaluate.py
b7a480ef141978be75c2cab127a6d4a4 pretrained/vgg16.h5
193a081e0defa3cfbe9a046de8082c35 output/index.hdf5

8 Conclusion

The highest accuracy is obtained in Model 3 where the first fully connected is used to extract features. It achieves a mAP of 80% at $k = 10$ on test instances.

Bibliography

[1] Ji. WAN, Dayong WANG, Steven C. H. HOI, and Pengcheng WU, Jianke ZHU, *Deep learning for content based image retrieval: A comprehensive study*,2014