TxMM A2 - Multimedia Clustering

Students: Janneke Nouwen (s1101750) Daan Brugmans (s1080742)
A2-MM Group: 31

Note: make sure to hand in a pdf of this file

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:
Janneke: In the ‚eiffel‘ image I see a statue of a person and a ‚real‘ person behind the statue. In the notredame image I see trees alongside a canal. I see window panes in the ‚louvre‘ image.
Daan: Some of the images do not contain the entirety of what they are labeled as; the invalides image only contains a part of the building. The image for the Arc d’Triomphe is not an image *of* the Triomphe, but an image *on* the Triomphe. Furthermore, some of the images do not have the main focus of the image in the center.

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:
Janneke: The details of the sacrecoeur are not very clear due to the low quality of the image. The notredame image has quite a fantasy/dreamy like style due to the color of the sky and the positionings of the trees and the notredame. The eiffel image has quite an artsy style due to the use of black and white and the positioning of the statue, person and eiffel tower.
Daan: First, some of the images taken do not necessarily have its label as its focus. The image of the Pompidou is taken inside of the Pompidou and focuses on a person posing. Second, some of the images’ quality, like the onve for the Louvre, are lower than one might prefer: the image is not very clear. Lastly, the image of the Musee Dorsay is taken stylistically, playing with light, which may not be a very accurate representation of what the museum actually looks like.

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
  '''
  avg_row_array = []
  for row in image_dict['cv2']:
    avg_row_array.append(np.array(row).mean(axis=0))

  avg_row_array = np.array(avg_row_array)
  avg_array = avg_row_array.mean(axis=0)
  return avg_array
```

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:
Janneke: A lot of average colors are very similar gray-ish values This gives very little information about the content of the image. All rgb values dilute to gray.
Daan: Most average colors tend to go towards a grey tint. This is likely because, as we calculate the average values of a color channel, the values of the color channels tend to be neither high nor low, but an inbetween value. When given these „middle“ color values, we get a shade of grey.

So we do not think that this is a good representation for color images

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:
Janneke: The clusters seem mostly based on how dark the lighting is in the picture. The images. This is visible in the plot, as all clusters are depicted on an almost straight line. This line is likely to represent how dark the clusters are. Cluster 3 is the cluster with the darkest images and it is the cluster with the lowest value of all colors in the plot. Next comes cluster 1. The cluster with the higest average color values is cluster 0. This cluster indeed has the lightest images. There are a lot of different buildings in each of the clusters.
Daan: Not all clusters make equally intuitive sense to me. The distinction between some of the clusters is clear; for example, cluster (index) 3 mostly contains images of buildings taking at night. These images contain a very dark night sky and the lights cast upon the building, which makes images of such pictures often a combination of black and yellow or white. However, the differences between cluster(s) (indexes) 0, 2, and 4, for example, do not intuitively make sense to me. They all seem to contain images taking during the day, often featuring a blue or grey sky, and the relevant object as the focus of the image. These clusters, to me, seem to all mostly contain images that are a combination of blue and grey/marble. When looking at the 3D scatter plot, I can see why that is the case. First, the cluster(s) (indexes) 0, 2, and 4 are all close to one another within the feature space, implying that they are more similar to each other than to other clusters. Furthermore, cluster (index) 3 is the furthest removed from those previous clusters, implying that it is the most dissimilar to those clusters. This is because the members of cluster (index) 3 has low values for red, green, and blue pixels. Meanwhile, the members of cluster(s) (indexes) 0, 2, and 4 generally all have high values for red, green, and blue pixels. Additionally, the spread of the data for cluster (indexes) 2 and 4 is relatively small when compared to the other clusters. Knowing this fact, alongside the fact that these two cluster (indexes) are right next to one another, would imply that the members of the clusters are very similar.

We agree on how we see the clusters. They make some sense, but do not help to distinguish between the buildings, only the brightness.

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, [3,3,3], [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:
Janneke: I do not see a clear distinction between the clusters. Some clusters are darker, but that distinction was more clear with less bins. The new clusters do not make a better distinction between the difference buildings. Cluster 1 does show some more yellow than the other clusters.
Daan: No, I do not think that the increase in bins has helped us in the clustering. Personally, intuitively, I feel that the 3x3x3 binning has resulted in clusters whose members are more closely related to each other than to the 32x32x32 binning. I think this happened because a binning of 32x32x32 means that there are 32 different bins for each color channel, meaning that every bin contains a range of 256 / 32 = 8 values, and we have 32 * 32 * 32 = 32.768 different combinations of such bins. Because these bins are so small, the 32x32x32 binning is very alike to simply analyzing the images without binning. In contrast, the 3x3x3 binning results in more general bins for the color channels, meaning that each bin is more different from every other bin. This would give every bin in the 3x3x3 binning more "value" than a 32x32x32 bin. Simply put: the 32x32x32 binning does not generalize the color channels enough.

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, n_clusters=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, n_clusters=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:
Janneke: When looking at individual clusters, they contain more of the same building, so there is less variation within clusters when the amount of clusters increases. Because the amount of cluster increases, there are also more clusters that are small and seem irrelevant. It keeps being hard to accurately separate the different buildings into the correct clusters. There exist a lot of images with similar colors and features that depict different buildings in our dataset.
Daan: When increasing the number of clusters from 12 to 24, the major perceived change is that the color schemes of every cluster grow to be more specific. That is to say, when increasing the number of clusters, the color palettes of the images within a cluster grow to be more uniform and specific. Nonetheless, some of these clusters seem to have very similar color palettes when compared to one another. This could tell us that the images in our dataset can initially be split into very few clusters with major color palette differences (like night/dark images, marble images and bright/day images), but can only be split on very specific combinations of colors when increasing the cluster count.

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:
Janneke: I find 12 clusters best suited. Clustering into 24 clusters creates too many small and irrelevant clusters. When clustering into only 6 clusters, there is less of a distinction made between the different buidings than with 12 clusters. Still, for 12 clusters this is far from perfect, but it does come closer in my opinion.
Daan: I think that 12 clusters suit the data the best. This is because the dataset contains images of 12 different labels. In an ideal situation, we would be able to find a generalized color histogram for every class label that is capable of identifying what type of attraction an image in our dataset is by only looking at the colors of the image. In reality, our clusters are not perfect like that, but it may be possible to improve the model by e.g. giving it more data. The one caveat to this is that the "general" class can contain many types of images that are unrelated to one another, yet in the same class. If we wanted to differentiate between the types of "general" images, 24 clusters might be preferred, but as this is not the case in our dataset, I think 12 still suits best.

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:
Janneke: Definitely not all images are in the ‚correct' cluster. Using 8 clusters might also not be the best idea, as the dataset contains 12 different landmarks. So 8 might be too low to accurately cluster these images based on the building they depict. We would need at least 12 clusters. Else there will not be enough clusters to place each landmark in a separate cluster.
Daan: I do not think that all images are in the 'correct' cluster. These clusters are based on the HOG values of the images' pixels, so our clusters are basically purely based on the degree to which the colors of pixels in an image change. This on its own doesn't seem to capture a lot of information about the things we are trying to cluster. I do not think that 8 is a good number of clusters, since we are trying to cluster on 12 different classes. This would automatically mean that we must group some of our classes in with a cluster that belongs to another class. I would suggest the use of 12 clusters for this reason.

We agree with each others answers.

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:
Janneke: I tried 4, 10, 12, 15, 18, 24. When using 4 clusters, there is way too much variation within the clusters. Only the last cluster has one consitent building. When using 10 clusters, the clusters 0, 1 and 4 seem correct, as they each cealrly represent a different building (The triomphe and the eiffel tower). However, the other clusters have quite a lot of variation. This is a bit similar when using 12 clusters. Here, clusters 0, 1, 3, 4, 7, 8 and 9 are very clear, but the other clusters are not. When using 15 clusters, only clusters 5, 7, 9, 10 and 14 are very clear, even though the amount of clusters increased. When using 18 clusters, clusters 3, 4, 7, 9, 13, 14 and 17 are clear. This is just as much clear clusters as when using 12 clusters, but that means that this method contains more messy clusters than when clustering is done with 12 clusters. When using a larger number of clusters, more and more small clusters appear. Especially when using 24 clusters. This does not approach the goal of clustering the buildings into their own cluster and makes it messier. In summary, i would use 12 clusters.
Daan: I tried out 6 clusters and 18 clusters, that is, 50% less and 50% more than 12 clusters. 6 clusters was too few: multiple Parisian landmarks were clustered together, when they shouldn't have been, although this clustering seems to make sense. 6 clusters had a cluster that consisted of images that were "taken inside of a marble building" and a cluster that could be described as "picture of a building with gates/gaps". This shows that the model was

able to still generalize the images well, just that there weren't enough clusters. 18 clusters seemed to be a bit too many: while most clusters only consisted of a particular landmark per the classification, some clusters seem to focus on a certain aspect of an image's composition. For example, there were clusters for "the Eiffel tower at night" (because of the Eiffel tower's lights) and a cluster that consisted of "diamond structures", like the glass of the Louvre and the underside of the Eiffel tower. I would conclude that a higher amount of clusters might be interesting when analyzing the composition of the image. Despite this, I would still say that 12 clusters worked the best, as the 12 clusters mapped the best to the 12 classifications of our dataset.

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

Answer:
Janneke: I notice that the model is quite focussed on shapes. The square lines of the eiffel tower are often clustered with the glass squares of the louvre for example. Arcs are often clustered together as well. It is better in recognising the buildings in different setting than the other two methods tried before. It is less influenced by the lighting and the angle, but it is definetely not perfect.
Daan: I think that the neural representations of the images are very telling of the contents of the image. Despite having to represent images as an embedding, and then having to cluster those images with KMeans, the most clusters clearly belonged to a particular landmark as recognized by a person.

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:

1. Semantic properties: Decisive role:

Janneke: Lighting (if the picture was taken during the day or the night) and the color of the most prominent object on the image (like the green color of a tree in front of a building)

Daan: Lighting seems to have had a great influence on the clustering. For these clusterings, I personally notice 2 things that separate the clusters: the color of the sky (blue, grey or black) and how well-lit the object in the foreground of the image is.

2. Semantic properties: No decisive role:

Janneke: Shapes of the objects in the image, amount of items in an image, large contrasts within an image

Daan: The shapes of the objects in the image do not seem to have an impact on the clustering.

3. Why?:

Janneke: Lighting influences the average color as a dark background skewes the average color to a darker color, and the other way around for a very blue or bright sky. It also influences where shadows fall or if light falls on objects. The color of the most prominent object is a big determinant of the average color, as the color of this object covers most of the image and therefore has a large influence on the average color. Shapes do not matter, as the average of all pixels is taken without taking the placing of pixels in mind. The same goes for the amount of items in an image. It does not matter if there are two smaller white buildings or one white building twice the size. Large color contrasts are also averaged out, because for example the average of white and black is gray.

Daan: The clustering here is fully based on the average color of the image. Since a lot of these images were taken outside, the sky often covers a significant part of the image, and since the sky is most often a single, mostly uniform select color, the color of the sky has a major influence on the average color. Of course, the object in the foreground also has a major influence.

4. When to use:

Janneke: When you want a representation of an image of a low-compexity. It is relatively quick, but does not give much information about the picture, except if you are only interested in the most prominent color on an image.

Daan: I wouldn't. Most of the time, the average color of an image is a shade of grey, which doesn't say a lot about an image. If I wanted to cluster on color, I would use the next feature representation.

We discuss and agree that there could be some specific use cases where this technique can be somewhat helpfull.

Color histograms:

1. Semantic properties: Decisive role:

Janneke: Large color contrasts within an image. Brightness of an image

Daan: Color palettes seem to play a major role here. Clusters seem to be based on certain groups of colors, what colors are prominent in the image.

2. Semantic properties: No decisive role:

Janneke: Details, Placings of objects in an image, Shapes

Daan: Once again, the shapes of the object in the image does not seem to have an effect on the clustering.

3. Why?:

Janneke: Pixel groups with large contrasts in color are collected into different bins, which can be used for clustering. The brightness of an image effects the colors that will be captured (darker for a picture taken at night). Details are not captured, as they will not show up in the roughness of the color bins. Placings of objects also do not matter, as we

only count how many pixelgroups we have per bin and therfore disregard the place of these picture groups. This is also the cause for shapes not being captured.

Daan: For this clustering, the only thing that matters are the colors in the image, since only a histogram of colors is what represents an image. This means that the things that the image depicts do not matter for the purposes of clustering.

4. When to use:

Janneke: When the average color does not give enough information about the picture, but color or contrast is a very important aspect that you want to cluster/classify on.

Daan: I would use this feature representation if I wanted to classify images by their color palettes, especially for a dataset where most images consist of a set of few different colors.

HOG:

1. Semantic properties: Decisive role:

Janneke: Edges with high contrast, placing of objects in an image

Daan: The location and structure of the focus of the image seem to have a major role in the clustering.

2. Semantic properties: No decisive role:

Janneke: Colors, brightness (as long as contrasts are still clear)

Daan: The specific colors of the image seem to have no effect on the clustering, only the contrast between colors of the image.

3. Why?:

Janneke: Edges with high contrast are represented using gradient vectors. The placing of objects matter, as their edges are stored in the gradiet vector in the correct order. The color does not matter, as you will see no difference between a green or a red tree, as long as the contrast with the background is visible. The same goes for the brightness. The representation of a polar bear in the snow will not look much different from a brown bear on the forest floor.

Daan: This clustering is mostly based on the edges of an image. As such, what determines the clustering, is the location at which pixels change color significantly. Because of this, the position of the focus of the image, the shape that that object is and the colors of its surroundings compared to it play a vital part in determining a clustering. However, since we do not take the colors themselves into account, they seem to have little to no influence on the clustering.

4. When to use:

Janneke: When the things you want do distinguish have very different shapes. It could also be used to distinguish between patterns.

Daan: I would use this feature representation if I wanted to recognize particular shapes in an image, irrespective of their size and location within the image.

Neural representations:

1. Semantic properties: Decisive role:

Janneke: Shapes and colors of objects. It depends highly on what the model was trained on. Here it was trained on imagenet, so it has seen a lot of different types of images.

Daan: The shapes within the image seem to have the most significant effect for the clustering. Image colors also have an important role.

2. Semantic properties: No decisive role:

Janneke: Image resolution

Daan: The location of the focus of the image seems to not matter for this clustering.

3. Why?:

Janneke: The model is trained to distinguish a lot of images. It can recognise different shapes and different colors. This includes for example arcs, squares, circles, and more complicated shapes. It also includes brightness of images, if the image is black-and-white, and large color contrasts.
The image resolution does not have a decisive role. All images are resized, as the model expects a specific resolution.

Daan: The neural representations cannot easily be explained, as they are high-dimensional representations of the image, but this also means that they encode a lot of information. These representations are the way in which the neural network has expressed the images and, assuming the model has generalized well enough, the expressions should contain general information about the contents of the image. The model has learned to recognize the focuses of the images, irregardless of where those focuses are in the image. Shapes are often a basic building block for generalizing the contents of an image. These facts should explain why the shape of an image matters a lot for this clustering, while the location of the focus of the image doesn't.

4. When to use:

Janneke: When a detailed representation is needed and both colors and shapes are relevant to the goal.

Daan: I would use this feature representation if I wanted to cluster on the general shapes of the focus of an image.

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:

Janneke: Not all images contain the building of the class it belongs to. Some contain the view from the top of building. Some contain other items as the most prominent item (e.g. people, trees) while the building is in the background. There also exist a lot of variation in the brightness. Some were taken at night, some where taken during the day. Per building, a lot of different angles are showed in the images. Some where taken from below, some from further away. Some images were taken of details inside te building, some images contain details of the outside of the building and some images contain the entire building.

Daan: I have discovered that the composition of an image of a particular landmark greatly influences how a model would cluster it: pictures taken inside of a landmark might as well be completely different landmarks than pictures taken of the landmark. I learned that a set of images of the same landmark can look vastly different, with different sets of color pallettes. For example, there often seemed to be a big difference between the Eiffel tower during the day and during the night. Lastly, I learned that the "general" landmark consists of many very different images, and that this fact can lead to messy clustering.

We agree that the same image can represented in various ways, having an important effect on the clustering. We also agree that there is a lot of variation within this dataset for similar landmarks.