TxMM A2 - Multimedia Clustering

Shaurya Gaur (s1116008)
Arina Schippers (s1036744)
Foteini Papadopoulou (s1122141)
A2-MM Group: 22

1 First impression

Question 1: Look at the images. Make three observations concerning the objects and scenes that you see depicted in the images (in computer vision, what people see in images is referred to as the "semantic content" of the image or "semantic properties").

Answer: The observations regarding the objects and the scenes that we depicted in the images are the following:

1. There are a lot of buildings (i.e. tourist attractions).
2. There are not always people in the images, but when they are, they are in the foreground.
3. Most of the buildings are white or beige.

Question 2: Now, look at the images again, and make three observations that are not related to objects and scenes, but rather are related to the style and quality of images.

Answer: The observations concerning the style and the quality of images that we depicted in the images are the following:

1. Some photos are clearly edited, but others are clearly photos snapped quickly by tourists.
2. Some photos have been taken vertically, but others horizontically.
3. Some photos don't capture the full building, but capture it clearly, while others capture a large landscape but not as clearly.

2 Clustering on simple features

Task 1: Implement the function `average_color` in the cell below using the information provided in the comments. (Hint: Remember that the original image can be accessed via `image_dict['original']`, and that you can convert this original image to a numpy array. (As a sanity check, make sure that you understand why the vector representation consists of three components in this case.))

Your code:

```
def average_color(image_dict):
  '''
  A function to compute the average RGB value of an image.
  First, average over rows to obtain an average value per column.
  Then, average over the resulting values to obtain one average
      value per color
  channel.

  :param image_dict: The dictionary containing the loaded image
  :return:          A 3-dimensional np array: 1 average per color
      channel
  '''
  image_array = np.array(image_dict['original'])
  return np.mean(image_array, axis=(0,1))
```

Question 3: Subsequently look at the images with indices 102, 153, 245, 439 and 555 and their average color by changing the index in the cell above. Then discuss and decide: Is this a good representation for color images or not (and why)? Describe a potential issue that you notice.

Answer: It loses a great deal of information, so we think this is not the best representation. A better solution may be to have fewer pixels than the original image, but not one big pixel. Moreover, it loses the image's semantic content. We retain no information about what is in the image. The average color can be very muted, which loses the bright color of the subject of the picture (specifically the gold building in image 153 and the bright face of the woman in image 102).

Question 4: Inspect the image montages. Do all clusters make equal sense to you, intuitively? Look at the 3-dimensional scatter plot again. Do you see a reason why/why not?

Answer: The clusters make good sense to us. Examining the 3-dimensional scatter plot, it seems like the six clusters are those with a similar amount of RGB values. This means that a single primary color (red, blue, or green) does not stand out as much, and most images in a cluster have a similar light level. It also makes sense cluster two has a lot of blue colors and this is also seen in the plot.

Task 2: Change the color histogram function below so that it uses 32 bins per color channel. Then run the second cell and inspect the results.

Your code:

```
def color_histogram_32bins(image_dict):
  ’’’
  Compute the normalized color histogram binned into 32x32x32 bins
      from the RGB image.
  :param image_dict: The dictionary containing the loaded image
  :return:          A 32768-dimensional np array
  ’’’
  # extract a 3D color histogram from the RGB color space
  im = image_dict[’cv2’]
  hist = cv2.calcHist([im], [0, 1, 2], None, [32,32,32], [0, 256,
      0, 256, 0, 256])
  # normalize the histogram
  hist = cv2.normalize(hist,hist)
  # return the flattened histogram as the feature vector
  return hist.flatten()
```

Question 5: Do you think that more bins helped us in the clustering? Why?

Answer: We think that adding more bins helps in the clustering, since similar images of the same monuments are more likely to be clustered together with 32 bins. For example, in cluster 2, there are more similar images of the Moulin Rouge with 32 bins, and in cluster 3, there are more black and white images of the Eiffel Tower together. With three bins, different buildings are more likely to be clustered together if their pictures include a blue sky, but with more bins, these are separated.

Task 3: Perform clustering with 12 clusters on the `chist_32bins_feature_vectors` and display the montages for these clusters in the cell below.

Your code:

```
y_kmeans_chist_12 =
    perform_k_means_clustering(chist_32bins_feature_vectors,12)
show_images_in_clusters(y_kmeans_chist_12, sample_pathnames,
    sample_images)

y_kmeans_chist_24 =
    perform_k_means_clustering(chist_32bins_feature_vectors,24)
show_images_in_clusters(y_kmeans_chist_24, sample_pathnames,
    sample_images)
```

Question 6: Which are the major cluster changes when increasing the number of clusters to 12 and then to 24? What does this tell us about our dataset's images? (Hint: You don't need to point out every small detail. Please try to answer this question at a high-level.)

Answer: When doubling the number of clusters to 12, similar pictures of the same building are in the same cluster. However, after doubling again to 24 clusters, these are more likely to be separated like for the Eiffel tower building and Moulin Rouge. This tells us that our dataset's images fit best into between 12 and 24 clear categories because of the similarities that there are between some images e.g. buildings, background.

Question 7: Looking at the semantic content of the clusters, which of the choices of the number of clusters do you find to be best-suited for our dataset using the color histogram feature vectors? Why?

Answer: Our group finds that 12 clusters are best suited for our images with color histogram feature vectors. With 24 clusters, a few clusters are very cohesive in their images while others seem more random, whereas with 12 clusters, each cluster seems like a more cohesive collection on average, containing images of the same monuments.

Question 8: Do you think that all images are in the 'correct' cluster? Do you think 8 is a good number of clusters for HOG features, considering the images' semantic content? Why?

Answer: While most images are in the 'correct' clusters, there are some outliers; many images of the Moulin Rouge windmill are in cluster 4, but a few of these images are also in various other clusters. A larger number would be ideal to cluster the dataset's images, since some clusters seem to contain 'sub-clusters' of different buildings. For example, cluster 2 has several images of the Arc de Triomphe as well as those of the Pantheon, and these could be separated into their own clusters.

3 Clustering using neural representations

Task 4: Now systematically vary the number of clusters in the cell below. Try out both more and fewer clusters. Then answer the questions below. (*No need to copy anything here*)

Question 9: Which numbers did you try out? Which number of clusters worked best for you? How did you make this decision?

Answer: The numbers that we tried out were 6, 12, 18, 24. The best number of clusters that worked best was 18.

1. For $n = 6$ clusters, half of the clusters are only of similar images, but the rest contain large collections of images which could easily be split into smaller clusters.

2. For $n = 12$ clusters, most of the cluster were good and portrayed one building. However, some cluster could be split into separate clusters. Such as cluster one has very similar images but cluster five has images of people, paintings and more. This cluster could be split up.

3. For $n = 18$ clusters, the clusters seem to accurately depict the landmarks. Only one of the clusters is a little 'messy', but they wouldn't fit into a separate cluster anyway.

4. For $n = 24$ clusters, the majority of clusters perform really well and we can say that each cluster represents a building, but there are a few clusters that contain a pretty low number of images. Some clusters contains random images. We think that 24 clusters overfits the data, as some clusters (i.e. 21, 22) could be joined.

We used 6,12 and 24 numbers of clusters because we wanted to compare them with the previous implementation of clustering on previous features and we used 18 clusters because we saw that 12 clusters performed well but some of them contained images that would be split more.

Question 10: What does this tell us about the neural representations?

Answer: Our clustering using neural representations tells us that this method seems to be the most effective when compared to methods like HOG and color histograms, since it uses both color and structure data. We've reached this conclusion since the clusters using this method seem to be a better fit to the data.

Question 11: Reflect on the "clustering bias". For each of the four image representation approaches, answer the following four subquestions:

1. Which semantic properties play a decisive role when clustering the images in the dataset using this representation? By which semantic properties are the images grouped? Remember "semantic properties" refer to what people ("human users") see in images

2. Which semantic properties play no decisive role?

3. Why? What causes the phenomena you observed in the first two subquestions?

4. When (i.e. for which kind of clustering problem) would you use this image representation?

Answer:

Average color:

- Semantic properties: Decisive role: Light level
- Semantic properties: No decisive role: Shapes, structure
- Why?: Averaging an image's colors into one led to a muted average color for most images, losing distinct colors. As a result, most average colors had even values for Red, Blue and Green content. Thus, the algorithm clustered images based on their overall light level, so light images were clustered together in cluster 3 and dark images were clustered together in cluster 4.
- When to use: When trying to detect how light or dark an image is (e.g. if an image was taken in day or night, inside or outside).

Color histograms:

- Semantic properties: Decisive role: Distribution of colors
- Semantic properties: No decisive role: Image structure
- Why?: The pixels of the image are downsized. After this the pixel will be put into a bin with how much of which RGB value they contain. The histograms do not retain where specific RGB values occur within an image - only how often they occur.
- When to use: If you want more specific information of what colors the pixels in the image consist of.

HOG:

- Semantic properties: Decisive role: edges, shapes, structure, semantic content (buildings)
- Semantic properties: No decisive role: colors, light level

- Why?: When there is a sudden change of color of a pixel between its direct 'neighbor' pixel and a specific direction, an 'edge' will be detected including the direction of this edge.
- When to use: When trying to detect images that contain similar geometric patterns and shapes (e.g. buildings) regardless of time of day.

Neural representations:

- Semantic properties: Decisive role: Structure, color, semantic content (buildings AND people)
- Semantic properties: No decisive role: light level
- Why?: The VGG19 model is trained to classify images, and is fed a cropped image as its input. Instead of users feeding partial data about an image manually, the large and dense network (with 143 million parameters) has been trained to model various complex features of images, and condense this to smaller feature vectors.
- When to use: When you want more information of the high level features of the data, not just a single one of them. This is the most effective in clustering images well, so we would recommend using this if you want a general classification or clustering. However, if just one of the features is required, another clustering method might be more time effective.

Question 12: At the beginning of the assignment, we told you that the data set contained Paris landmarks. However, now you have carried out a clustering study, you have gained more insight into the semantic content of the images in the data set. What have you discovered?

Answer: This dataset contains pictures of Paris landmarks like the Moulin Rouge, the Louvre, Eiffel Tower, the Grande Arche, the Arche de Triomphe, Sacre Coeur, and the Pompidou centre. There are also many images of paintings, statues, people, street art, and more.