DATA MINING COURSEWORK 2

BY

MEGHANA KUMAR

20093575

Text Mining

Average Runtime: 5.78832 seconds for entire Question 1

Question 1.1

Metric	Number	Value
All Possible Sentiments	5 sentiments	['Neutral' 'Positive' 'Extremely Negative' 'Negative' 'Extremely Positive']
Second Most Popular Sentiment	9917 tweets	Negative
Date with Greatest Extremely Positive Tweets	545 tweets	25-03-2020

Before Data Pre-processing

0 @MeNyrbie @Phil_Gahan @Chrisitv https://t.co/i...
1 advice Talk to your neighbours family to excha...
2 Coronavirus Australia: Woolworths to give elde...
3 My food stock is not the only one which is emp...
4 Me, ready to go at supermarket during the #COV...

After removing non-alphabetical, multiple whitespace, and converting all to lowercase

0 menyrbie phil gahan chrisitv https t co ifz f...
1 advice talk to your neighbours family to excha...
2 coronavirus australia woolworths to give elder...
3 my food stock is not the only one which is emp...
4 me ready to go at supermarket during the covid…

Question 1.2

Attribute	Values
Total Number of All Words	1350959
Number of Distinct Words	80071
10 Most Frequent Words	[('the', 44919), ('to', 38509), ('t', 29901), ('co', 24153), ('and', 24107), ('https', 24007), ('covid', 23238), ('of', 21570), ('a', 19964), ('in', 19359)]

Total Number of All Words - Removing Stopwords and <2 char words	777813
Total Number of Distinct Words - Removing Stopwords and <2 char words	79252
10 Most Frequent Words - Removing Stopwords and <2 char words	[('https', 24007), ('covid', 23238), ('coronavirus', 18210), ('prices', 7959), ('food', 7182), ('supermarket', 7096), ('store', 6932), ('grocery', 6284), ('people', 5625), ('amp', 5198)]

Interpreting the FreqDist Object
<FreqDist with 80071 samples and 1350959 outcomes>
<FreqDist with 79252 samples and 777813 outcomes>
Samples = unique distinct words
Outcomes = total words

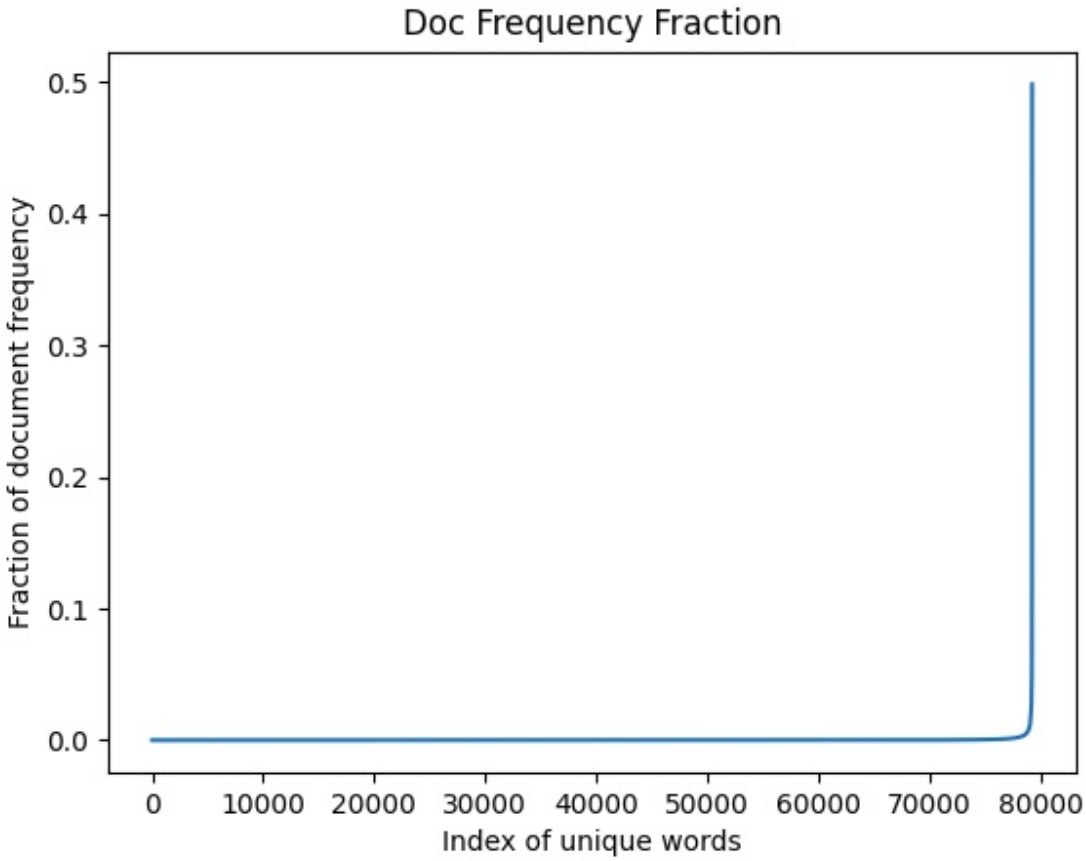
Observations: Removing stopwords is a crucial step in data processing because it retains only the relevant words that will actually be helpful in discriminating between the documents to build an efficient model for sentiment analysis. Before the stopwords and small words are removed, the top 10 frequent words are all mostly stopwords which isn't helpful for document analysis.

Comments: Using a simple split() function to tokenize *instead* of the nltk.word_tokenize on the pandas Dataframe reduced the run time significantly (by about 6 seconds)

The values are slightly different when the tokenization is done through word_tokenize and it is mentioned below. split() provides a close approximation though.

Attribute	**Values**
Total Number of All Words	1351476
Number of Distinct Words	80067
Total Number of All Words - Removing Stopwords and <2 char words	777580
Total Number of Distinct Words - Removing Stopwords and <2 char words	79248

Question 1.3



Observations: It is not practical to have the 80000 words in the term-document matrix, so it makes sense to pick the documents that actually are important to make the distinction. By the tf-idf logic, the terms that appear in almost all documents are not as useful because they are not useful to make a distinction.
An visual approach is to simply cut off the graph when it hits the inflection point and exclude those terms greater than when the sudden increase in probabilities happens.

Since this is a 5-class classification, a random guess would have an average accuracy of 20%. So, using that as a rough estimate, any term that appears in more than 20% of the documents can be left out of the document matrix since it will not be helpful for discrimination. Semantically, it means that if it is present in more than 20% of documents in a presumably balanced dataset, then it is not a unique term for that class.
Number of terms appearing in greater than 20% of documents is 3
Number of terms appearing in lesser than 20% of documents is 79249

To trim the document matrix even more, it is also experimented by cutting off words that appear in more than 1% of documents
Number of terms appearing in greater than 1% of documents is 237
Number of terms appearing in lesser than 1% of documents is 79015

Question 1.4

For this exercise, the vectorization is done using CountVectorization.
This produces a sparse representation of the term-document matrix.
The data is not split into train-test split and an analysis the model is trained directly on the corpus.

WITHOUT CLEANING DATA

Accuracy	74.75%
Error Rate	0.2525 (25.25%)

To compare the error rates of this exercise, an experiment is run to train the model after tokenization and removing stop words and small words < 2 characters. These results are mentioned below.

AFTER CLEANING DATA

Accuracy	76.96%
Error Rate	0.2304 (23.04%)

The error rate is reduced from 25.25% to 23.04%. This reiterates the good practice of tokenization and removal of stop words to extract the true semantic meaning of the document and removing unnecessary words.

This error rate is still quite large which indicates that this is a sophisticated problem that requires a more complex model, or more data to derive a better performance.

- Please Turn Overleaf -

Image Processing

Question 2.1

avengers_imdb.jpg

Original Image

Height	Width	Channels
1200	630	3

The threshold used for Black & White is 128 pixel value or 0.5 when values are normalized from [0,1] using the skimage library.

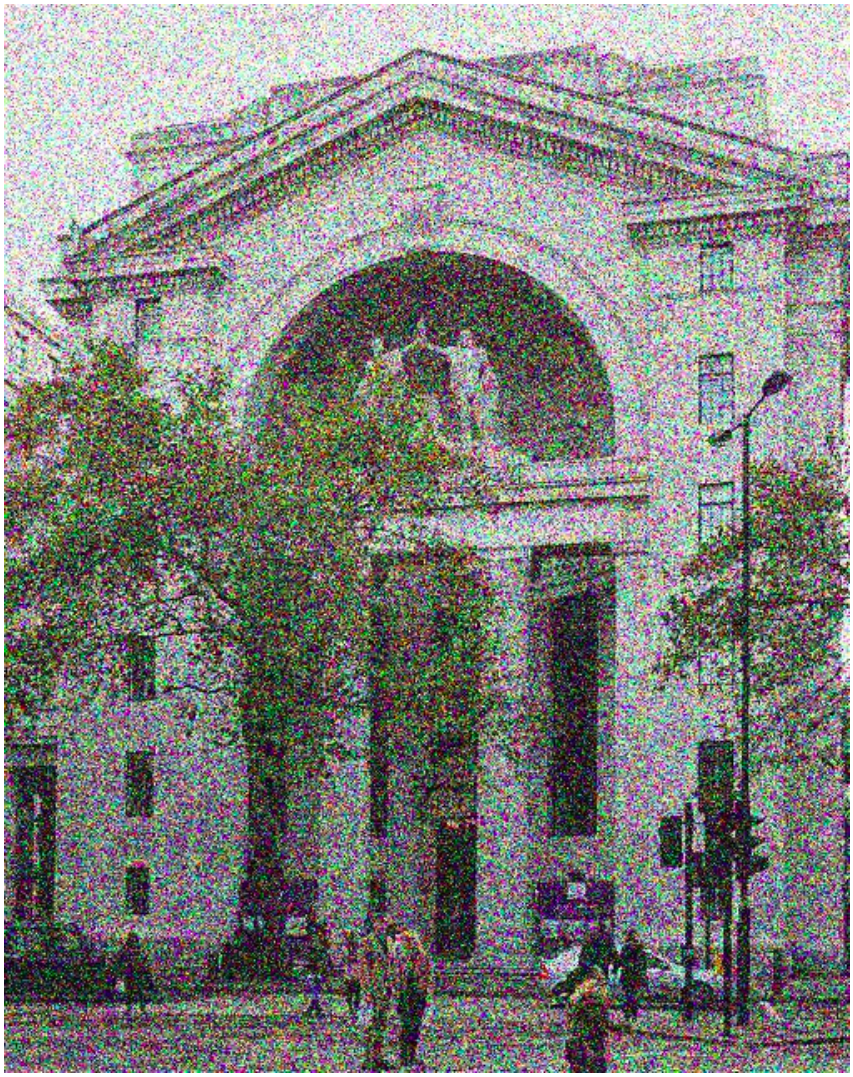
GREYSCALE

BLACK AND WHITE

Usefulness: Grayscale is used to reduce the number of bits required to store the image. (16 bits to 8 bits) And binary, even more so - A single bit is enough to store each pixel. For movie posters like this, if space is a constraint, in review websites or booking websites, images can be stored as grayscale or binary to render and store images efficiently. The images are still recognizable and easily identified because they are popular and well-associated in our brains, even without color.

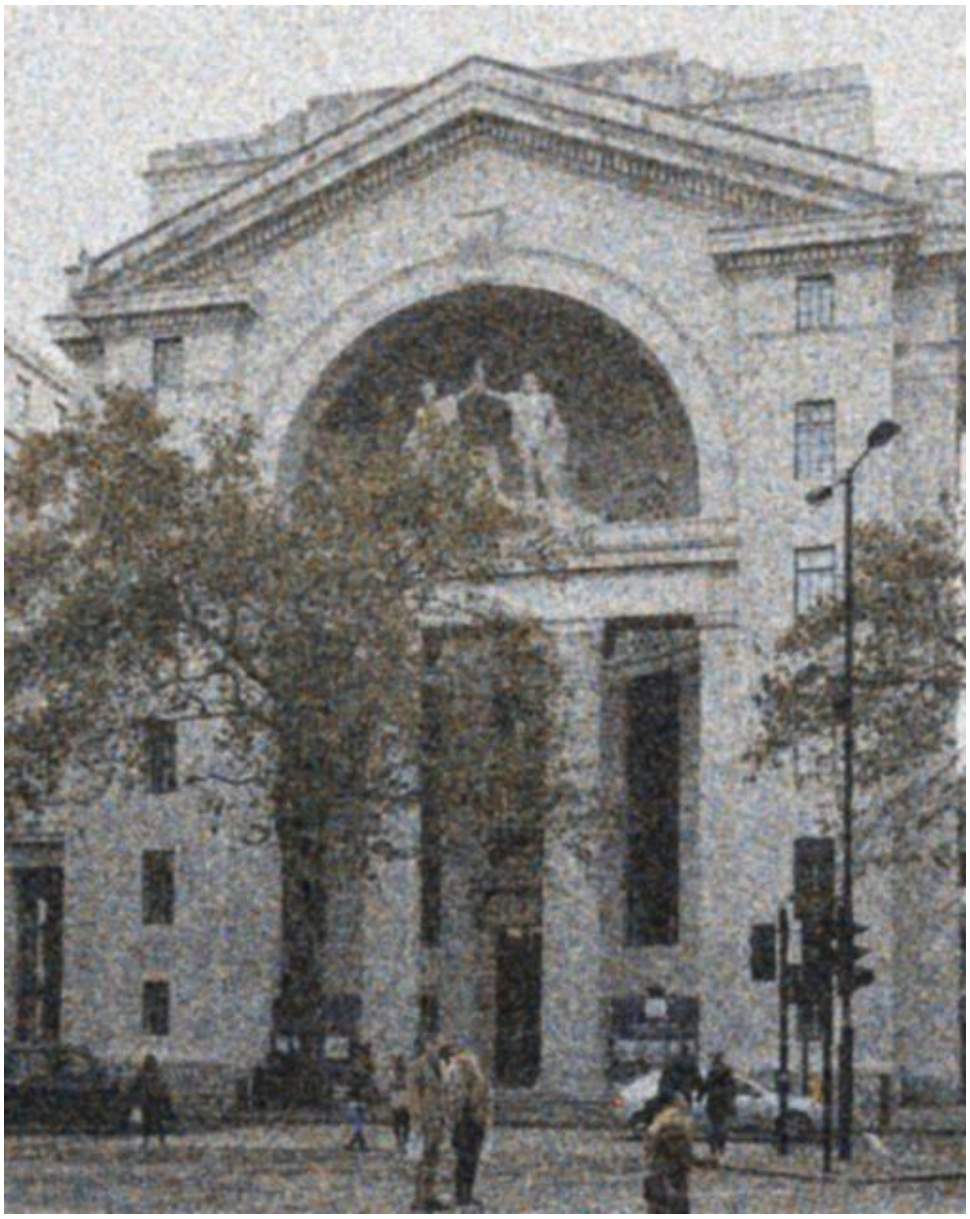
Question 2.2

Gaussian Random Noise
Variance = 0.1
Library used = skimage.util.random_noise



Observations: The image is perturbed using a gaussian noise, color information is lost for pixels, and noise is introduced, yet the picture retains its structural integrity and there is scope for reconstruction. Filters and masks will be explored and their results will be compared in this exercise.

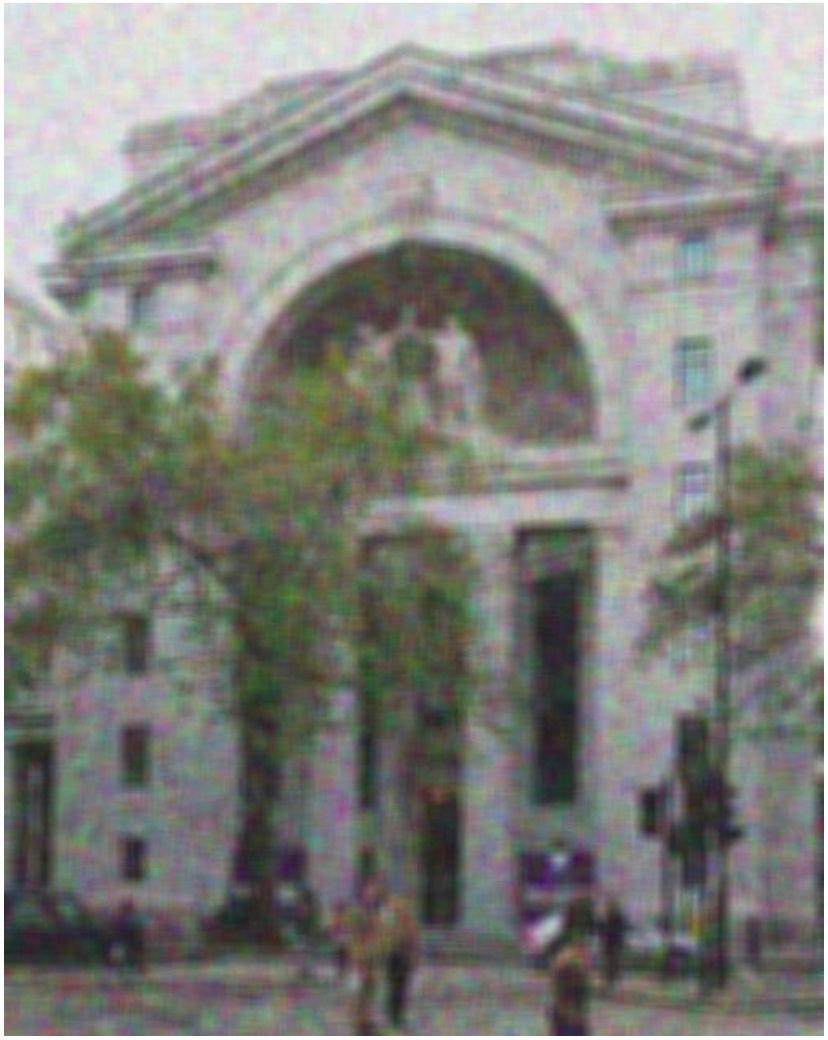
Gaussian Filter
Sigma = 1
Library used = scipy.ndimage.gaussian_filter



Observations: Gaussian filter reconstructs the image and approximates lost pixels using a normal distribution on its neighbouring pixels. While there is a slight blur, it is still sharper than the image obtained by the uniform filter. But the reconstruction severely compromises on the color channels.

<u>This image is obtained when the uniform filter is applied on the perturbed image directly.</u>

Uniform Filter
Sigma = 1
Library used = scipy.ndimage.gaussian_filter

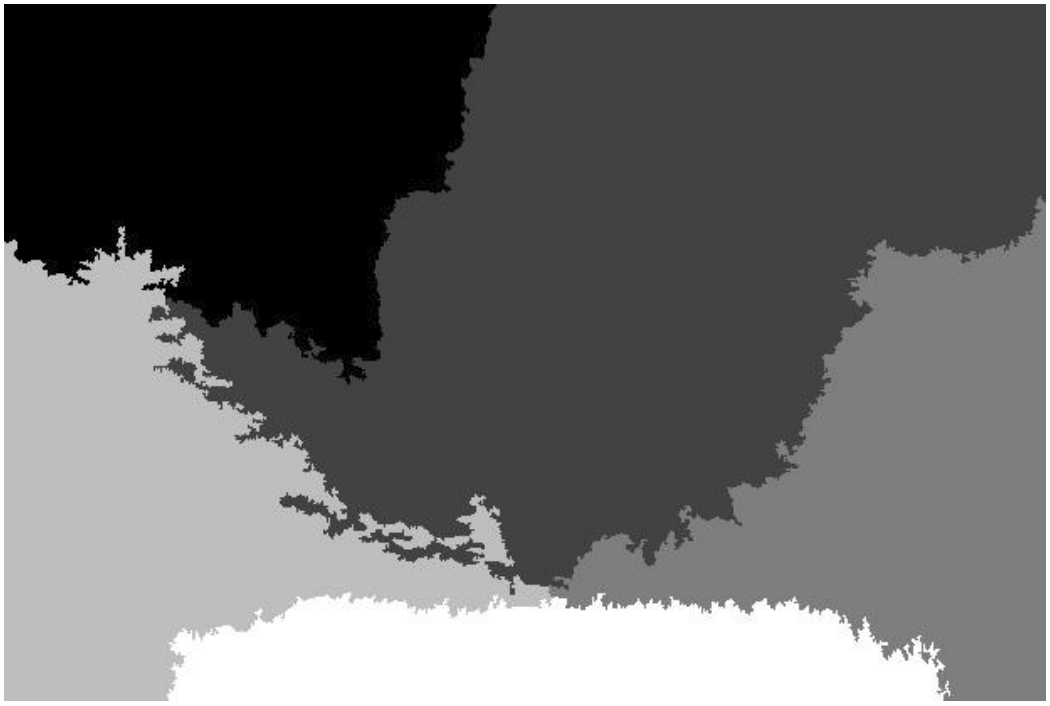


Observations: Uniform filter reconstructs the image but introduces a significant blurring. But the one thing it wins over the gaussian filter is that it retains most of the color information.

Usefulness: Random Noise is useful for performing experiments on which filter works best for the use case.
Gaussian filters and uniform filters used in the correct situation can be very useful in reducing noise.
Gaussian filter retains structure while color information may be lost. Useful for architecture photos - like Bush House
Uniform filter retains color while blurring the structure. Useful for paintings, art etc

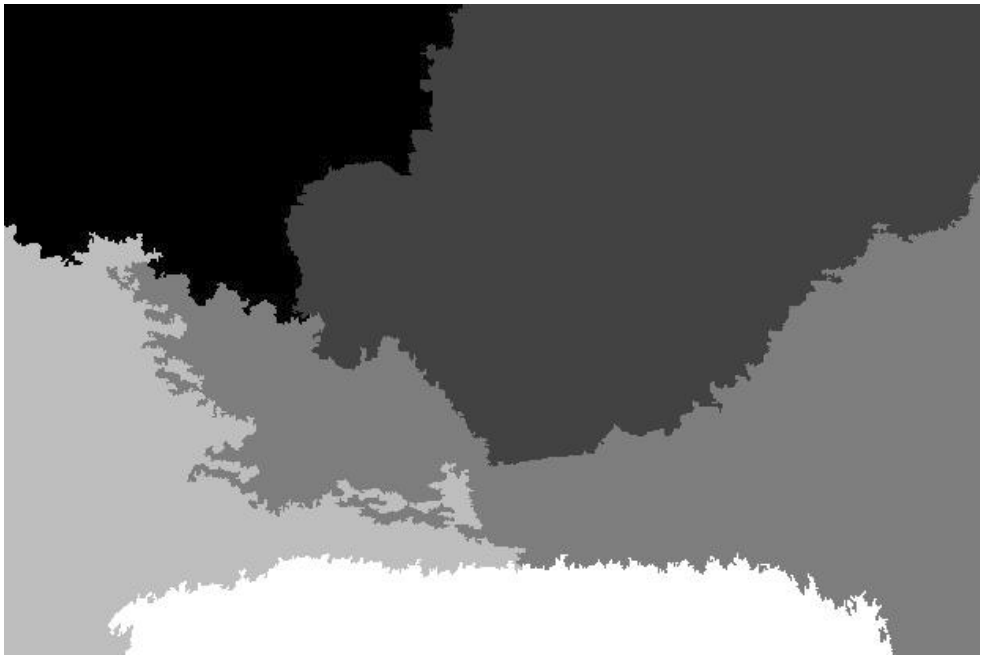
Question 2.3

<u>K-Means Segmentation - default values</u>
Number of Segments - 5
Compactness - 20
Library used = skimage.segmentation.slic & mark.boundaries



Superimposed Image

<u>K-Means Segmentation - parameter tuning</u>
Number of Segments - 5
Compactness - 15
Library used = skimage.segmentation.slic & mark.boundaries



Superimposed image

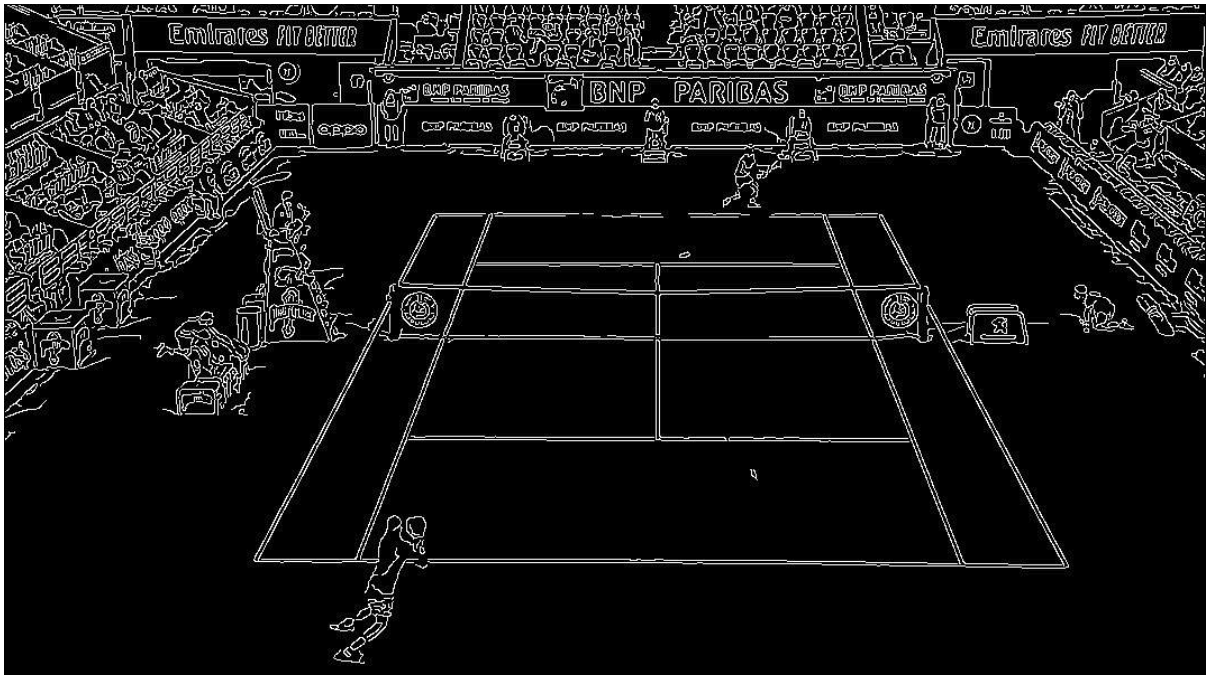
Observations: Experimenting and tuning compactness helps separate the ground and sky better which did not happen with the default values. This parameter depends strongly on image contrast and on the shapes of objects in the image.

Usefulness: When sophisticated segmentation algorithms are too resource-intensive, k-means acts as a simple tool which can help segmentation tasks in an unsupervised context. Annotation is one of the biggest challenges to segmentation, and hence clustering algorithms provide a simpler and fairly effective way to perform the unsupervised task when no annotation is available.

- Please Turn Overleaf -

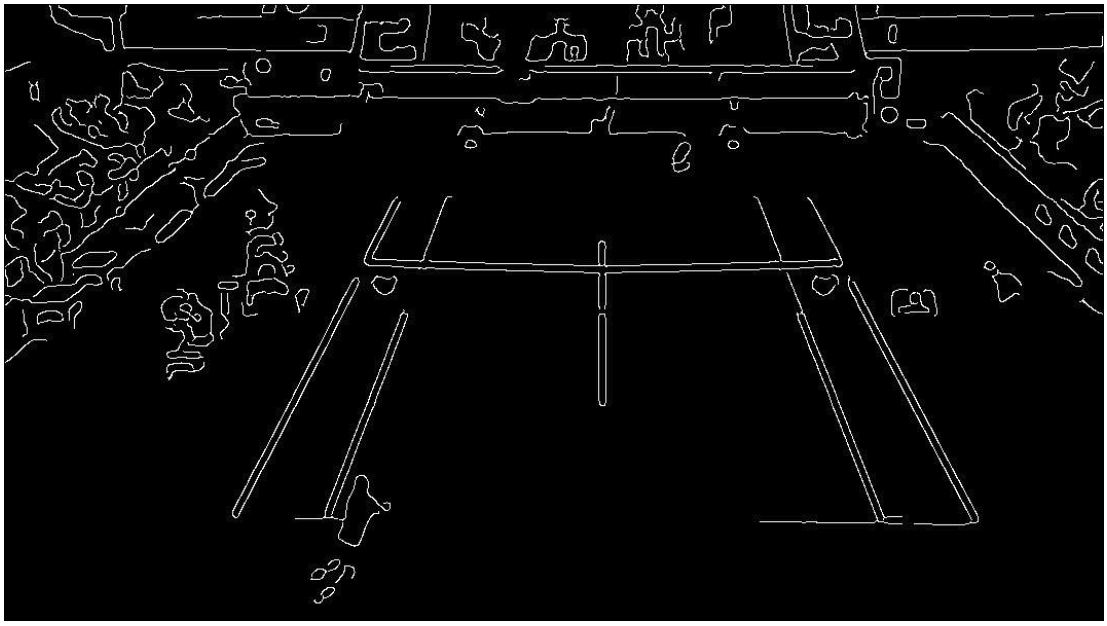
Question 2.4

Canny Edge Detection - best results
Gaussian Filter with sigma = 0.55
Canny with sigma = 1

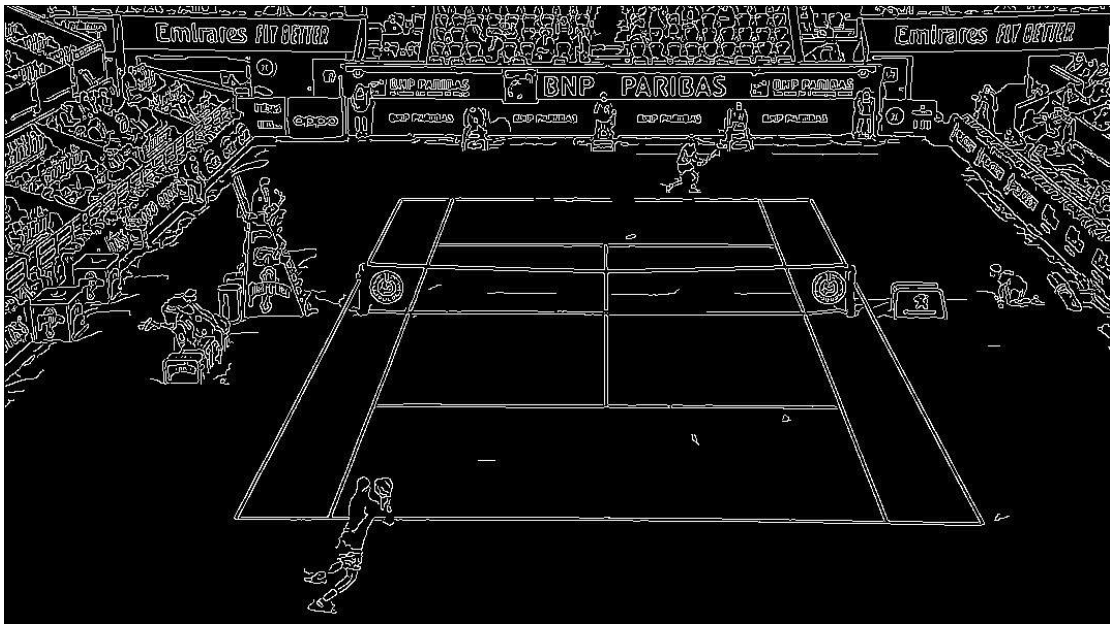


Observations: These parameters provide a good approximation of the canny edge detection by using the gaussian filter to reduce noise and that only important features are learnt. The sigma value is low enough to have learnt the main features and ignore the noise. The canny edge detection is applied with a sigma value of 1. Any higher value will result in loss of detail as shown below.

Canny Edge Detection - other results - not enough detail
Gaussian Filter with sigma = 1
Canny with sigma = 3

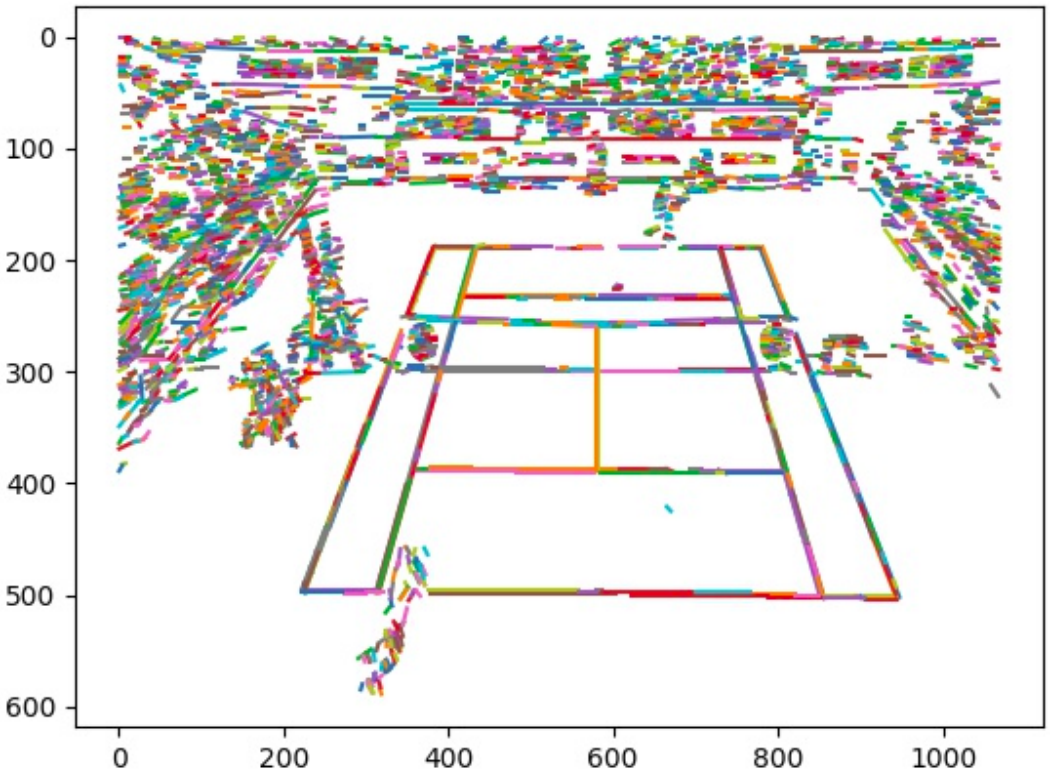


Canny Edge Detection - other results - too much noise
No Gaussian Filter Applied
Canny with sigma = 1



If no gaussian filter is applied, the noise is not filtered and shows unnecessary detail shown above.

Hough Transform - Probabilistic Hough Transform
The Hough Transform is applied after applying the Canny Edge Detection
Threshold = 10
Line Length = 5
Line Gap = 3



Usefulness: Edge detection algorithms can be used for line/edge detection and especially in the context of sports, can be used for tracking the game, body-pose estimation for game-play strategy. It can be effective tool to simulate games, or to make calls as an automated referee - was the ball inside the line or out. It can be useful to visualize and form a starting point to several comp vision algorithms.