MULTIMEDIA UNIVERSITY

# Visual Information Processing

# Assignment 2

Group Name : K&K

Member:

| | |
|---|---|
| Goh Kun Shun | 1151101980 |
| Koh Hong Ching | 1151101808 |

## Abstract

We've designed an image retrieval system to retrieve images of leaves given a query in the form of an image. The system will compare the features of the queried image with the images in the database, and retrieve the top images with the highest cosine similarity. We applied SIFT algorithm for feature extraction, and tried different feature representation, namely bag-of-words and TF-IDF. The result shows that our features extraction methods for both representation are able to beat the baseline method. The database consist of 10 classes of plants.

# Introduction

In this assignment, we had built an image retrieval system that retrieves a set number of image of leaves, given an image as query. This image retrieval system is built on the MalayaKew (MK) Leaf dataset, which consist of the images of the leaves of 10 species classes of plant. For the retrieval system, we have 50 images of the leaves for each species. So a total of 500 images is in the retrieval system.

The retrieval system first extract the visual features of all the leaves in the dataset, using SIFT (scale-invariant feature transform). The features are then clustered using K-means algorithm to produce a set of visual words. The feature representation of the images, including that of the query image, are then compared using cosine similarity and, the top 10 most similar images are retrieved.

We've compared the performance of the image retrieval system using different settings: (1) SIFT features with bag-of-word, (2) SIFT features with TF-IDF weighting (3) RGB histogram, which serves as the baseline method.

In general, we found that using bag-of-word and TF-IDF method outperform the baseline method. Both TF-IDF and bag-of-word had more or less similar performance, although sometimes one of them will slightly outperform the others.

# Description of Methods Used

## Feature Extraction

We first extract the features of each image using SIFT. We set the contrast threshold to 0, to capture as many key points as possible. Each image will have different number of keypoints and thus its descriptor. We save only the descriptors from all images, which represent the local features of the key points of each image.

## Generating Visual Vocabulary

To generate the visual vocabulary, we perform clustering on the local features of all images using K-mean. After some experiment, we settle on K=50. This means we will have 50 visual vocabulary or word, these words will be used as the global feature for our calculating the similarity of the images.

We save the K-mean model, so we can map the features extracted from SIFT to one of the 50 visual words. The model works for images in the database, as well as query image, which are not in the database. More details on Image Retrieval section.

## Feature Representation

We set out to test different way of representing the features: (1) Bag-of-words, (2) TF-IDF (term frequency–inverse document frequency).

Bag-of-words is the represented as the count of the occurences of each visual word that are found in the image.

TF-IDF is used to compensate for the uneven occurrence of visual words across all images. It apply weighting to the visual words based on its term frequency (TF) and its inverse document frequency (IDF). Visual words that are rare across all images, and/or frequent in a particular image are given a higher weightage.

## Image Retrieval

During retrieval, we first extract the features from the querying image, and we map the features to the visual words, using the K-mean model we saved earlier. We will the obtain the features in either bag-of-words or TF-IDF.

This step is repeated for the other 500 images. However, we will use the preloaded features of the 500 images we saved earlier in previous step, to speed up the process.

The querying image are then compared to the 500 images using cosine similarity. The higher the cosine similarity is, the more similar it is. The top 50 most similar images are then retrieved, but only the top 10 are displayed, ranking from the most similar one to the least.

## Results & Analysis

The result is evaluated by these metrics: Mean Average Precision (MAP), Average Precision@K (AP@K) and Recall Rate@K.

Mean Average Precision@K (MAP@K) used to determine the performance of each category for each method. In our case, we are retrieving 50 images, so K is 50. This may be helpful to evaluate overall performance of the retrieval system.

$$\text{Mean Average Precision (MAP)} = \frac{1}{Q}\sum_{q=1}^{Q}(\text{AP})_q$$

The Average Precision@K (AP@K) is used to measure the performance of single query image. In our case, we are retrieving 50 images, so K is 50.

$$\text{Average Precision }(\text{AP})_q = \frac{1}{m}\sum_{k=1}^{K}\frac{r_k}{k}$$

Recall Rate@K used to measure the accuracy of retrieving images of the correct category, given the total number of images retrieved. In our case, we are retrieving 50 images, so K is 50.

$$\text{Recall rate} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{r_k}{R}$$

Average Precision can be found by measuring the area under curve (AUC) of Precision-Recall (PR) curve.

## Query Image: C2

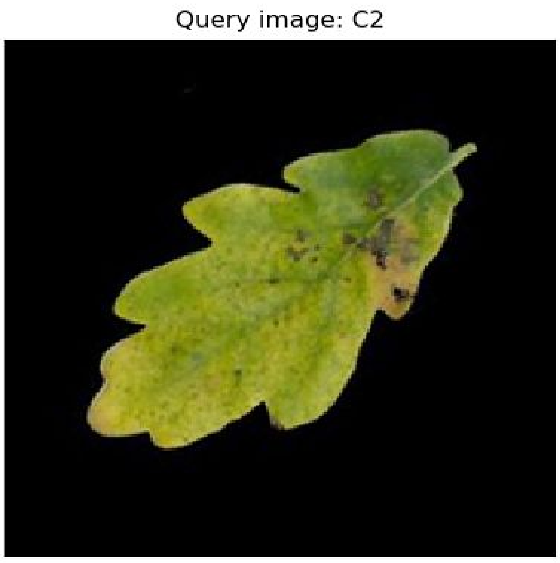


Figure 1 : Query Image

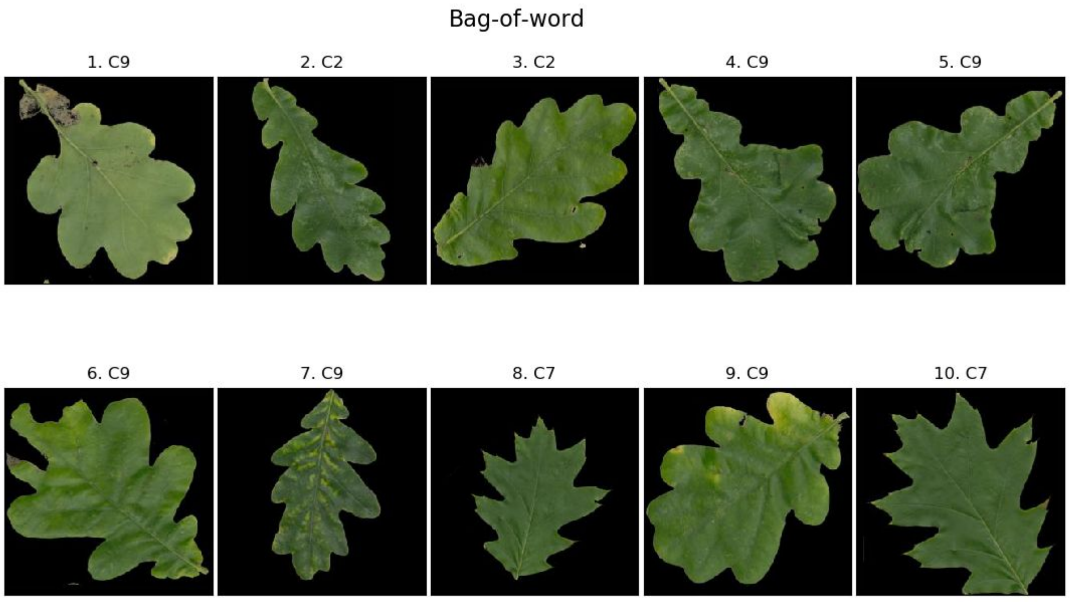


Figure 2 : Image Retrieval from database by Bag-of-Word

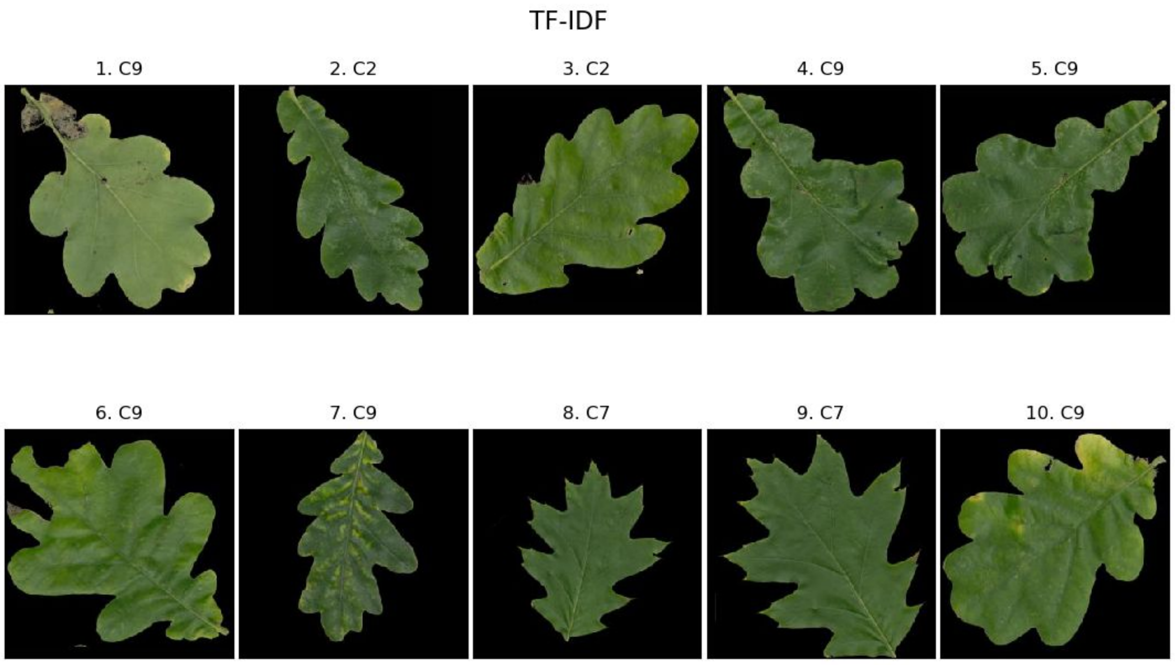


Figure 3 : Image Retrieval from database by Bag-of-Word

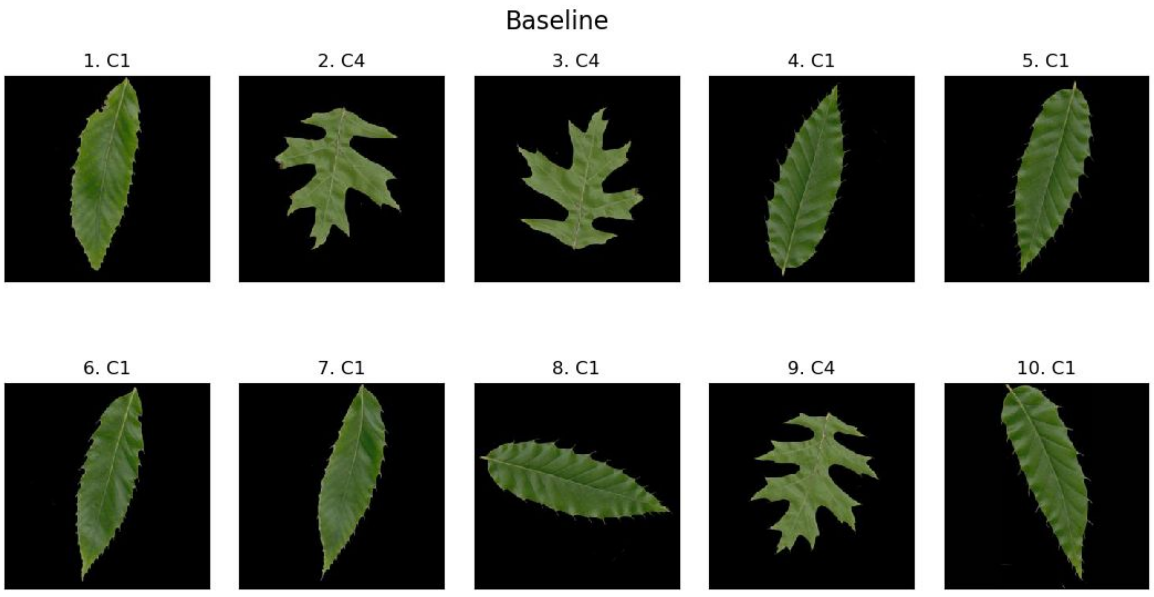
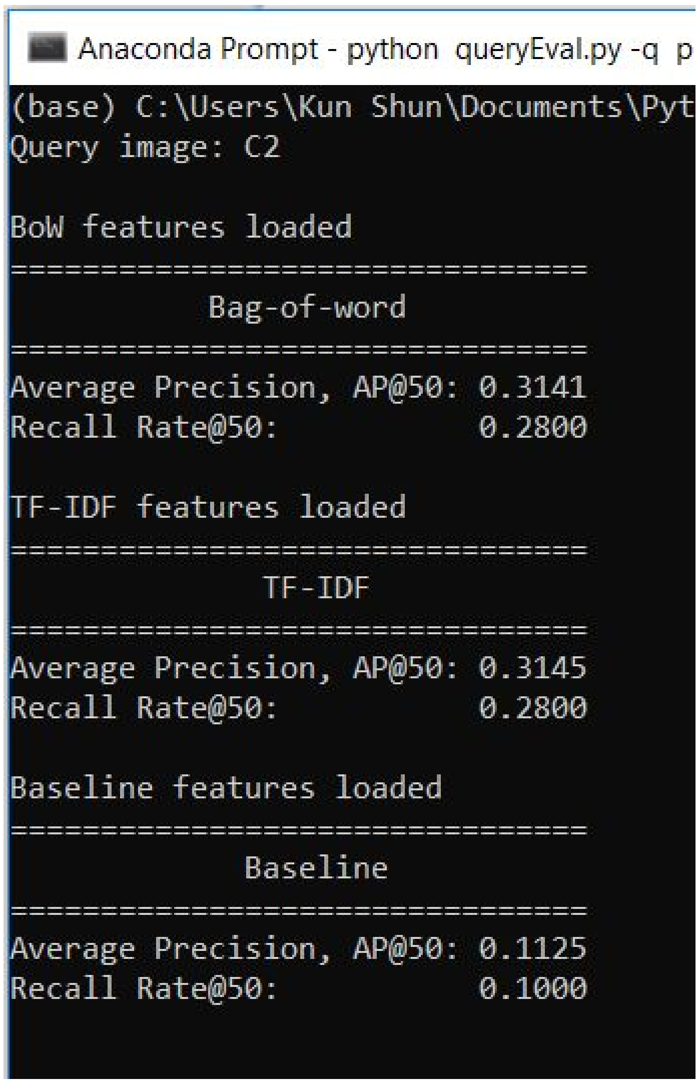


Figure 4 : Image Retrieval from database by Bag-of-Word



Anaconda Prompt - python queryEval.py -q p

```
(base) C:\Users\Kun Shun\Documents\Pyt
Query image: C2

BoW features loaded
================================
          Bag-of-word
================================
Average Precision, AP@50: 0.3141
Recall Rate@50:           0.2800

TF-IDF features loaded
================================
             TF-IDF
================================
Average Precision, AP@50: 0.3145
Recall Rate@50:           0.2800

Baseline features loaded
================================
            Baseline
================================
Average Precision, AP@50: 0.1125
Recall Rate@50:           0.1000
```

Figure 5 : Result of Query C2 image

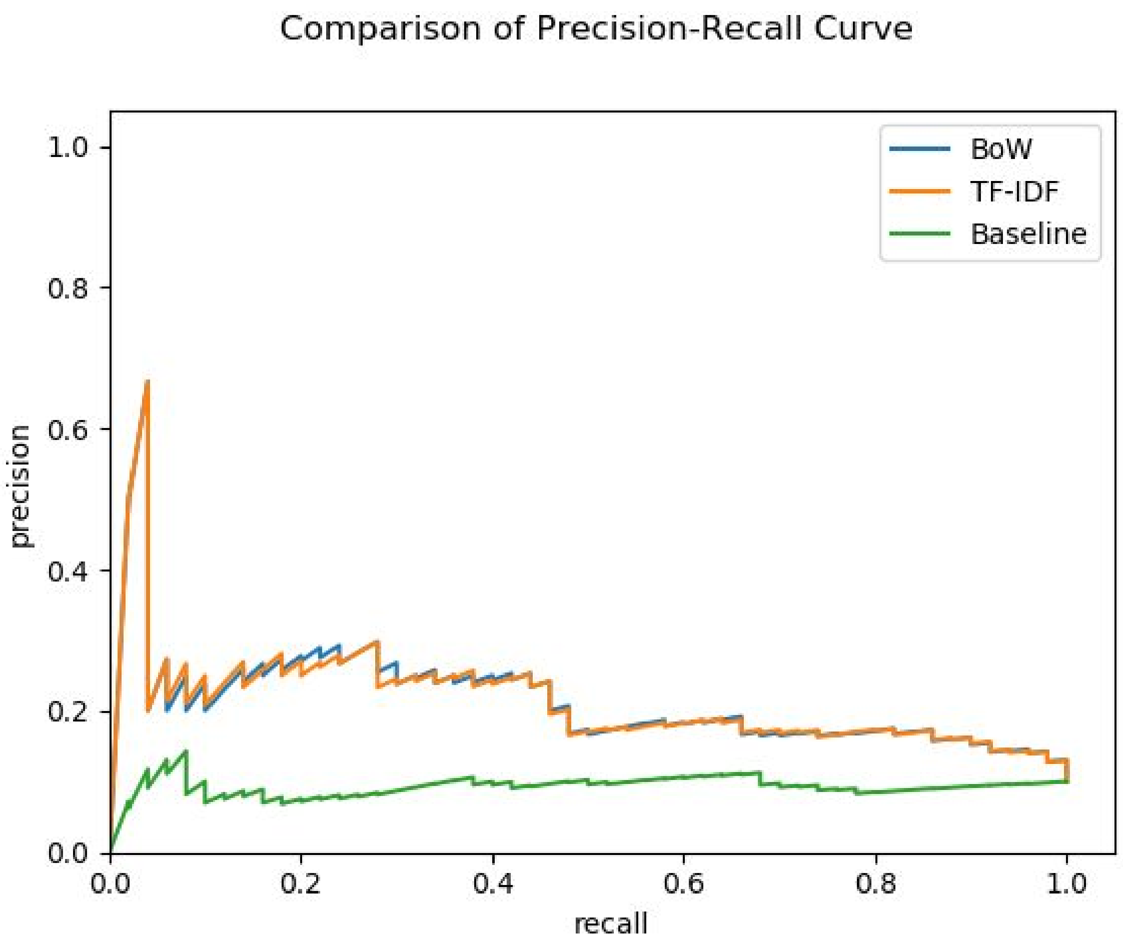


Figure 6 : Precision-Recall (PR) curve for C2 query

For the C2 query image, the retrieval system achieves the highest AP@50 of 0.3145 with TF-IDF, followed by BoW with slightly lower AP@50 of 0.3141 and followed by the lowest AP@50 of 0.1125 with Baseline. The recall rate@10 of both BoW and TF-IDF is 0.2800 which is slightly higher than the 0.1100 of Baseline's recall@rate. The Bow and TF-IDF has approximately the same area under curve which is higher than the baseline. In figure 6, the graph has shown that the average percision of BoW and TF-IDF is higher than baseline.

## Query Image: C4

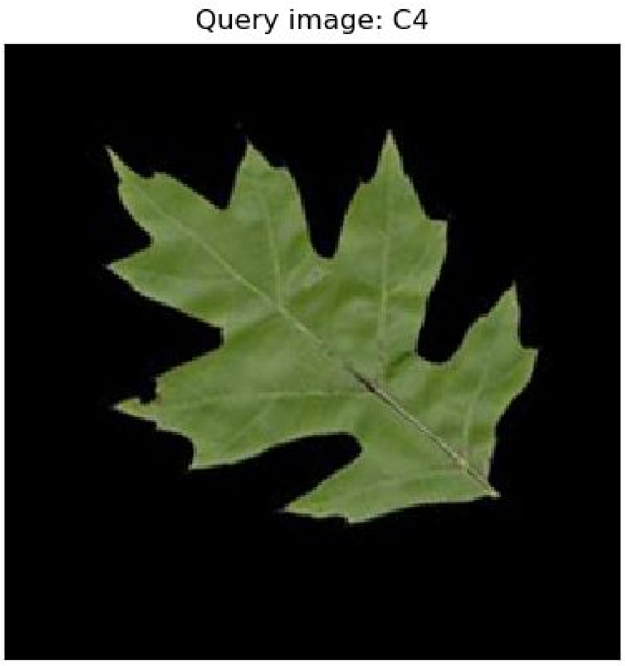


Figure 7 : Query Image

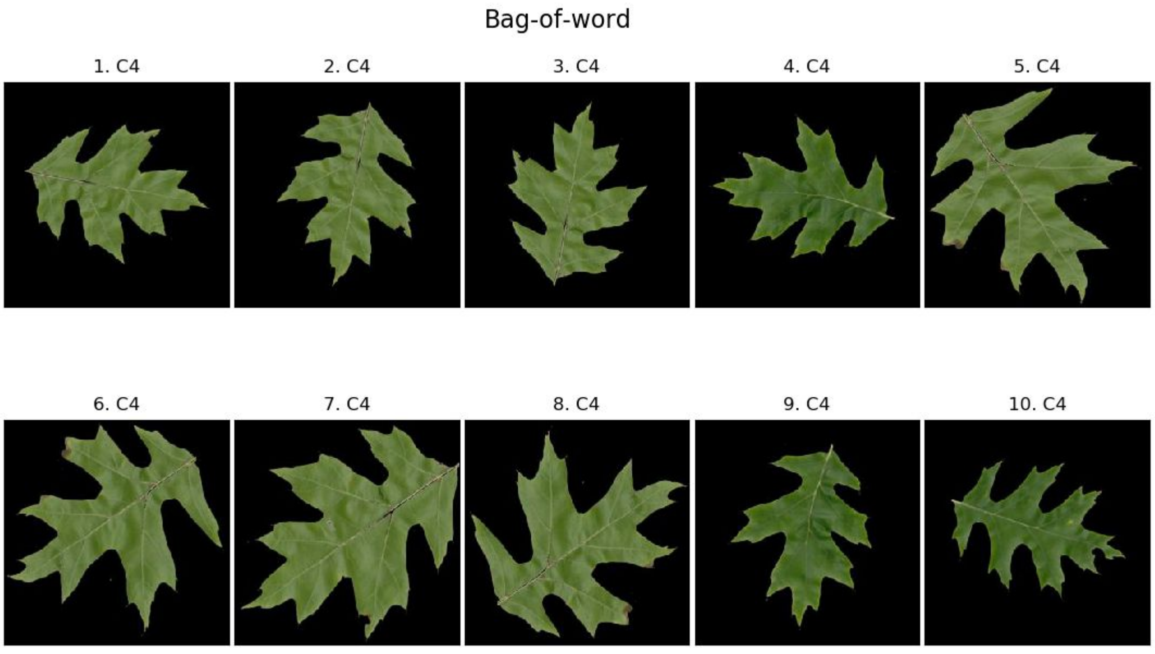


Figure 8 : Image Retrieval from database by BoW

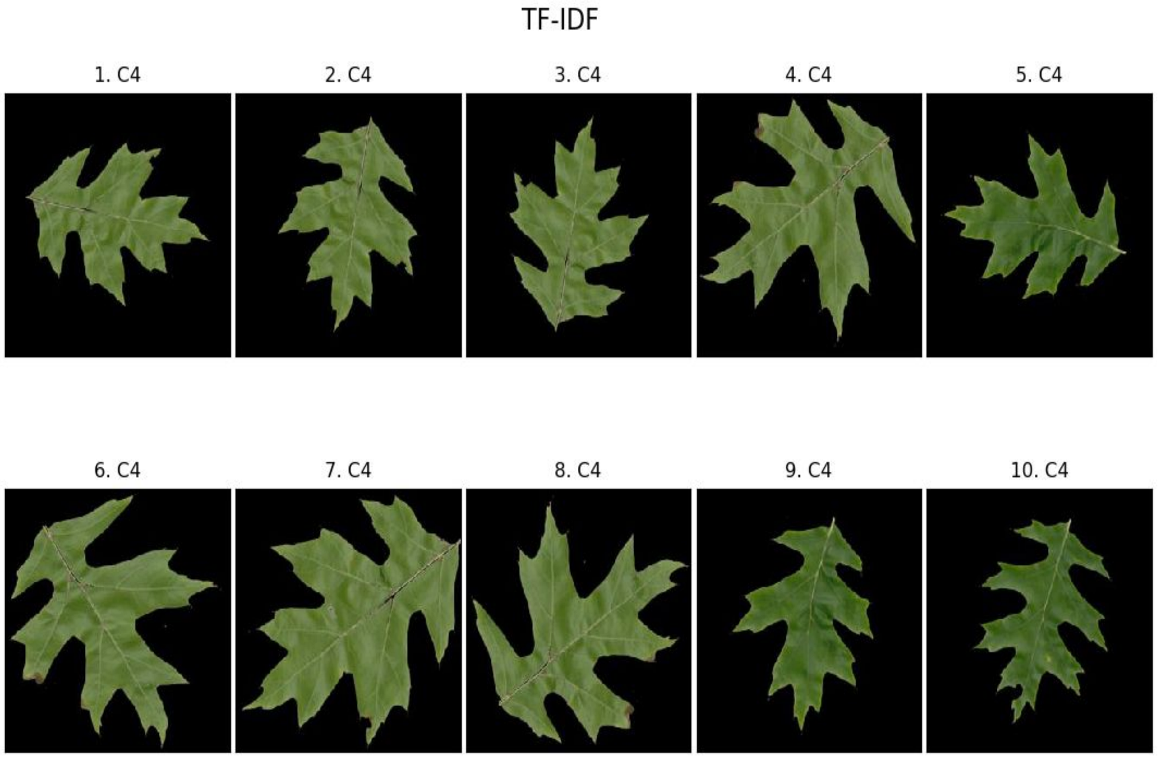


Figure 9 : Image Retrieval from database by TF-IDF

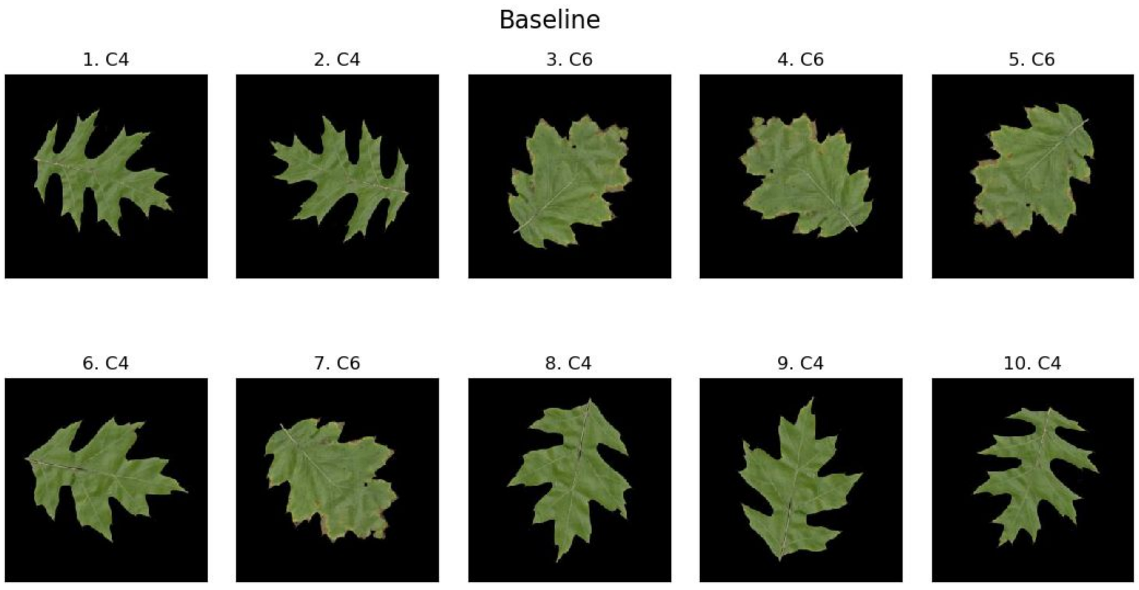
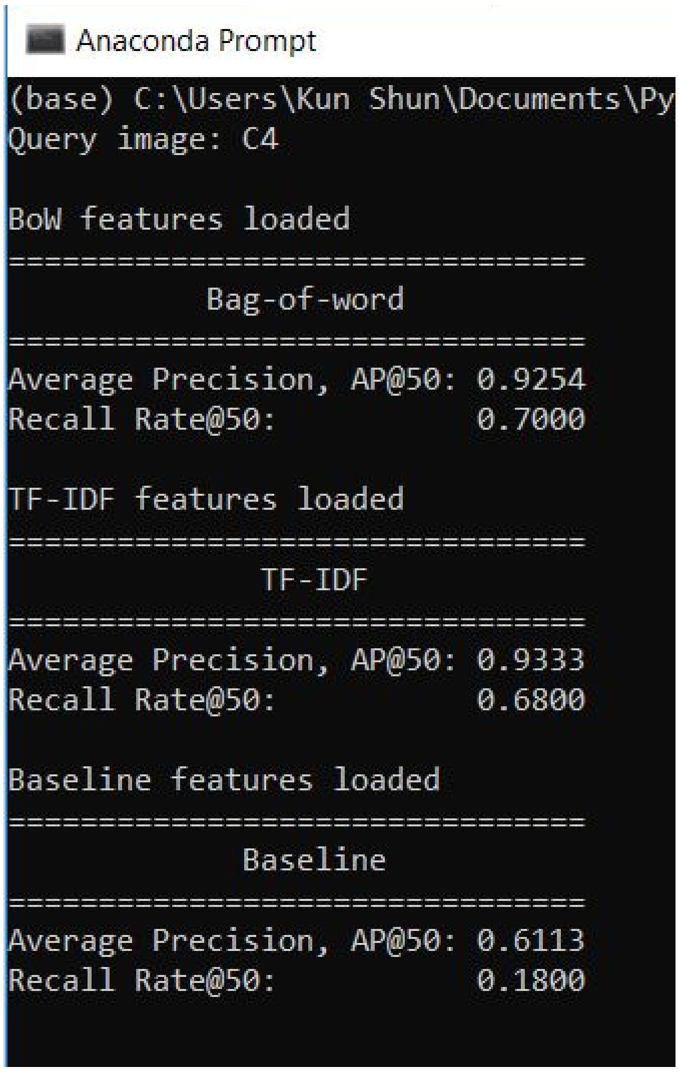


Figure 10 : Image Retrieval from database by Baseline



```
(base) C:\Users\Kun Shun\Documents\Py
Query image: C4

BoW features loaded
==================================
           Bag-of-word
==================================
Average Precision, AP@50: 0.9254
Recall Rate@50:           0.7000

TF-IDF features loaded
==================================
              TF-IDF
==================================
Average Precision, AP@50: 0.9333
Recall Rate@50:           0.6800

Baseline features loaded
==================================
             Baseline
==================================
Average Precision, AP@50: 0.6113
Recall Rate@50:           0.1800
```

Figure 11 : Result of Query Image C4

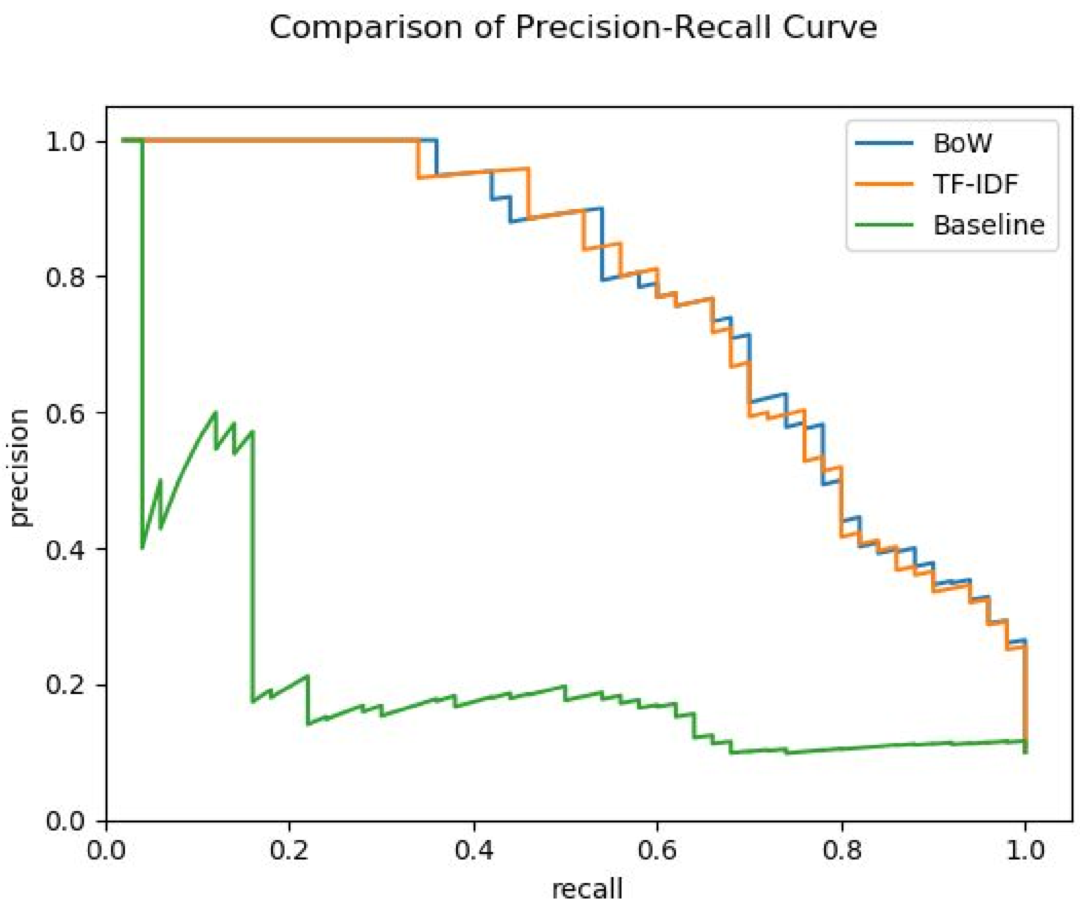


Figure 12 : Precision-Recall (PR) curve for C4 query image

For the C4 query, the BoW and TF-IDF both had achieved high average precision@50 which is 0.9254 and 0.9333 respectively. The recall rate@50 of BoW and TF-IDF which is 0.7000 and 0.6800 also higher than the recall rate@50 of baseline which is 0.1800. The area under curve of BoW and TF-IDF is higher than baseline method. All these metrics shown that the BoW and TF-IDF for this query beats the baseline method for this query.

## Query Image: C5

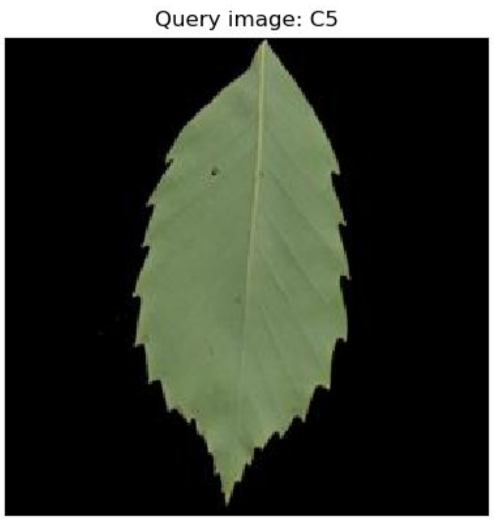


Figure 13 : Query Image

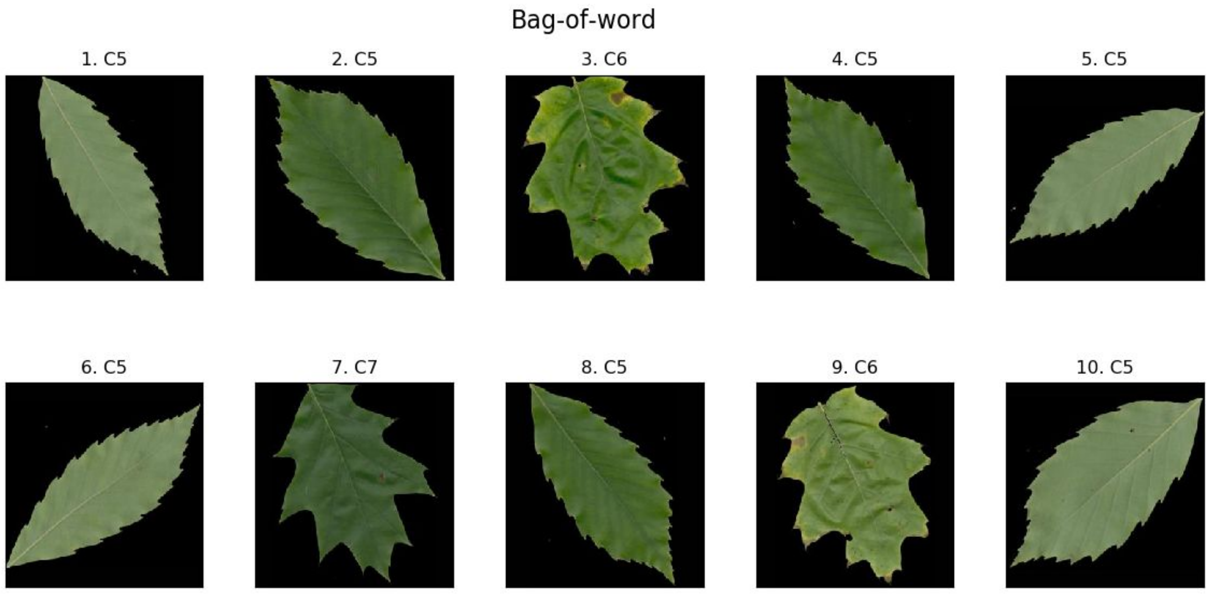


Figure 14 : Image Retrieved from database for BoW

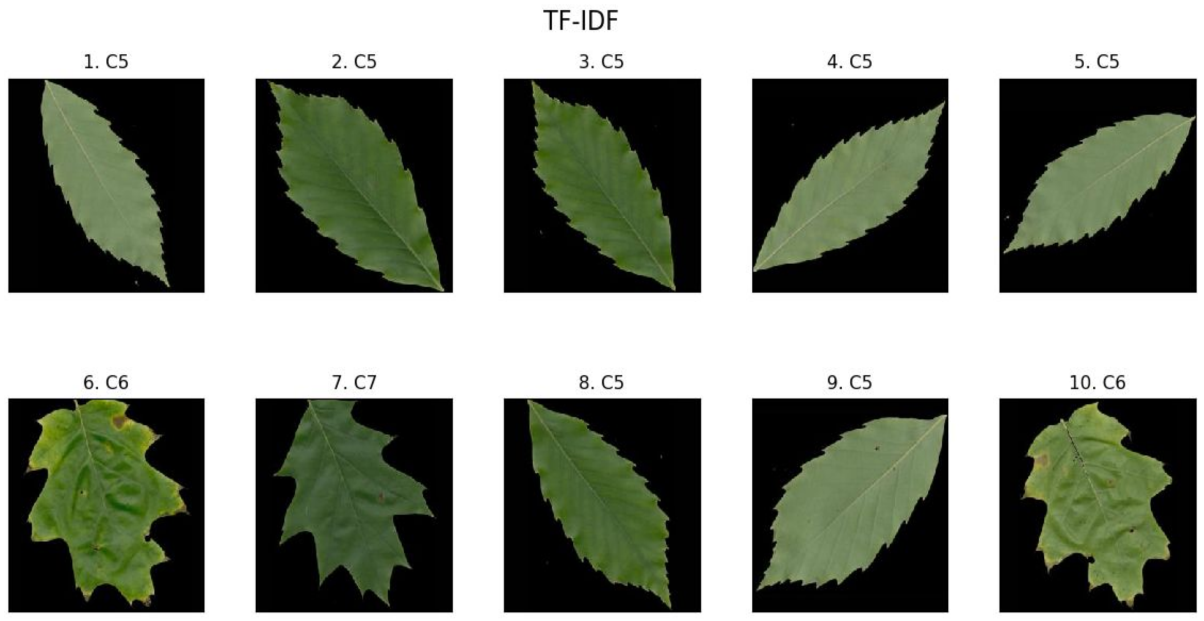


Figure 15 : Image Retrieved from database for TF-IDF

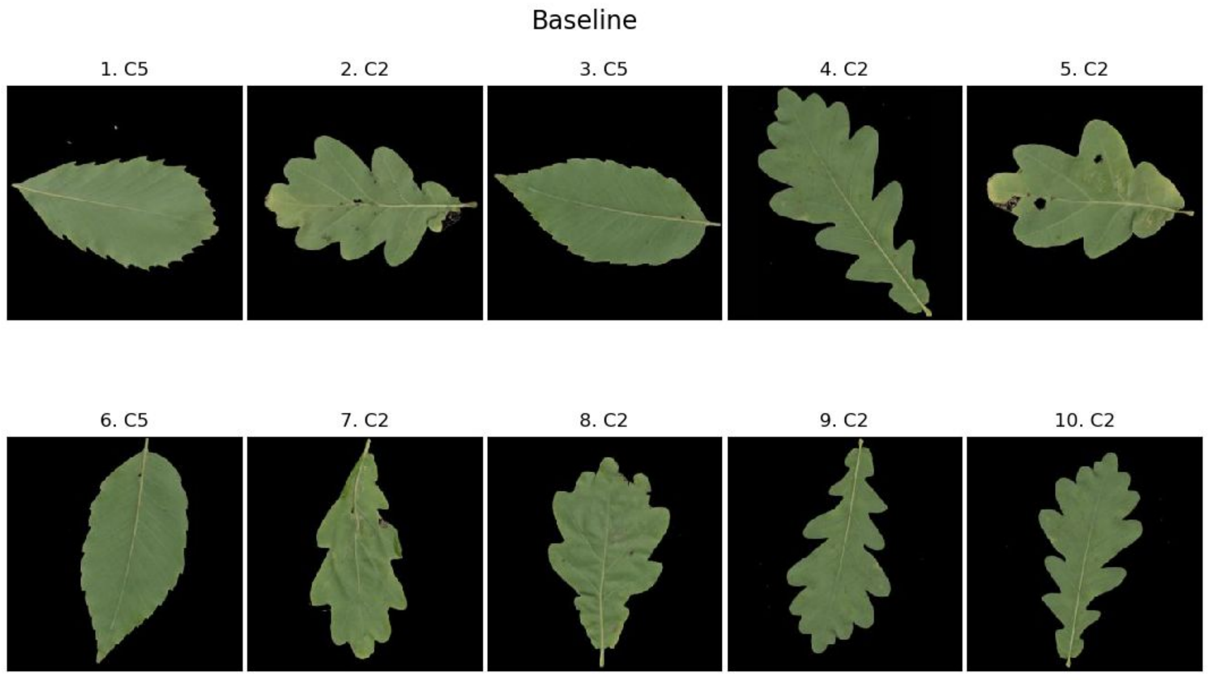
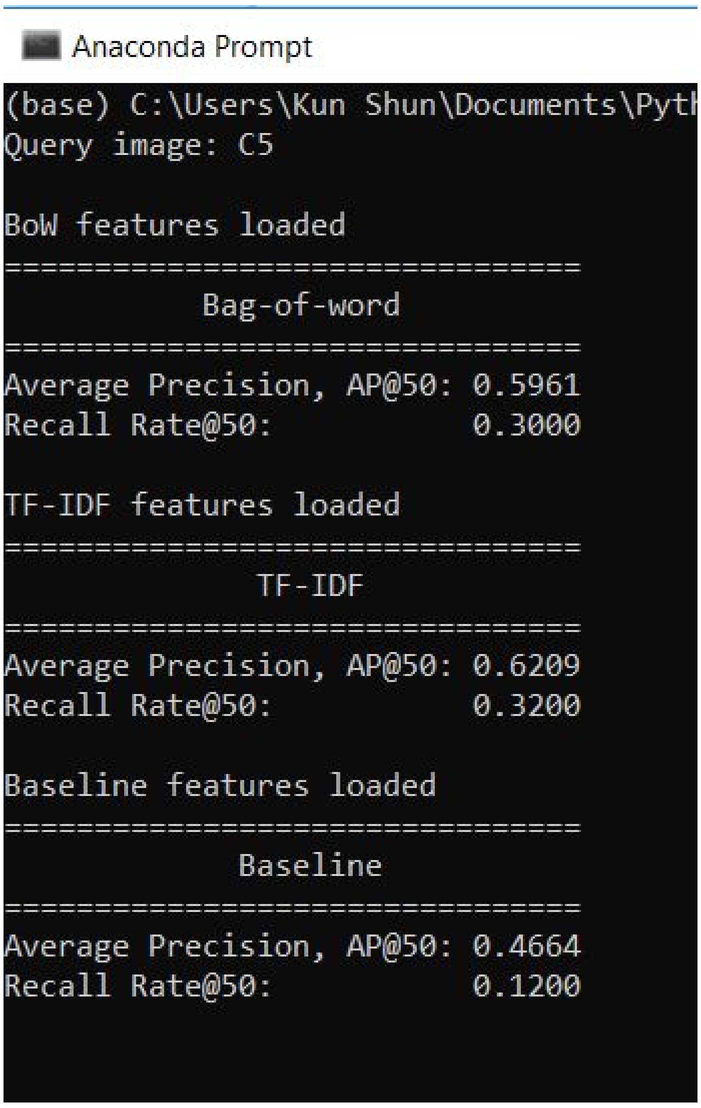


Figure 16 : Image Retrieved from database for Baseline



```
(base) C:\Users\Kun Shun\Documents\Pyth
Query image: C5

BoW features loaded
================================
          Bag-of-word
================================
Average Precision, AP@50: 0.5961
Recall Rate@50:           0.3000

TF-IDF features loaded
================================
             TF-IDF
================================
Average Precision, AP@50: 0.6209
Recall Rate@50:           0.3200

Baseline features loaded
================================
            Baseline
================================
Average Precision, AP@50: 0.4664
Recall Rate@50:           0.1200
```

Figure 17 : Result of C5 query image

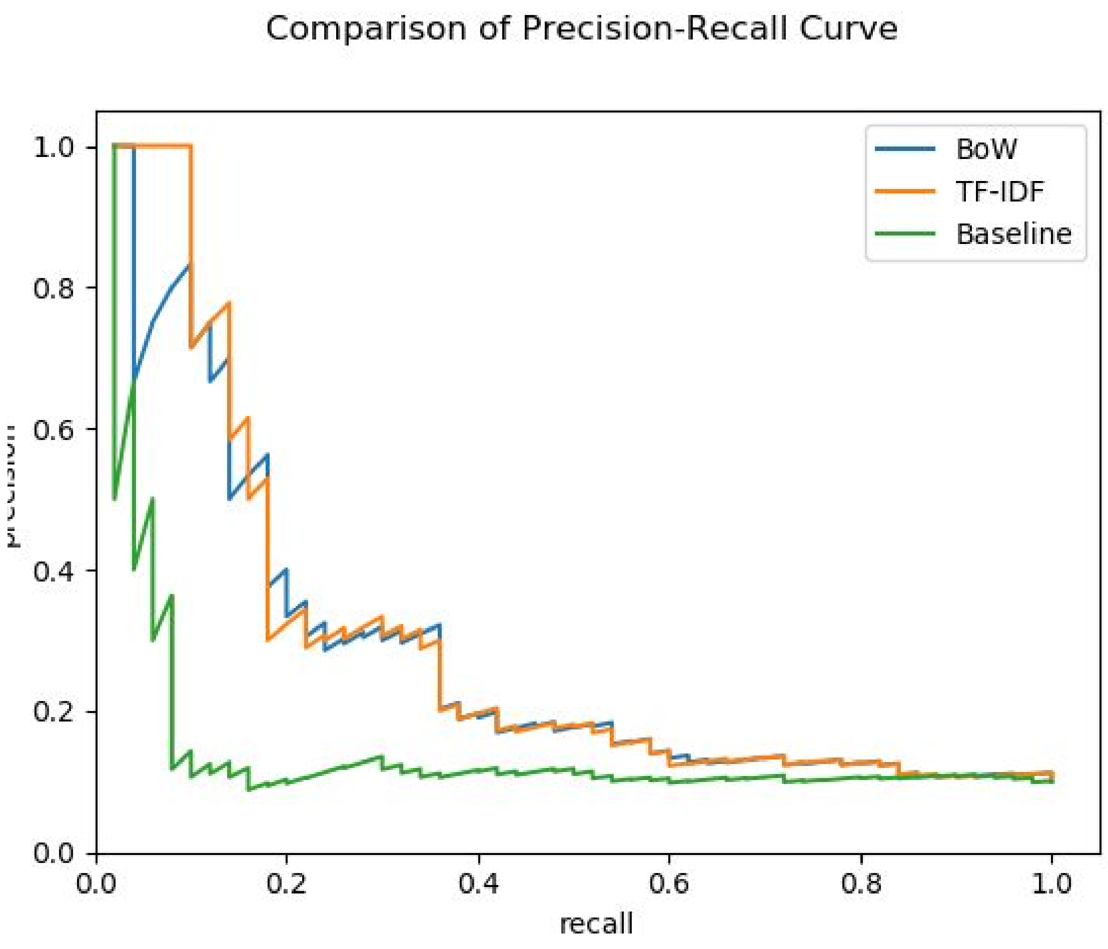


Figure 18 : Precision-Recall (PR) curve for C5 query image

For query using image q_C5, the retrieval system achieves the highest AP@50 of 0.6209 score with TF-IDF, with BoW falls slightly behind with 0.5961. Both are able to achieve a high Recall Rate@10 of approximate to 0.3. The area under curve reflects this as well, with TF-IDF having the largest area under curve, followed by BoW, which means they both have high average precision. All these metrics shows that both BoW and TF-IDF beats the baseline method, for this query.

## Query Image: C6

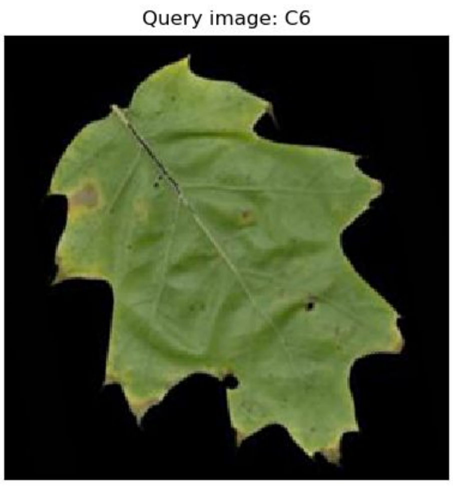


Figure 19 : Query Image

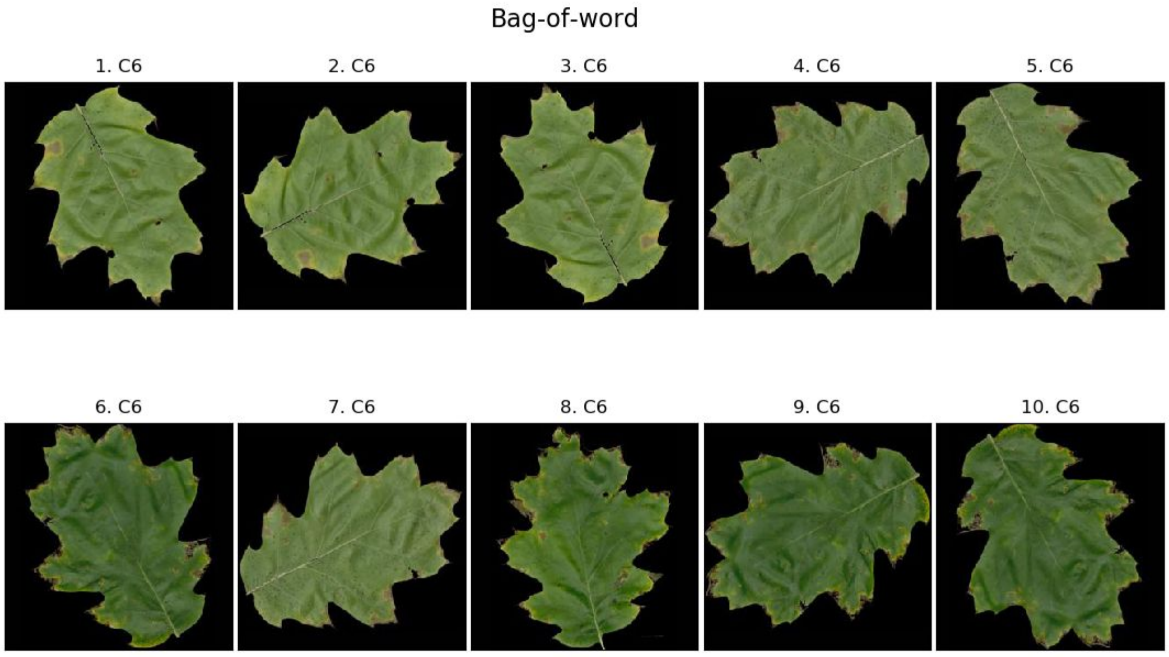


Figure 20 : Image Retrieval from database for BoW

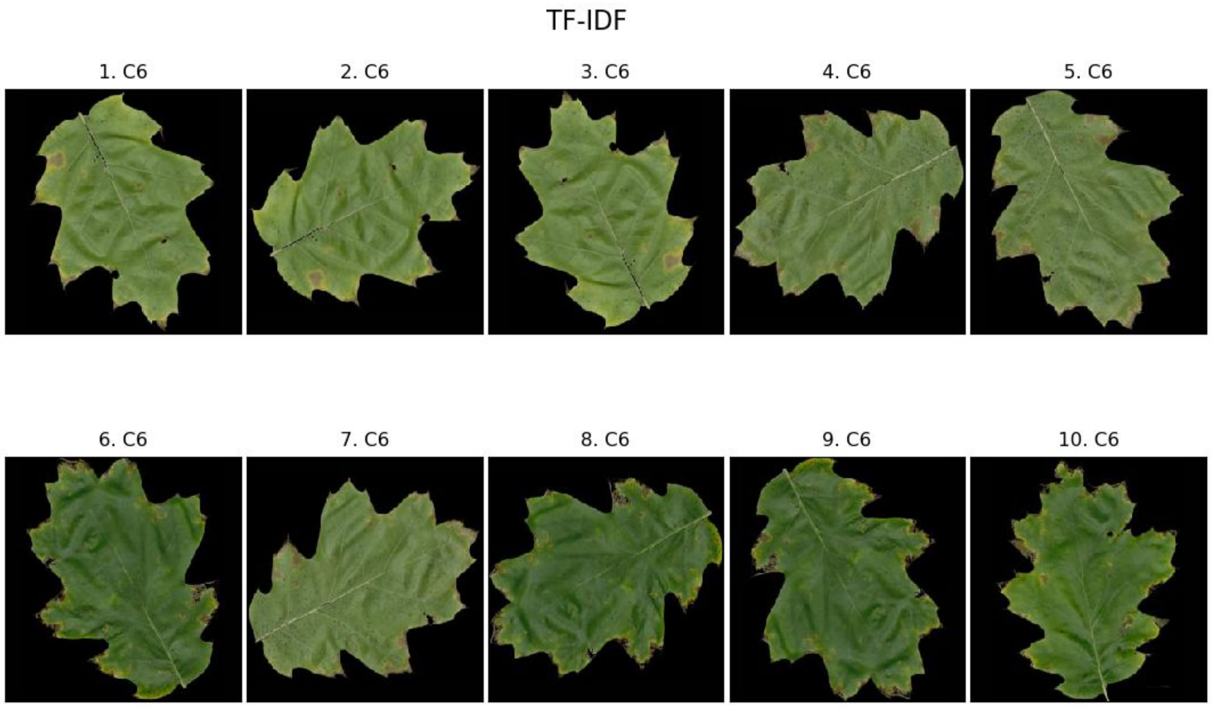


Figure 21 : Image Retrieval from database for TF-IDF

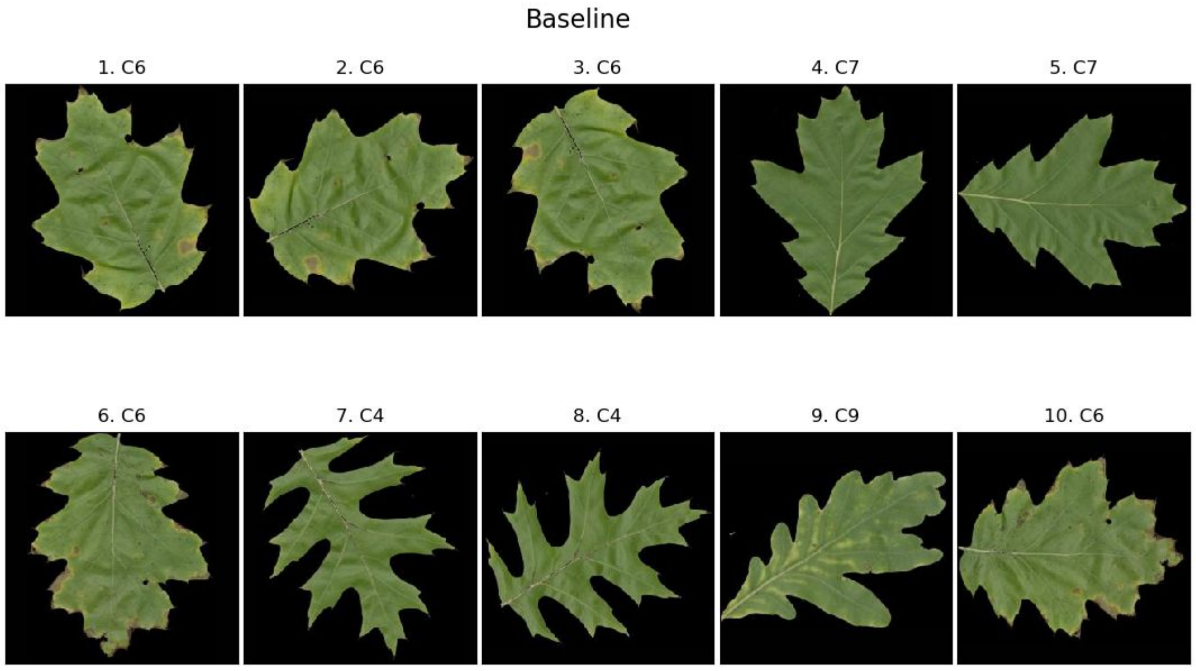
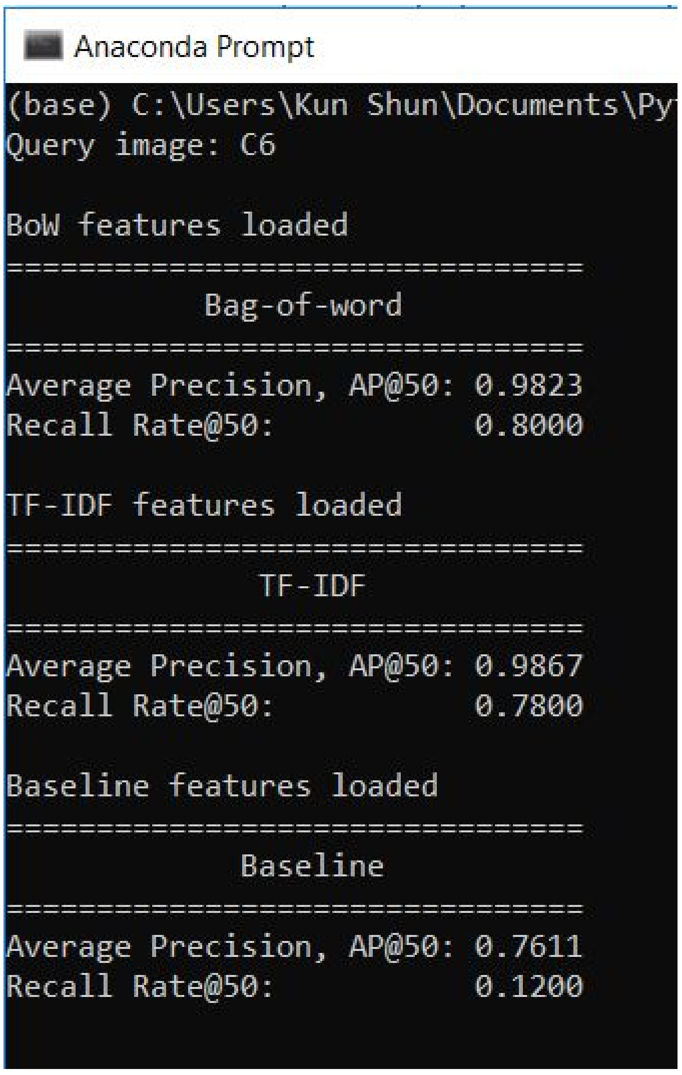


Figure 22 : Image Retrieval from database for Baseline



```
(base) C:\Users\Kun Shun\Documents\Py
Query image: C6

BoW features loaded
================================
          Bag-of-word
================================
Average Precision, AP@50: 0.9823
Recall Rate@50:           0.8000

TF-IDF features loaded
================================
             TF-IDF
================================
Average Precision, AP@50: 0.9867
Recall Rate@50:           0.7800

Baseline features loaded
================================
            Baseline
================================
Average Precision, AP@50: 0.7611
Recall Rate@50:           0.1200
```

Figure 23 : Result of C6 Query Image

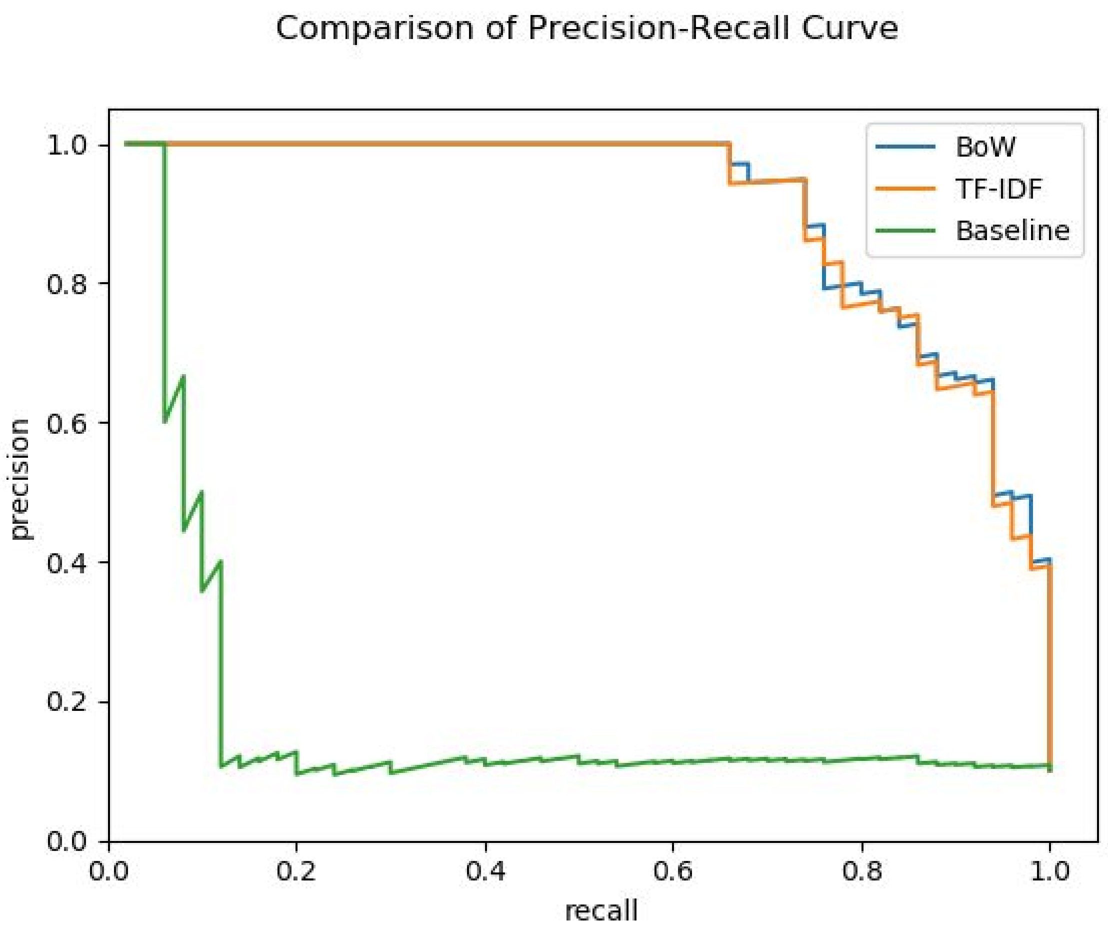


Figure 24 : Precision-Recall (PR) curve for C6 Query Image

For the C6 query, the BoW and TF-IDF both had achieved high average precision@50 and high recall rate@50 which is approximately to 1.0000 and 0.8000 respectively. The Area Under Curve (AUC) of Precision-Recall(PR) curve of BoW and TF-IDF is higher than Baseline. All these metrics shown that the BoW and TF-IDF for this query beats the baseline method for this query.

## Query Image: C10

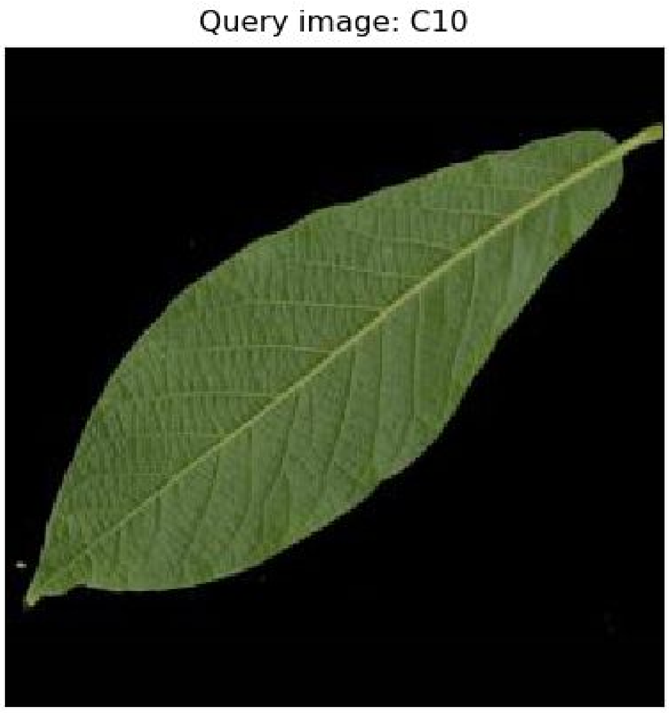


Figure 25 : Query Image

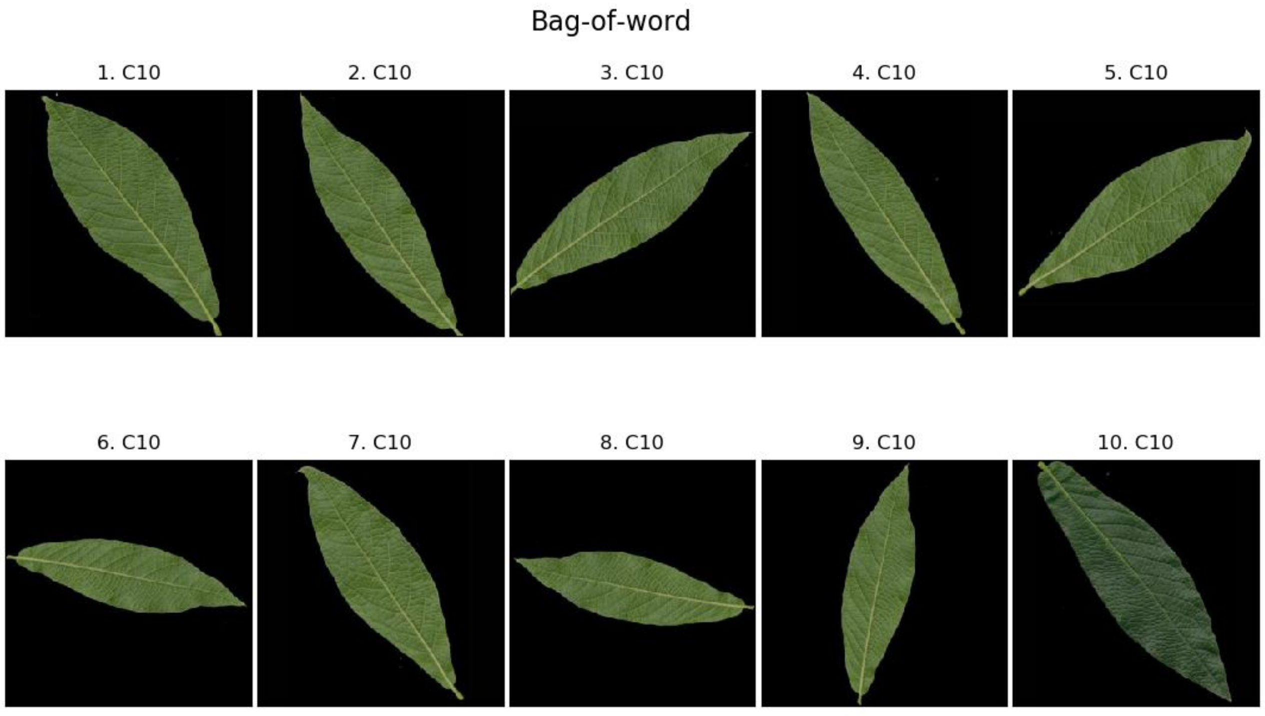


Figure 26 : Image Retrieval from database for BoW

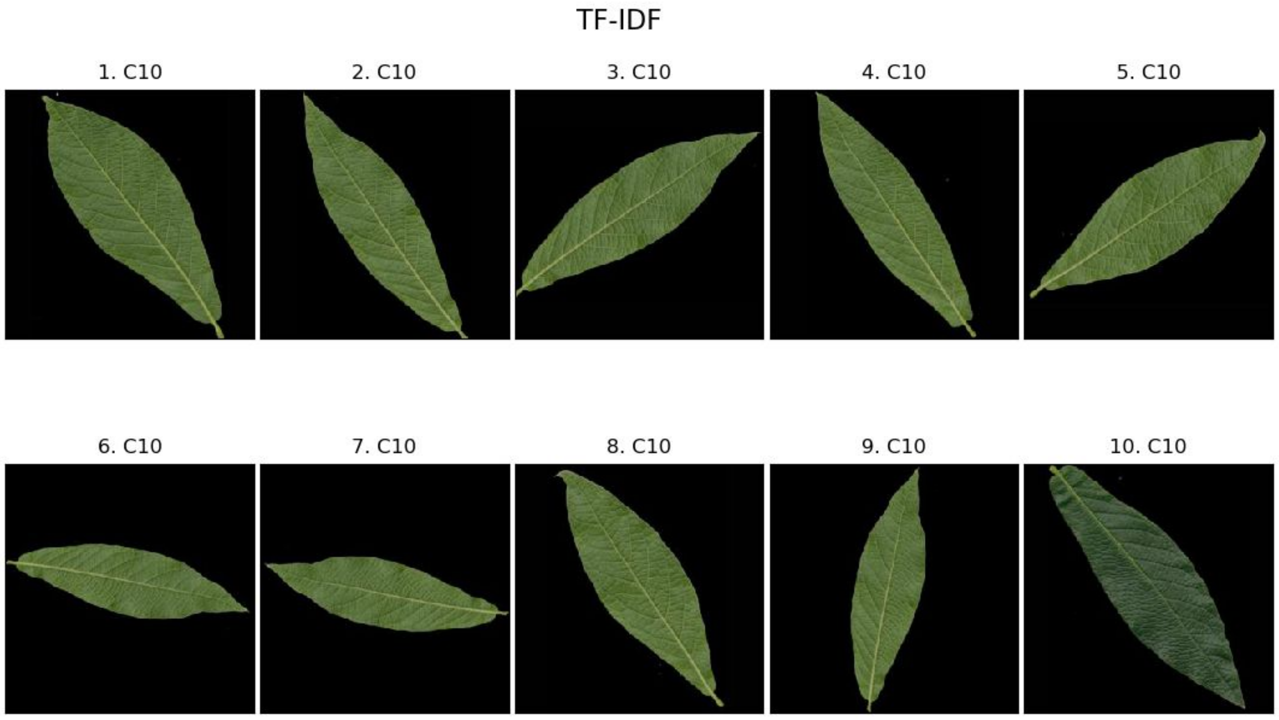


Figure 27 : Image Retrieval from database for TF-IDF

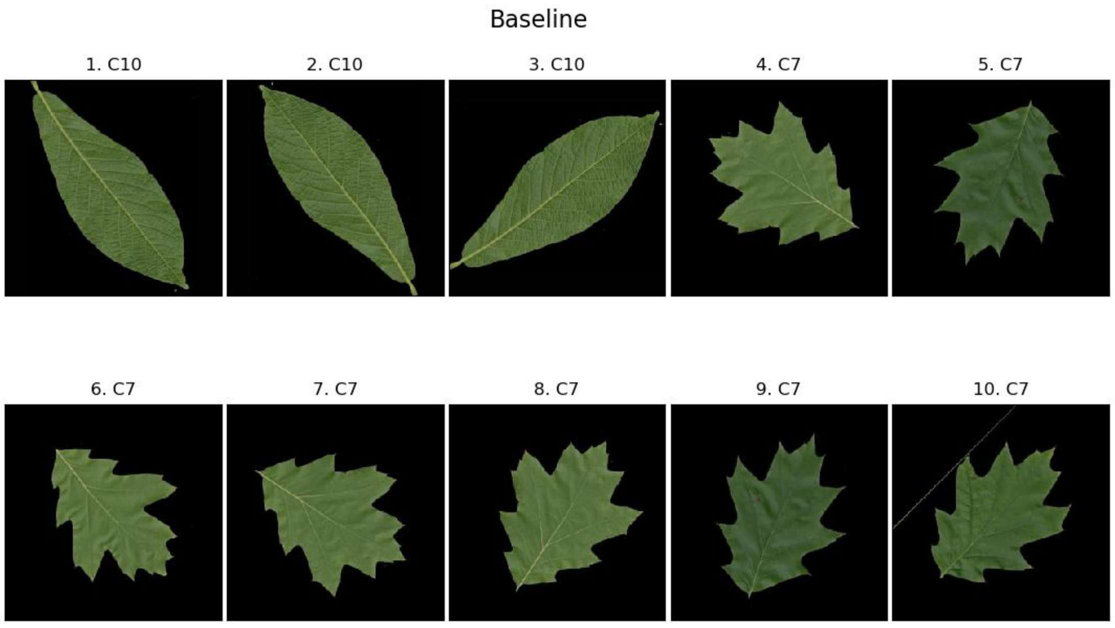
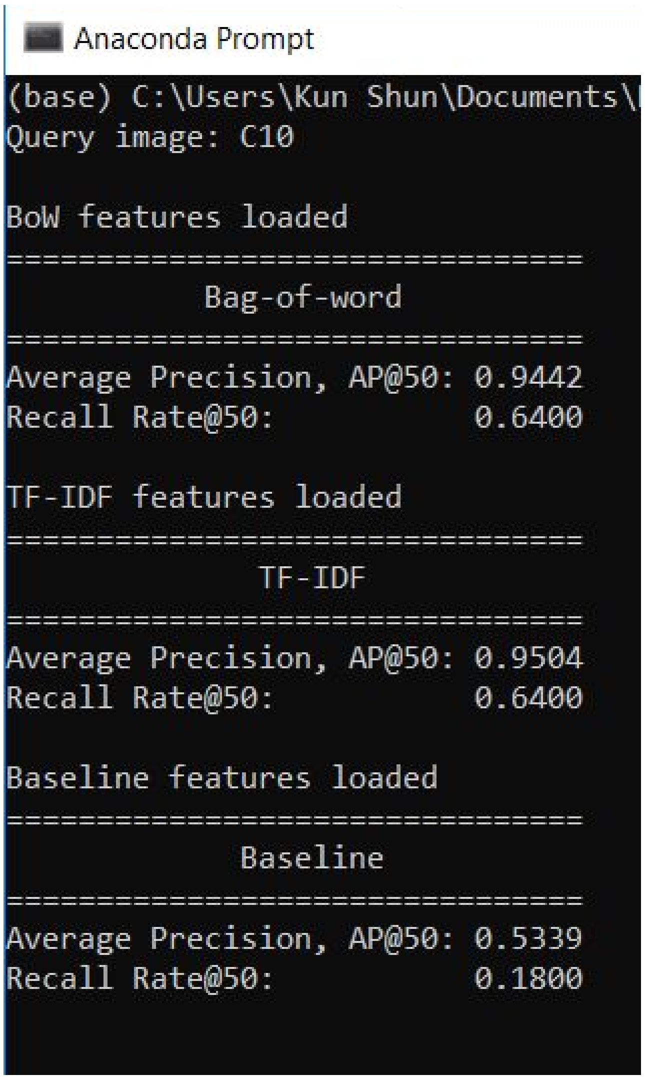


Figure 28 : Image Retrieval from database for Baseline



```
(base) C:\Users\Kun Shun\Documents\
Query image: C10

BoW features loaded
=================================
           Bag-of-word
=================================
Average Precision, AP@50: 0.9442
Recall Rate@50:           0.6400

TF-IDF features loaded
=================================
              TF-IDF
=================================
Average Precision, AP@50: 0.9504
Recall Rate@50:           0.6400

Baseline features loaded
=================================
             Baseline
=================================
Average Precision, AP@50: 0.5339
Recall Rate@50:           0.1800
```

Figure 29 : Result of C10 Query Image

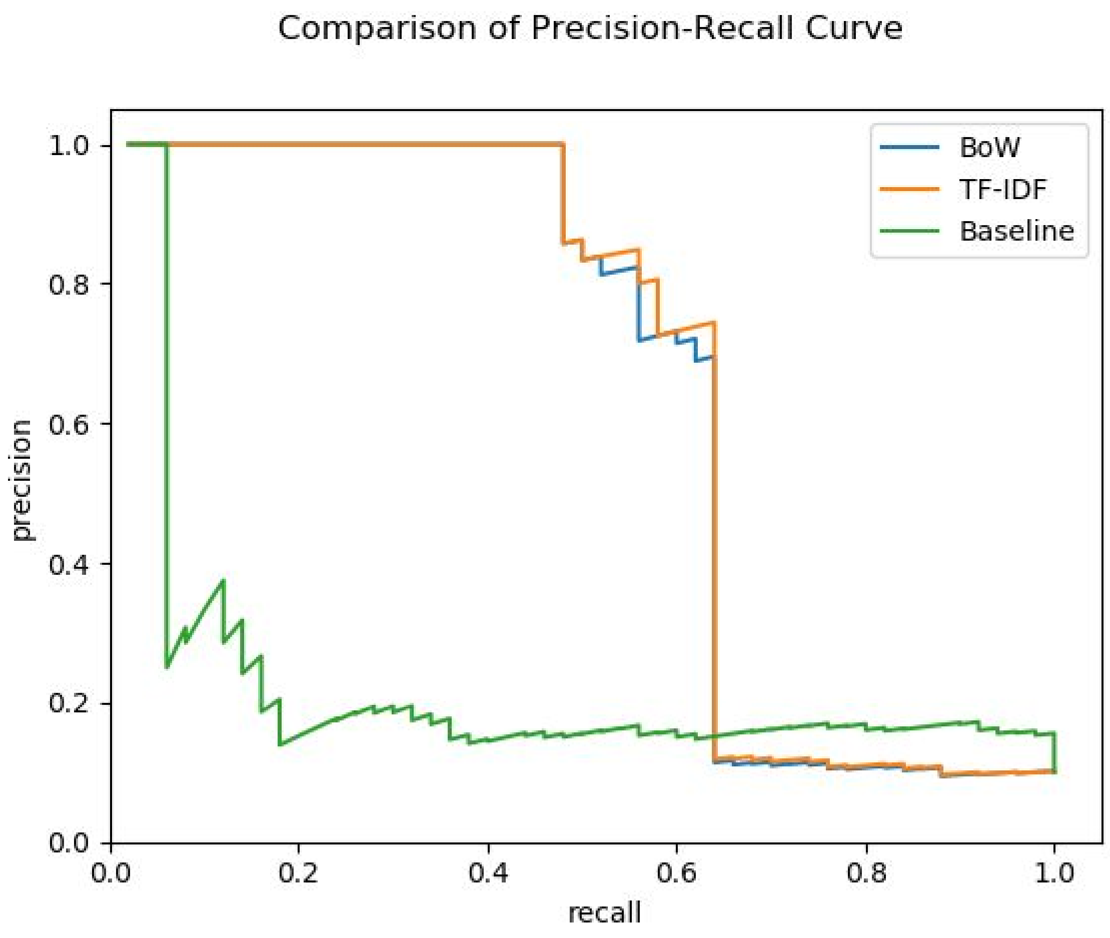


Figure 30 : Precision-Recall (PR) curve for C10 Query Image

For the C10 query, the retrieval system achieves almost perfect AP@50 value which is 0.9 that nearly to 1.000 value for BoW and TF-IDF. The recall rate@50 for BoW and TF-IDF is the same which is 0.6400 and higher than the baseline recall rate@50 which is 0.1800. The area under curve reflects this as well, BoW has slightly higher area under curve than TF-IDF. followed by the baseline. All these metrics shown that the BoW and TF-IDF for this query beats the baseline method for this query.

## Full Evaluation

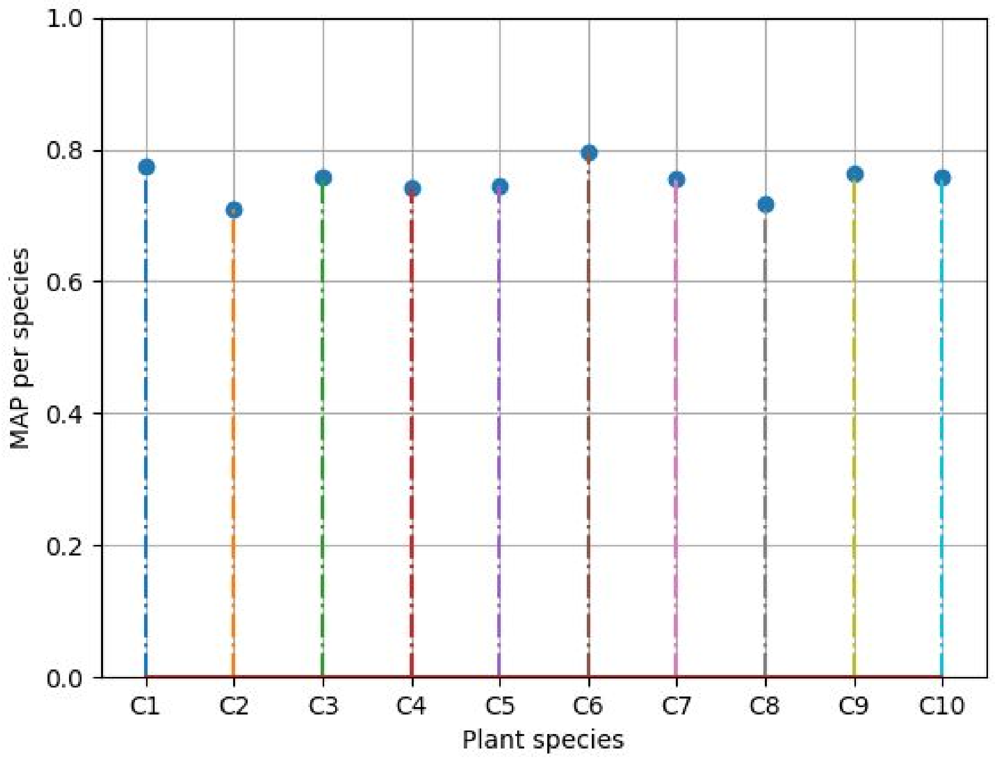


Figure 31 : Mean Average Precision (MAP) of Bag of Words (BoW)

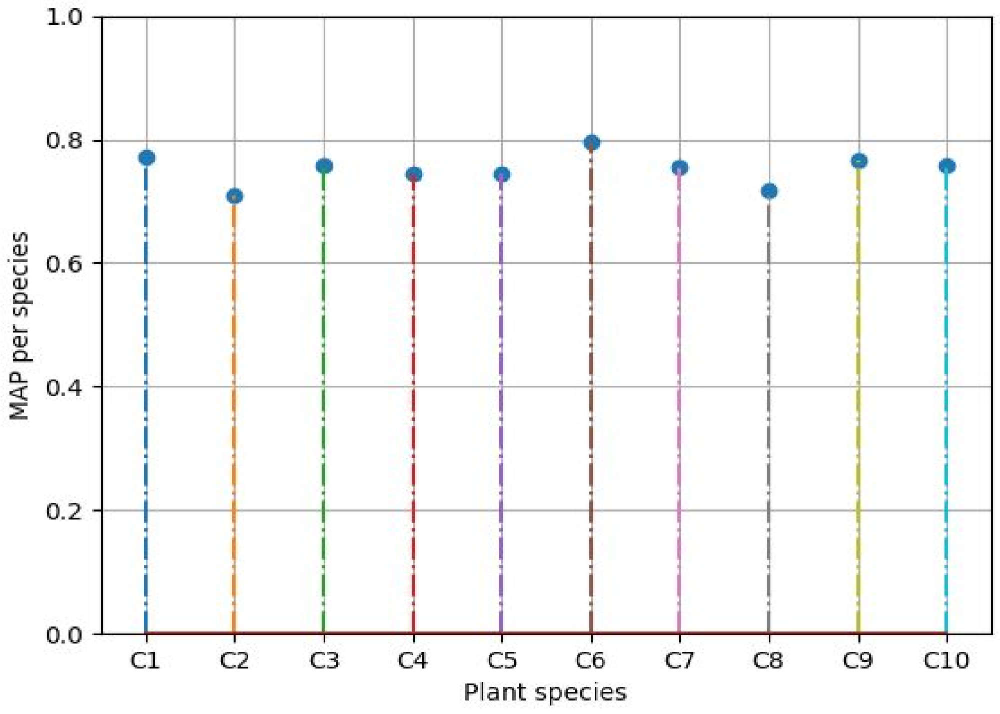


Figure 32 : Mean Average Precision (MAP) of TF-IDF

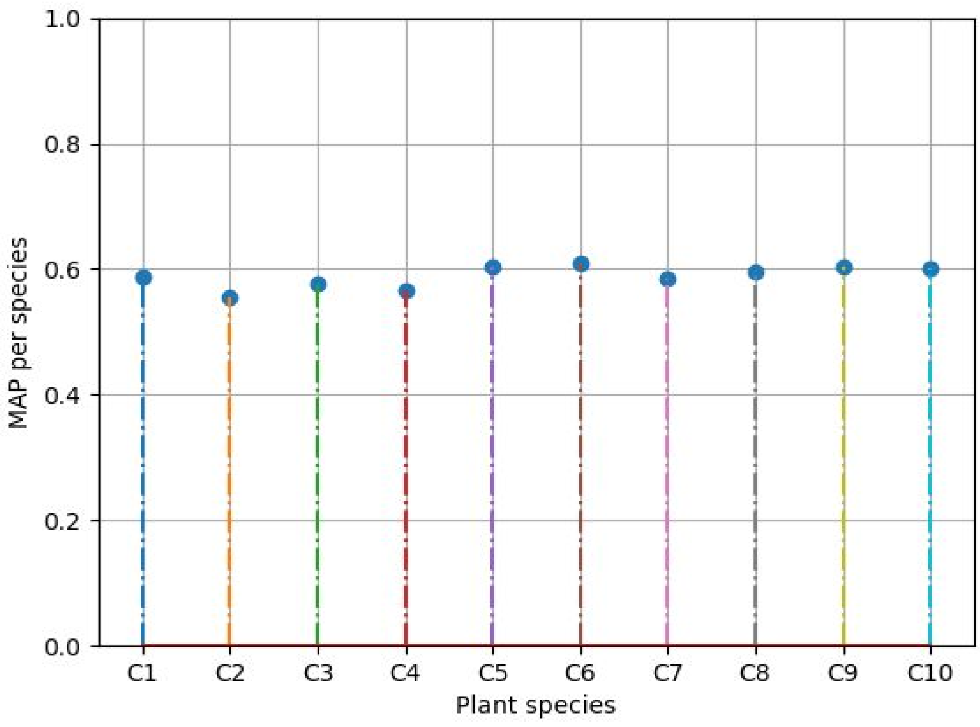
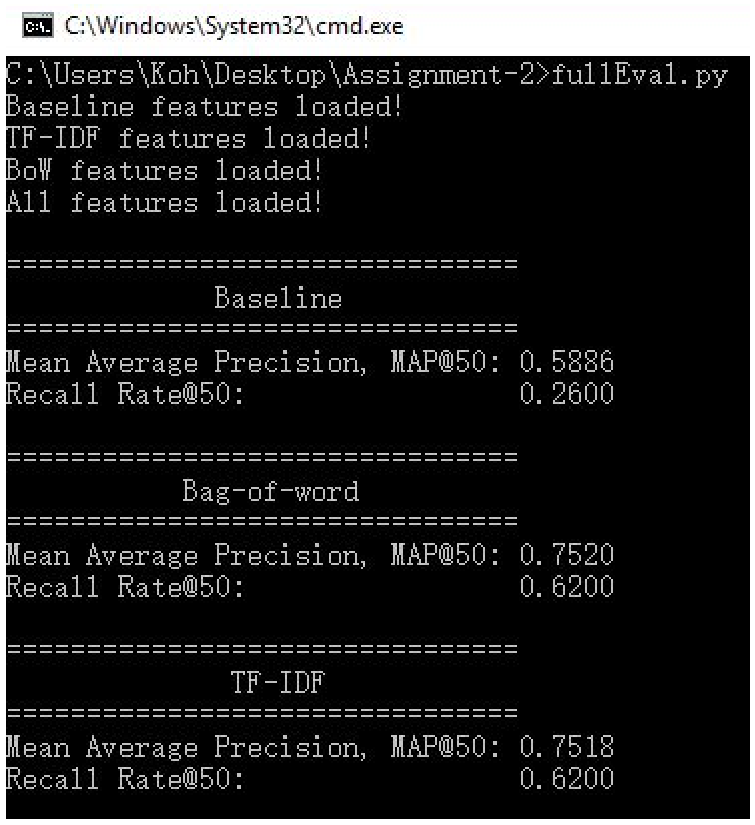


Figure 33 : Mean Average Precision (MAP) of Baseline



```
C:\Users\Koh\Desktop\Assignment-2>fullEval.py
Baseline features loaded!
TF-IDF features loaded!
BoW features loaded!
All features loaded!

=================================
             Baseline
=================================
Mean Average Precision, MAP@50: 0.5886
Recall Rate@50:                 0.2600

=================================
           Bag-of-word
=================================
Mean Average Precision, MAP@50: 0.7520
Recall Rate@50:                 0.6200

=================================
              TF-IDF
=================================
Mean Average Precision, MAP@50: 0.7518
Recall Rate@50:                 0.6200
```

Figure 34 : Result of full Evaluation

Overall, the retrieval system achieves the highest mean average precision (MAP@50) of 0.7520 value with BoW, followed with slightly lower value of 0.7518 by TF-IDF. The recall rate@50 of BoW and TF-IDF is the same value which is 0.6200 and it is higher than the baseline method. The full evaluation shows that BoW and TF-IDF has a better performance than baseline method across all classes.

## Suggestions for Improvement

For certain queries, baseline method will outperform the SIFT BoW and TF-IDF features. For example querying with 145.jpg from the database, baseline method actually beats both BoW and TF-IDF for the first 10 images retrieved.

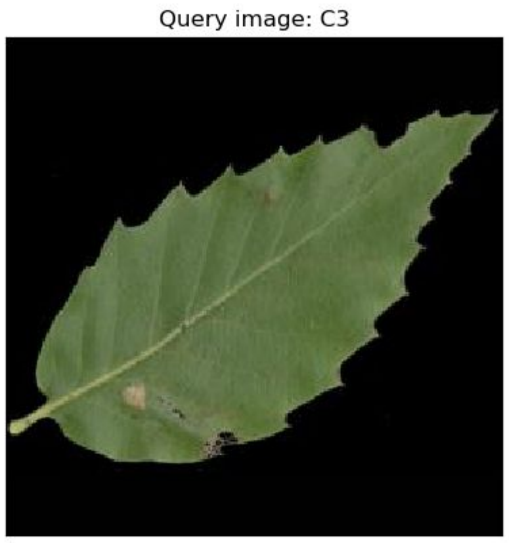


Figure 35 : Query Image

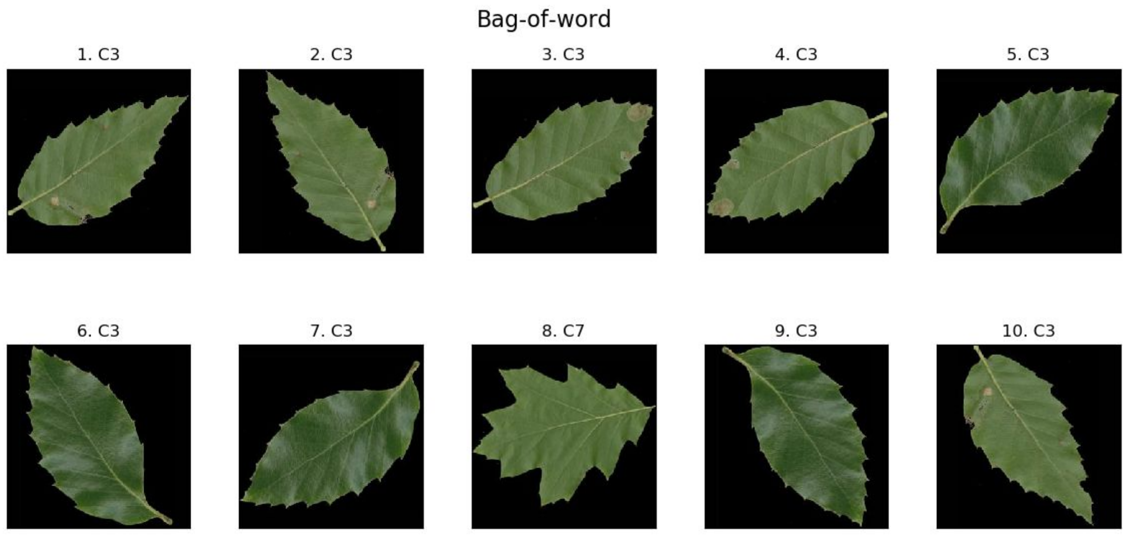


Figure 36 : Image Retrieval from database by BoW

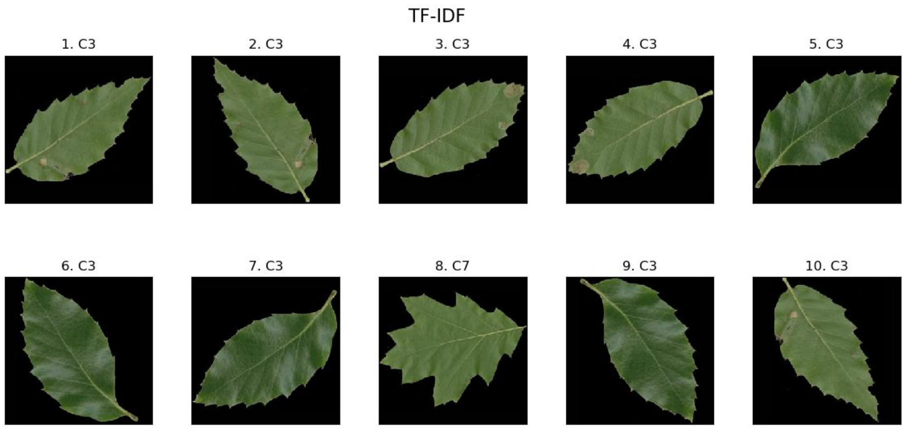


Figure 37 : Image Retrieval from database by TF-IDF

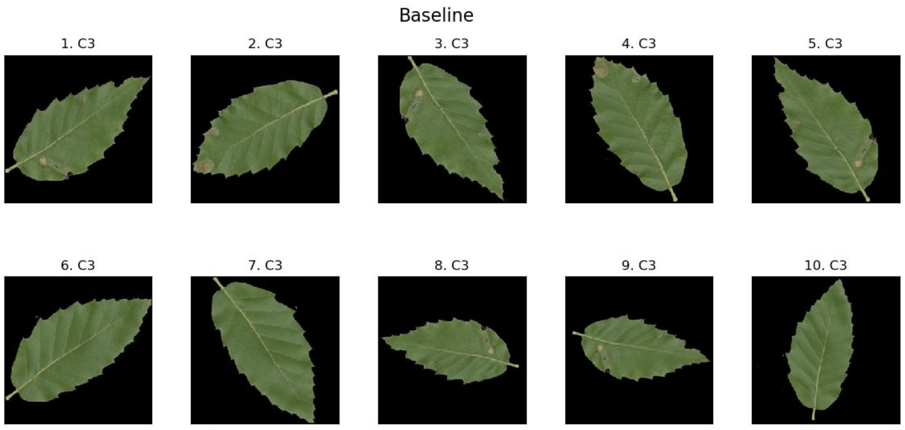


Figure 38 : Image Retrieval from database by Baseline

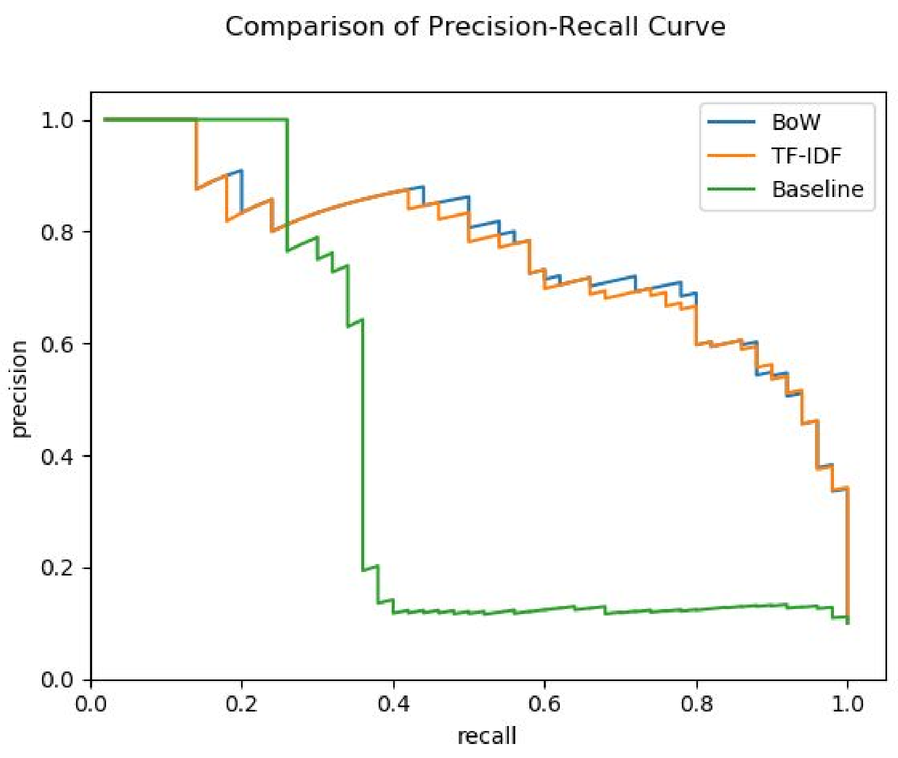


Figure 39 : Precision-Recall (PR) curve for 145.jpg image query

In this case, although the PR curve shows that overall the average precision of both BoW and TF-IDF. This is due to the fact that SIFT doesn't extract color as features from the image, while baseline method rely on the color histogram to as features, and it is effective, at least for the first 10 images.

Also in querying C2, we can see that both TF-IDF and BoW are confused by C9, which is a visually similar class. This shows the model can yet to be improved.

Now we've identified the problem with this model, we propose to improve it by adding features such as color histogram, which is used by the baseline method, this may help the model to use color to identify the similar images. Next, we can also fine tune the K-means model with different number of K. We can also try out different seed value, since it was indetermistic by default, which means we might get different features each run. By setting the seed value to a constant number, we can easily replicate the result each time.

The contrast threshold of the SIFT function, can be adjusted to filter away insignificant key points, thus it might improve the clustered visual word.

Other feature extraction methods such as SURF, may improve the speed of feature extraction, although in our assignment this is not a problem as SIFT is known to be slower but more accurate than SURF.