# Visual Search Assignment

Preferrable use the Readme.md file in an application where you can see the Markdown Preview (Atom, Sublime, Pycharm, etc)

Project Structure

- MATLAB Scripts<br>

- Python (Directory)<br>

scripts for SIFT, transfer learning and SVM classification

- cwork\_basecode\_2012 (Directory)<br>

basecode provided and improvised<br>

-descriptors(Directory, needs to be created by user)

-Results(Directory, needs to be created by user)

-PR\_Plots(Directory, needs to be created by user)

Setting up your project

1. Create a descriptors folder

2. Create a Results folder

3. Make sure the MSRC\_ObjCategImageDatabase\_v2 is in the top project-level folder or where the README.md file exists

4. Add cwork\_basecode\_2012 folder to MATLAB path

SIFT Descriptors - Bag of Visual Words, Transfer Learning and SVM Classification was computed in Python.

1. Install Python 3.6=<

2. Install opencv-Python

3. Install Keras

4. Install sklearn

5. Install Scipy

6. install argparse

Compute Global Colour Histogram

1. Create a “colorHisto\_4” folder inside descriptors folder

2. Run the following command

computeGlobalColorHistDescriptors("RGBHISTO\_4", 4);

The above command will create .mat descriptors inside colorHisto\_4 folder with quantization 4

ComputeGridTextureDescriptors

1. Create “TD\_25\_30” folder in descriptors folder

2. Run the following command

computeGridTextureDescriptors("TD\_25\_30", 25, 20);

This computes texture descriptors with grid size 25 and 20 edge orientation ranges

Compute Principal Components

1. Create PCA\_TD\_25\_30 folder in descriptors folder

2. Run the following command

computePCA("TD\_25\_30", "PCA\_TD\_25\_30", 20);

This will compute the top 20 principal components of your features dimensions and save the descriptors in PCA\_TD\_25\_30 folder. It will also save the projection matrix in PCA\_TD\_25\_30 folder.

To compute components that contribute 97% of the variance

computePCA("TD\_25\_30", "PCA\_TD\_25\_30", 0.97, true);

Perform VisualSearch

[p, r, ap] = cvpr\_visualsearch("colorHisto\_4", "cosine" );

To perform visual search on class 4 and return top 20 Results

[p, r, ap] = cvpr\_visualsearch("colorHisto\_4", "Euclidean", 20, 4);

Perform Visual Search with Mahalanobis (Only for PCA descriptors which have projectionMatrix.mat in their DESCRIPTOR\_SUBFOLDER)

[p, r, ap] = cvpr\_visualsearch("PCA\_TD\_25\_30", "Mahalanobis", 20, 4);

## SIFT and BOVW

To compute the SIFT Descriptors followed by computing Bag of Visual Words (BOVW) use `compute\_sift\_bovw\_Descriptors.py` in the Python Folder

Syntax:

python compute\_sift\_bovw\_Descriptors.py "path to dataset"(optional) output\_path(mandatory) k-clusters(mandatory)

Defaults

dataset\_path is the top-level project folder or where README.md file is. (../MSRC\_ObjCategImageDatabase\_v2/Images/)

Example:

Create a SIFT\_BOVW folder inside descriptors folder

python compute\_sift\_bovw\_Descriptors.py "..descriptors/SIFT\_BOVW" 100

Will compute BOVW with 100 n\_clusters and save the features in SIFT\_BOVW folder

If your dataset is in a different path, for example in home directory, use:

python compute\_sift\_bovw\_Descriptors.py --dataset\_path "~/MSRC\_ObjCategImageDatabase\_v2/Images/" "..descriptors/SIFT\_BOVW" 100" 100

Transfer Learning

To compute the transfer learning descriptors with vgg19, resnet50or resnet152 use compute\_transfer\_learning\_descriptors.py in the Python Folder

### Syntax:

python compute\_transfer\_learning\_descriptors.py --dataset\_path "path to dataset"(optional) output\_path(mandatory) model\_name(mandatory)

model\_name values: ["vgg19", "resnet50", "resnet152"]

Defaults

dataset\_path is the top-level project folder or where README.md file is. (../MSRC\_ObjCategImageDatabase\_v2/Images/)

### Example:

Computing vgg19 descriptors

Create a VGG19 folder inside descriptors folder

python compute\_transfer\_learning\_descriptors.py "..descriptors/VGG19" "vgg19"

Will compute features using pre-trained VGG19 model and save the features in VGG19 folder

If your dataset is in a different path, for example in home directory, use:

python compute\_transfer\_learning\_descriptors.py --dataset\_path "~/MSRC\_ObjCategImageDatabase\_v2/Images/" "..descriptors/ResNet\_50" "resnet50"

Testing your Descriptors

Defaults

Distance Metrics- ["Euclidean", "Cosine", "Manhattan"]

points = [3, 5, 10, 15, 20]

Use:

calculate\_test\_results(DESCRIPTOR\_SUBFOLDER, [3, 5])

This will calculate Average precision per class and Mean Average precision for top 3 and top 5 results with cosine, Euclidean and Manhattan Distance. The Average Precision per class is calculated by querying randomly for that class and averaging the 10 average precisions. This will create a Results.xlsx in the Results and 3 sheets one for each distance type

calculate\_test\_results(descriptors/TF\_25\_30);

To calculate with Mahalanobis Distance (make sure there is a projectionMatrix.mat file in your DESCRIPTOR\_SUBFOLDER)

calculate\_test\_results("DESCRIPTOR\_SUBFOLDER", [3, 5] ,true)

Compute PR Graphs

Syntax

computePRGraph(DESCRIPTOR\_SUBFOLDER, distanceMetrics)

This will compute PR graphs for distanceMetrics and save it in PR\_Plots Directory, one image for each distance metric. Use "Mahalanobis" only for PCA descriptors (projectionMatrix in the DESCRIPTOR\_SUBFOLDER);

Example

computePRGraph("RGBHISTO\_4"); %Will compute Euclidean, cosine and Manhattan by default

To compute Mahalanobis and cosine with PCA descriptors:

computePRGraph("PCA\_TD\_25\_30", ["Mahalanobis", "cosine"]);

SVM Classification

Syntax

python svm\_training.py "path to features"(mandatory) --kernel (optional) --gamma (optional) --test\_size (optional)

This will also print the Accuracy, Classification report with precision and recall per class and mean average precision and plot\_conf\_matrix a confusion matrix

Default Values

kernel - 'sigmoid'<br>

gamma - 0.00001<br>

test\_size - 0.15<br>

More information about kernels and gamma at https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

Example:

Training with vgg19 descriptors

python svm\_training.py "../descriptors/vgg19/" --kernel "sigmoid" --gamma 0.000001

Training with a different test size

python svm\_training.py "../descriptors/vgg19/" --test\_size 0.2

#### Best parameters for VGG19

kernel - 'sigmoid'<br>

gamma - 0.000001<br>

#### Best Parameters for ResNet\_50

kernel - 'sigmoid'<br>

gamma - 0.00001<br>