ISPR

2° midterm

assignment 2 – LDA on images

Frigo Giacomo (ID: 626201)
g.frigo@studenti@unipi.it

code snippets

```python
def sift_descriptors(path, drawKeypoints=False):
    image = cv2.imread(path)
    sift = cv2.xfeatures2d.SIFT_create()
    # detect keypoints and compute descriptors
    kp, des = sift.detectAndCompute(image, None)
    if (drawKeypoints):
        cv2.drawKeypoints(image, kp, image)
        cv2.imshow("keypoints", image)
        cv2.waitKey(0)
    return kp, des
```

```python
# extract descriptors from all the images (training + test)
for image in training_set + test_set:
    mser_kp, mser_des = mser_descriptors(image['path'].__str__())
    orb_kp, orb_des = orb_descriptors(image['path'].__str__())

    kp = np.concatenate((mser_kp,orb_kp))
    des = np.concatenate((mser_des, orb_des))
```

run kmeans on the training set and create the BoVW

```python
kmeans_obj, assigned_clusters = kmeans(all_training_descriptors, n_clusters=n_clusters)
print("Kmeans completed\nCreating images data structures..")
# create BoW for each image in the training set (array of tuples (cluster_code, # descriptors))
slot_start_index = 0
for image in training_set:
    image['assigned_clusters'] = assigned_clusters[
                                 slot_start_index:slot_start_index + image['descriptors'].shape[0]]
    slot_start_index += image['descriptors'].shape[0]

    # histogram
    image['hist'] = [0 for x in range(0,n_clusters)]
    for cluster in image['assigned_clusters']:
        image['hist'][cluster] += 1

    # creating array of tuples (cluster_code, # descriptors)
    image['frequencies'] = [(i, value) for i, value in enumerate(image['hist'])]
```

```python
# create BoW for each image in the test set
for image in test_set:
    # get cluster ids of test img descriptors exploiting the kmeans run before
    image['assigned_clusters'] = kmeans_predict(kmeans_obj, image['descriptors'])

    # histogram
    image['hist'] = [0 for x in range(0, n_clusters)]
    for cluster in image['assigned_clusters']:
        image['hist'][cluster] += 1

    image['frequencies'] = [(i, value) for i, value in enumerate(image['hist'])]
```

create the BoVW for each image in the test set

code snippets

```
# init LDA model
lda_model = ldamodel.LdaModel([image['frequencies'] for image in training_set], num_topics=n_topics,
                              alpha=alpha, per_word_topics=True, passes=passes, iterations=iterations)
```

```
for image in test_set:
    # get document topics from LDS model
    # set per_word_topics = True in order to get also topic distribution for each word
    image_topics, word_topics, phi_values = lda_model.get_document_topics(image['frequencies'],
                                                                          per_word_topics=True)
    output_image = cv2.imread(image['path'].__str__())
    for i, keypoint in enumerate(image['keypoints']):
        word = image['assigned_clusters'][i]
        # take the topic with higher probability
        topic = get_topic(word_topics, word)
        if topic is None:
            continue
        cv2.drawKeypoints(output_image, [keypoint], output_image, color=topic_colors[topic])
    cv2.imshow(image['filename'].__str__(), output_image)
```

results

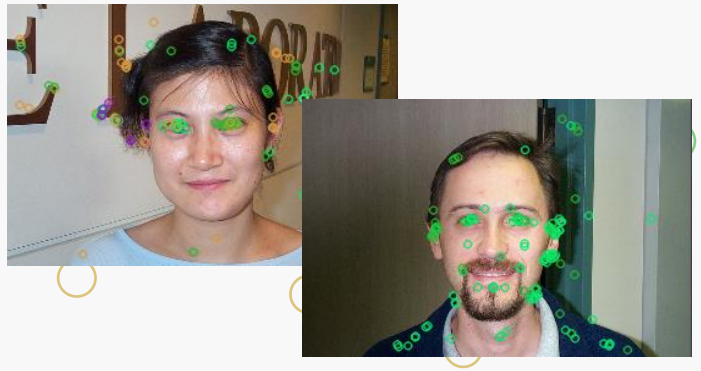
α = 0.01

α = 1

α = 10

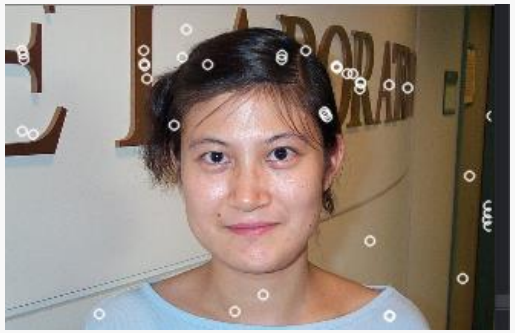
α = 50

k = 500, topics = 12, α = 0.01

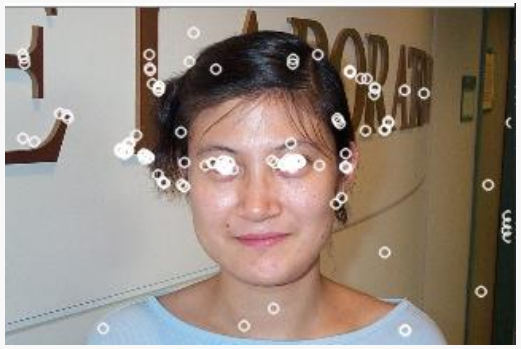


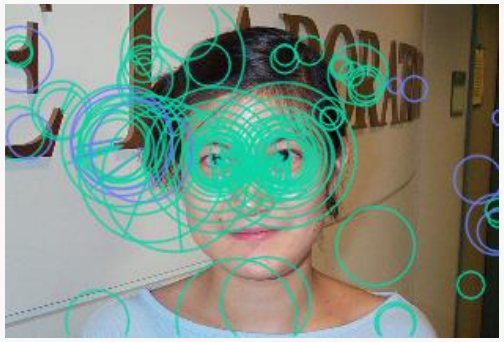
MSER vs MSER+ORB

MSER



MSER + ORB





MSER

MSER + ORB