Building a text based, image based and a multimodal search engine

1st Ayush Goyal
Department of Mathematics
Indian Institute Of Technology, Delhi
New Delhi, India
2019MT10961

2nd Avadhesh Prasad
Department of Mathematics
Indian Institute Of Technology, Delhi
New Delhi, India
2019MT60747

3rd Aman Chauhan
Department of Mathematics
Indian Institute Of Technology, Delhi
New Delhi, India
2019MT60742

4th Saransh Agarwal
Department of Mathematics
Indian Institute Of Technology, Delhi
New Delhi, India
2019MT60763

5th Harsh Sharma
Department of Mathematics
Indian Institute Of Technology, Delhi
New Delhi, India
2019MT60628

***Abstract*—The web is the huge and most extravagant well-spring of data. To recover the information from World Wide Web, Search Engines are commonly utilized. Search engines provide a simple interface for searching for user query and displaying results in the form of the web address of the relevant web page, but using traditional search engines has become very challenging to obtain suitable information**

This paper proposed three search engine

1) **First search engine is a text-based search engine which takes as input a word (any word!), an algo (Algo 1:TF-IDF, Algo 2: Word2Vec) and outputs the articles (Wikipedia articles) which are closest to the given word**

2) **Second search engine is an image based search engine which takes as input an image to output the images which are visually closest to the input image using SIFT and Bag of Visual Words to build an index and to perform the search**

3) **Third search engine is a multi modal search engine that uses the previous two search engines. It takes in a word or an image as input and outputs the closest words and images**

***Index Terms*—World Wide Web, Search Engine, TF-IDF, Word2Vec, SIFT, Bag of Visual Words, Machine Learning.**

I. Introduction

A search engine is a software system that is designed to carry out web searches. They search the World Wide Web in a systematic way for particular information specified in a textual web search query. The search results are generally presented in a line of results, often referred to as search engine results pages (SERPs). The information may be a mix of links to web pages, images, videos, infographics, articles, research papers, and other types of files. Some search engines also accepts image queries.

This paper utilizes Machine Learning Techniques to discover the utmost suitable articles/images for the given query.

The section II reviews the literature used to build our search engine. In Section III Objective is explained. Section IV deals with proposed system which is based on machine learning technique and Section V contains the final observations.

II. Literature Review

A. TF-IDF

In information retrieval, tf–idf short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. tf–idf is one of the most popular term-weighting schemes today.

B. Word2Vec

Word2vec is a technique for natural language processing published in 2013. The word2vec algorithm uses a neural network model to learn word associations from a large corpus of text. Once trained, such a model can detect synonymous words or suggest additional words for a partial sentence. As the name implies, word2vec represents each distinct word with a particular list of numbers called a vector. The vectors are chosen carefully such that a simple mathematical function (the cosine similarity between the vectors) indicates the level of semantic similarity between the words represented by those vectors.

C. SIFT

The scale-invariant feature transform (SIFT) is a computer vision algorithm to detect, describe, and match local features in images, invented by David Lowe in 1999. Applications include

object recognition, robotic mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking, individual identification of wildlife and match moving.

SIFT keypoints of objects are first extracted from a set of reference images and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. From the full set of matches, subsets of keypoints that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches. The determination of consistent clusters is performed rapidly by using an efficient hash table implementation of the generalised Hough transform. Each cluster of 3 or more features that agree on an object and its pose is then subject to further detailed model verification and subsequently outliers are discarded. Finally the probability that a particular set of features indicates the presence of an object is computed, given the accuracy of fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence.

D. Bag of Visual Words

In computer vision, the bag-of-words model (BoW model) sometimes called bag-of-visual-words model can be applied to image classification or retrieval, by treating image features as words. In document classification, a bag of words is a sparse vector of occurrence counts of words; that is, a sparse histogram over the vocabulary. In computer vision, a bag of visual words is a vector of occurrence counts of a vocabulary of local image features.

To represent an image using the BoW model, an image can be treated as a document. Similarly, "words" in images need to be defined too. To achieve this, it usually includes following three steps: feature detection, feature description, and codebook generation. A definition of the BoW model can be the "histogram representation based on independent features". Content based image indexing and retrieval (CBIR) appears to be the early adopter of this image representation technique.

Computer vision researchers have developed several learning methods to leverage the BoW model for image related tasks, such as object categorization. These methods can roughly be divided into two categories, unsupervised and supervised models. For multiple label categorization problem, the confusion matrix can be used as an evaluation metric.

III. Objective

To build a text based, image based and a multi modal search engine

For text based search engine, Given a set of N words/phrases, we downloaded text (not html) of corresponding wikipedia articles as our database. The search engine takes as input a word and an algo (textbfAlgo 1:TF-IDF, Algo 2:Word2Vec) to output the following

1) Article which are closest to the given word (ranked from 1 to N)
2) Closest possible search terms for the search engine
3) How many articles appear in the same group with both the algorithms with the given search term
4) Time taken to make the search

For the image based search engine, using the same set of N words, we downloaded top 50 images from Google Images, resized them to be of max 1000x1000 pixels maintaining the aspect ratio without resizing the smaller images to be our database for search. The search engine takes as input an image and an algo (textbfAlgo 1:SIFT, Algo 2:Bag of Visual Words) to output the following

1) The images which are visually closest to the input image and also their tag (i.e. the word we used to download it). (Ranked from 1 to 50)
2) The statistics i.e. the time taken to retrieve them and how many images from the same search word are in top-50 results

For the multi-modal search engine, we use the previous two search engines. The search engine takes as input a word or an image to output the following

1) The closest words and images (Ranked).
2) All the statistics developed before.

IV. Methodology

A. Text-based Search Engine:

Data pre-processing
Steps involved are:
1. LowerCase
2. Word Tokenization
3. Removed Stopwords
4. Word Lemmatization

B. TF-IDF

Formula for calculating the tf-idf value:
tf-idf(t,d) = tf(t,d) * $log(N/df + 1)$
Where tf - term frequency, N - total number of documents,
t - query, d - document under consideration,
df - number of documents containing the query word,
tf(t,d) - frequency of query word in the document under consideration.

Here we have generated TF-IDF using the TfidfVectorizer from Sklearn library. Then we create the vector for the input query. For each article, we find the cosine similarity between the vectors and using that we rank the articles. For finding the closest search terms from the index, we found the cosine similarity between the vectors of the tokens(clean words present in the document) and the query vector, and take out the top 10 closest search terms.

C. Word2Vec

Here we need to find the word embeddings (word vector) of all the tokens (clean words of every document). For generating the word embeddings, here we use the Fasttext module from the gensim library (Fasttext is used so that our text based search engine can even handle unseen words which are not present in the vocabulary generated from the tokens from all the given documents). We find the vector against our input query too. Now using these vectors, we find the cosine-similarity between these vectors and for each article we find the token which has max-similarity with the query word and rank the articles on the basis of this max-similarity value. In the process we have found the similarity vlaue of each token, so using these values we find the closest search terms of the query word in the index.

D. SIFT

What we do here in this algorithm is that given a query image we find images in our database that are closest visually to the query image. For this we first find the keypoints and descriptors of the query image as well as all the images in the dataset using SIFT. We then use a Flann based matcher to find the matches between the query image and the corresponding image in our dataset. We have used flann matcher as it gives better time performance than the brute force matcher. Once all the matches are found, we find the good matches and discard the rest of the matches using ratio test. Once this is done, the closest image will be the one with the maximum number of good matches. Thus, we sort the list of good matches in descending order which will give us the closest N images for the query image. Now we draw the images showing the matches between the query image and the ones from the dataset.

E. Bag Of Visual Word

In this algorithm, for a given query image we find the closest visually related image from our database.For this, we find keypoints and descriptors for every image using inbuilt extractor like sift or orb. We then built visual words from these descriptors using kmeans algorithm which provide us k-clusters(visual word). we then use these visual words to plot histogram for every image in the dataset using predict function on visual words. Further, we plot histogram for the given query image. Then we use KNN algorithm as a similarity matching algorithm to find the k nearest histogram which are closest to query image histogram. The closest histogram will corresponds to more similar images i.e. more visually related images and are given lower ranks. Now, we draw all the output images and our query image.

F. Multimodal Search Engine

For multimodal search engine we are given a word or image and we have to find closest words and image. We have a curated dataset where we have a mapping of documents and images (as both are for top 100 Wikipedia articles).In the feature space we will find neighbourhood of text with images and vice Versa in feature space like Text-¿ image. We have used association of word2vec with image features (using SIFT) for the above task and maintain a dictionary for the same.

If the given query is a word then by using word2vec we find closest embeddings for the given word and then find closest image for the corresponding embeddings from our dataset. For example if a given word is iit delhi then iit madras is similar word has iit is a common word. Suppose if we have King then Queen is similar word as it is closer in semantic space.

So if we have an image of a king or text input King, the documents and images related to king will rank higher followed by those of queen if no other word is similar semantically or in spelling to King.

If the given query is a image then we use sift algorithm for finding closest image and then use the corresponding tags for the closest words.

G. Integrating Audio to a search engine

Audio-based semantic models enable the computation of word semantic representations by taking into account the association of tags with audio clips in clip collections.We used the Bag of Audio Words (BoAW) technique to do this. We combine the linguistic and auditory features to accomplish this. A three step process is employed to do this.First, we extract the acoustic (sound/audio) features from segmented audio clips. Then we compute the clip vector representations using the Bag of Audio Words (BoAW) method. What essentially is done is that the acoustic features extracted from the segments of a clip are quantized to the nearest clusters (called as audio words) like we did in the case of Bag of Visual Words (BoVW). This clip is then represented as a bag/histogram of clusters/audio words. In the third step we compute the tag representations where each tag derives a BoAW representation by averaging the representations of the clips annotated with this tag.

H. Integrating Video to a search engine

One of the key steps to integrate video to a search engine is searching for videos by image. It is the process retrieving from the corpus of videos containing similar frames to the input image. To achieve this we turn videos into embeddings which is to say that we extract the key (important) frames and convert their features (image features) into vectors. This is done by cutting the query video into frames and extract vectors of the key frames using image feature extraction models like VGG and then insert the extracted vectors (embeddings) into Milvus. For video search, we use same VGG model to convert the input image into a feature vector and insert it into Milvus to find the most similar vectors. Finally, we retrieve the corresponding videos from Minio on its interface according to the correlation values between our query video and the test video that we have found.

V. Experimental Results

Input query: **steve**

A. TF-IDF

Articles closest to given word ranked(from 1 to N):
(1, 'Steve Jobs')
(2, 'Jennifer Aniston')
(3, 'The Beatles')
(4, 'Taylor Swift')
(5, 'Ariana Grande')
(6, 'Kim Kardashian')
(7, 'COVID-19 pandemic')
(8, 'Leonardo DiCaprio')
(9, 'Justin Bieber')
(10, 'Bill Gates')
(11, 'London')
(12, 'Star Wars')
(13, 'List of highest-grossing films')
(14, 'China')
(15, 'Darth Vader')
(16, 'Miley Cyrus')
(17, 'Australia')
(18, 'Abraham Lincoln')
(19, 'September 11 attacks')
(20, 'Lil Wayne')
(21, 'Academy Awards')
(22, 'Japan')
(23, 'Johnny Depp')
(24, 'Germany')
(25, 'LeBron James')
(26, 'New York City')
(27, 'Harry Potter')
(28, 'Kobe Bryant')
(29, 'Stephen Hawking')
(30, 'Dwayne Johnson')
(31, 'List of Presidents of the United States')
(32, 'The Big Bang Theory')
(33, 'Donald Trump')
(34, 'Barack Obama')
(35, 'Elizabeth II')
(36, 'India')
(37, 'World War II')
(38, 'Michael Jackson')
(39, 'United Kingdom')
(40, 'Cristiano Ronaldo')
(41, 'Lady Gaga')
(42, 'Sex')
(43, 'Adolf Hitler')
(44, 'Eminem')
(45, 'Game of Thrones')
(46, 'World War I')
(47, 'Elon Musk')
(48, 'Canada')
(49, 'Freddie Mercury')
(50, 'Lionel Messi')
(51, 'Michael Jordan')
(52, 'Selena Gomez')
(53, 'Russia')
(54, 'Rihanna')
(55, 'Albert Einstein')
(56, 'Muhammad Ali')
(57, 'Ted Bundy')
(58, 'Nicky Minaj')
(59, 'Will Smith')
(60, 'Singapore')
(61, 'Israel')
(62, 'John Cena')
(63, 'Bruce Lee')
(64, 'Elvis Presley')
(65, 'Diana, Princess of Wales')
(66, 'Charles Manson')
(67, 'Marilyn Monroe')
(68, 'Sexual intercourse')
(69, 'Katy Perry')
(70, 'Winston Churchill')
(71, 'Tom Brady')
(72, 'Periodic Table')
(73, 'Glee0(TV series)')
(74, 'Brad Pitt')
(75, 'Madonna')
(76, 'Britney Spears')
(77, 'Earth')
(78, 'William Shakespeare')
(79, 'Mark Zuckerberg')
(80, 'Joe Biden')
(81, 'Adele')
(82, 'The Walking Dead (TV series)')
(83, 'How I Met Your Mother')
(84, 'Kanye West')
(85, 'Tupac Shakur')
(86, 'Angelina Jolie')
(87, 'John F. Kennedy')
(88, 'Scarlett Johansson')
(89, 'List of Marvel Cinematic Universe films')
(90, 'Chernobyl disaster')
(91, 'Queen Victoria')
(92, 'France')
(93, 'Tom Cruise')
(94, 'Breaking Bad')
(95, 'Arnold Schwarzenegger')
(96, 'Pablo Escobar')
(97, 'Keanu Reeves')
(98, 'Mila Kunis')
(99, 'Vietnam War')
(100, 'Meghan, Duchess of Sussex')
(101, 'United States')

10 closest search terms to the query word from index:
1 humility
2 reiterate
3 hereditary

4 colonist
5 reliable
6 receiver
7 refresher
8 vintage
9 mountain
10 paths

Time Taken to make the search:
CPU times: user 148 ms, sys: 99.5 ms, total: 248 ms Wall time: 147 ms

B. Word2Vec

Articles closest to given word ranked:
(1, 'The Beatles')
(2, 'Justin Bieber')
(3, 'Kim Kardashian')
(4, 'Steve Jobs')
(5, 'Taylor Swift')
(6, 'Leonardo DiCaprio')
(7, 'COVID-19 pandemic')
(8, 'Ariana Grande')
(9, 'Jennifer Aniston')
(10, 'Bill Gates')
(11, 'Angelina Jolie')
(12, 'Tom Cruise')
(13, 'Keanu Reeves')
(14, 'Muhammad Ali')
(15, 'Brad Pitt')
(16, 'World War I')
(17, 'September 11 attacks')
(18, 'Earth')
(19, 'Tupac Shakur')
(20, 'Marilyn Monroe')
(21, 'Diana, Princess of Wales')
(22, 'Elvis Presley')
(23, 'United States')
(24, 'Abraham Lincoln')
(25, 'Meghan, Duchess of Sussex')
(26, 'Britney Spears')
(27, 'Lionel Messi')
(28, 'France')
(29, 'United Kingdom')
(30, 'Dwayne Johnson')
(31, 'Australia')
(32, 'Ted Bundy')
(33, 'Germany')
(34, 'Elon Musk')
(35, 'LeBron James')
(36, 'New York City')
(37, 'The Walking Dead (TV series)')
(38, 'Breaking Bad')
(39, 'Arnold Schwarzenegger')
(40, 'Pablo Escobar')
(41, 'Charles Manson')
(42, 'Madonna')
(43, 'Tom Brady')
(44, 'Michael Jackson')
(45, 'Freddie Mercury')
(46, 'Scarlett Johansson')
(47, 'Katy Perry')
(48, 'Bruce Lee')
(49, 'List of highest-grossing films')
(50, 'China')
(51, 'Mark Zuckerberg')
(52, 'Sexual intercourse')
(53, 'Winston Churchill')
(54, 'Adele')
(55, 'India')
(56, 'Game of Thrones')
(57, 'Michael Jordan')
(58, 'Darth Vader')
(59, 'Miley Cyrus')
(60, 'Lil Wayne')
(61, 'Albert Einstein')
(62, 'Vietnam War')
(63, 'Queen Victoria')
(64, 'Glee0(TV series)')
(65, 'Chernobyl disaster')
(66, 'Donald Trump')
(67, 'Canada')
(68, 'John F. Kennedy')
(69, 'Will Smith')
(70, 'Barack Obama')
(71, 'Elizabeth II')
(72, 'Cristiano Ronaldo')
(73, 'Adolf Hitler')
(74, 'Lady Gaga')
(75, 'Japan')
(76, 'Rihanna')
(77, 'Russia')
(78, 'Nicky Minaj')
(79, 'Stephen Hawking')
(80, 'Joe Biden')
(81, 'London')
(82, 'Harry Potter')
(83, 'Israel')
(84, 'The Big Bang Theory')
(85, 'John Cena')
(86, 'Periodic Table')
(87, 'Academy Awards')
(88, 'Johnny Depp')
(89, 'Kobe Bryant')
(90, 'List of Marvel Cinematic Universe films')
(91, 'William Shakespeare')
(92, 'Eminem')
(93, 'Selena Gomez')
(94, 'Kanye West')
(95, 'Mila Kunis')
(96, 'World War II')

(97, 'Sex')
(98, 'Star Wars')
(99, 'How I Met Your Mother')
(100, 'Singapore')
(101, 'List of Presidents of the United States')

10 closest search terms to the query:
1 apple
2 tomorrow
3 buckle
4 spearhead
5 hafnium
6 wrongful
7 sawtelle
8 magdalene
9 pournelle
10 bienstock

Time taken to make the search:
CPU times: user 4.9 s, sys: 53.4 ms, total: 4.96 s Wall time: 4.94 s

Rank in Word2Vec of first 10 closest articles in tf-idf to query:
Article: Steve Jobs
Rank in tf-idf: 1
Rank in Word2Vec: 4
Article: Jennifer Aniston
Rank in tf-idf: 2
Rank in Word2Vec: 9
Article: The Beatles
Rank in tf-idf: 3
Rank in Word2Vec: 1
Article: Taylor Swift
Rank in tf-idf: 4
Rank in Word2Vec: 5
Article: Ariana Grande
Rank in tf-idf: 5
Rank in Word2Vec: 8
Article: Kim Kardashian
Rank in tf-idf: 6
Rank in Word2Vec: 3
Article: COVID-19 pandemic
Rank in tf-idf: 7
Rank in Word2Vec: 7
Article: Leonardo DiCaprio
Rank in tf-idf: 8
Rank in Word2Vec: 6
Article: Justin Bieber
Rank in tf-idf: 9
Rank in Word2Vec: 2
Article: Bill Gates
Rank in tf-idf: 10
Rank in Word2Vec: 10

Then user is given the choice of selecting an algorithm and article of his choice:
Suppoe the selected algorithm is word2vec, and selected article is Leonardo DiCaprio

link of the article: Leonardo DiCaprio

tf-idf rank of article with respect to query: 8
Word2Vec rank of article with respect to query: 6

10 Closest Search terms using word2vec of the query in the selected article: 1 apple
2 tee
3 rio
4 robert
5 felt
6 mine
7 similar
8 emi
9 homo
10 breakthrough

C. SIFT

Our query image for the SIFT part is as follows.

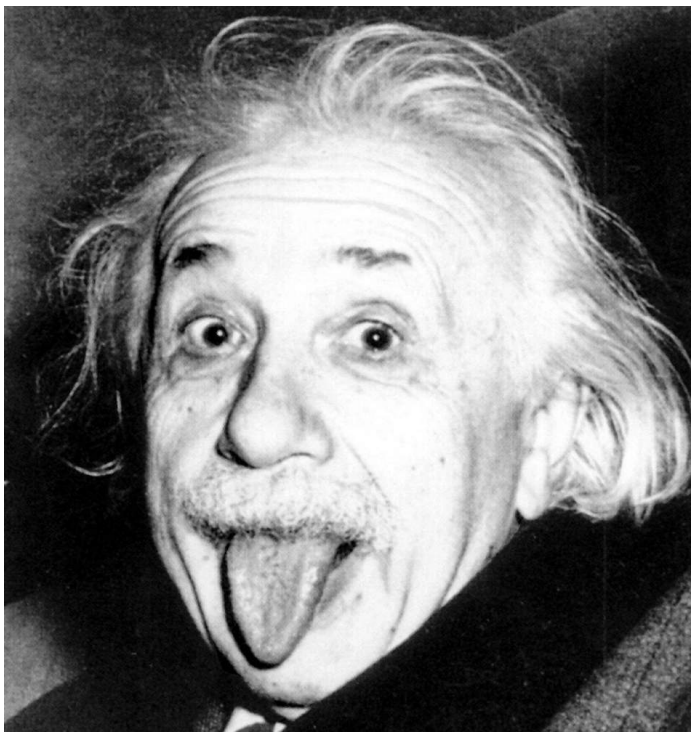

Fig. 1. Albert Einstein

The matches between the above query image and the closest images from the dataset are as follows:-

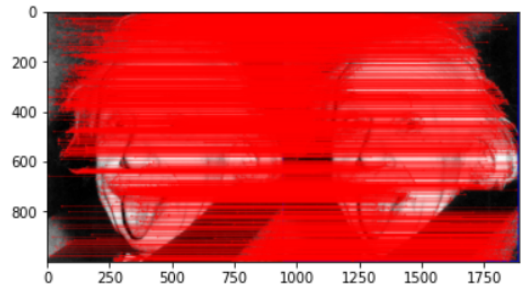

Fig. 2. rank 1 match

We can observe that we have found a correct match for our query image as in fig 2 which is of Albert Einstein and the mapping of the features between both the images is correct. This is the closest image from the dataset.

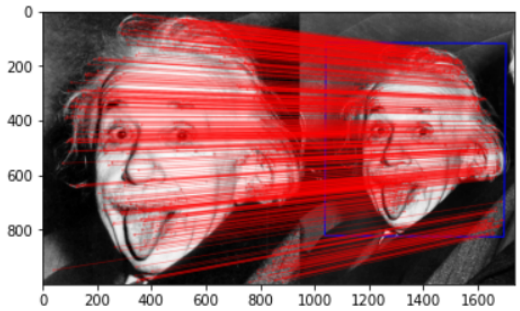

Fig. 3. rank 2 match

The match as in fig 3 is the next closest match for our query image. Again the match is correct which is that of Albert Einstein and the corresponding features are correctly matched.

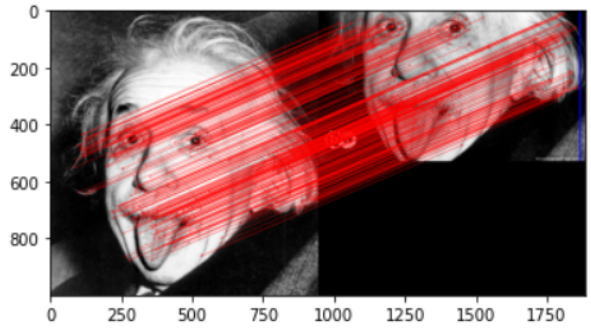

Fig. 4. rank 2 match

The match in fig 4 is the third closest match for the query image. Here too we have a true match of Albert Einstein and the corresponding features of eyes, noses etc are correctly matched.

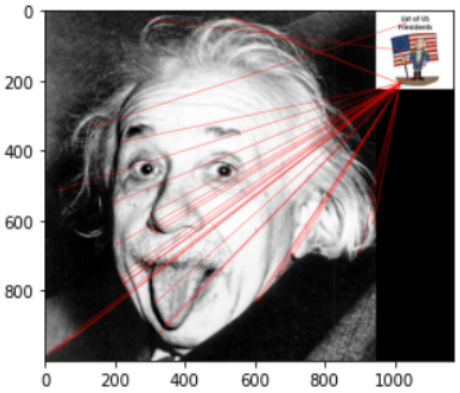

Fig. 5. rank 4 match

The match in fig 5 is an incorrect match. It matches our query image of Albert Einstein to the list of president of the United States which is incorrect.

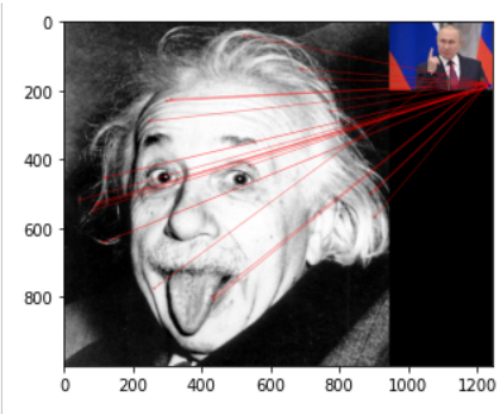

Fig. 6. rank 5 match

The match in fig 6 is another incorrect match. It matches our query image of Albert Einstein to that to Russia which again is wrong.

Time taken by the SIFT algorithm to find the closest matches for our query image = 1186.95s

D. Bag of Visual Word

Here, Figure 7 corresponds to the input image in case of bag of visual words.

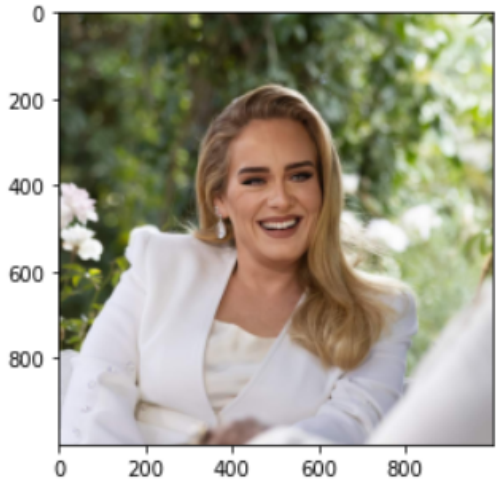

Fig. 7. Input pic: Adele

Figure 8 is the overall histogram for k=5 clusters.

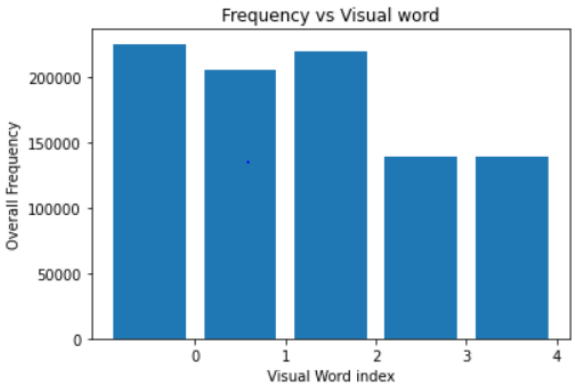

Fig. 8. Histogram

Fig 9 is the 1st closest image for Adele which is a correct match.

Fig. 9. rank1 image: Adele

Fig 10 is the 2nd closest image for Adele which is a correct match.

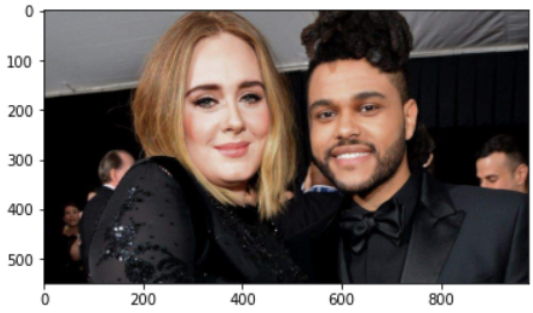

Fig. 10. rank2 image: Adele

Fig 11 is the 3rd closest image for Adele which is an incorrect match as it matches Adele to Adolf Hitler.

Fig. 11. rank3 image: Adolf Hitler

Figure 12 is the overall histogram for k=10 clusters.

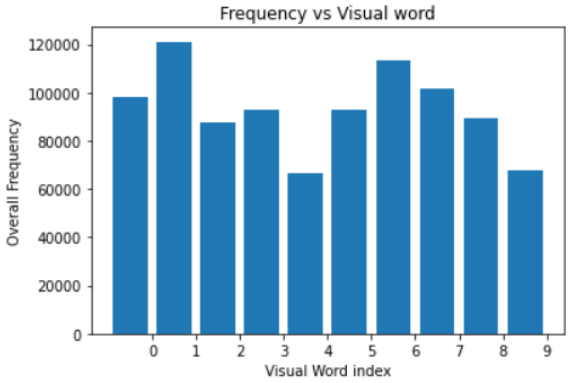

Fig. 12. Histogram

Fig 13 is the 1st closest image for Adele which is a correct match.

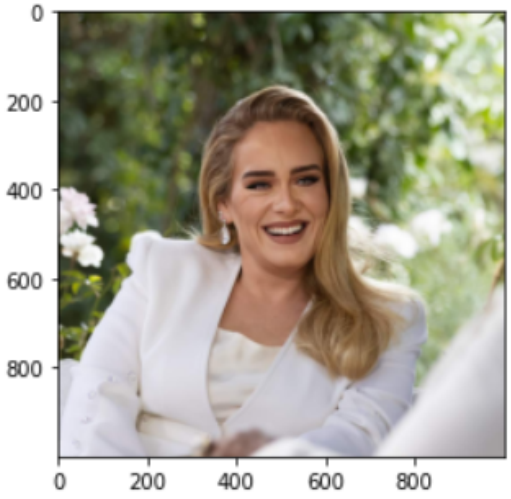

Fig. 13. rank1 image: Adele

Fig 14 is the 2nd closest image for Adele which is a correct match.

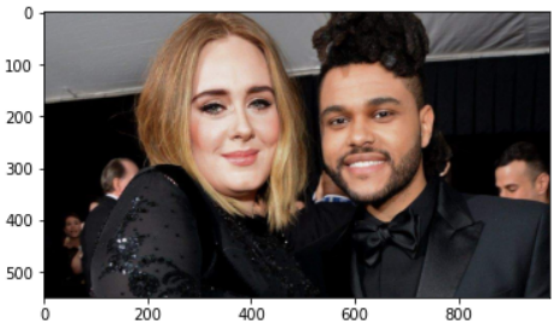

Fig. 14. rank2 image: Adele

Fig 15 is the 3rd closest image for Adele which is a correct match.

Fig. 15. rank3 image: Adele

Fig 16 is the overall histogram for k=50 clusters.

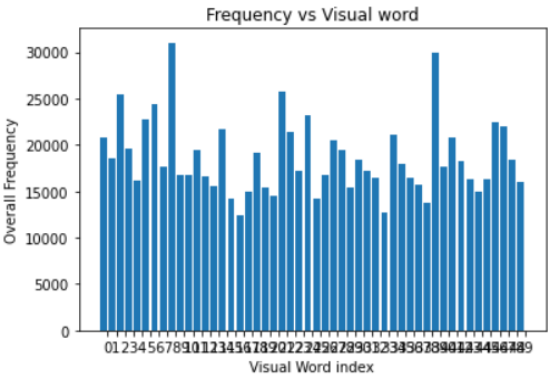

Fig. 16. Histogram

Fig 17 is the 1st closest image for Adele which is a correct match.

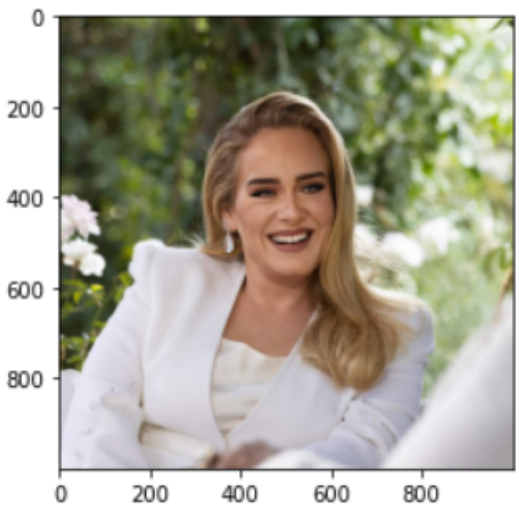

Fig. 17. rank1 image: Adele

Fig 18 is the 2nd closest image for Adele which is a correct match.

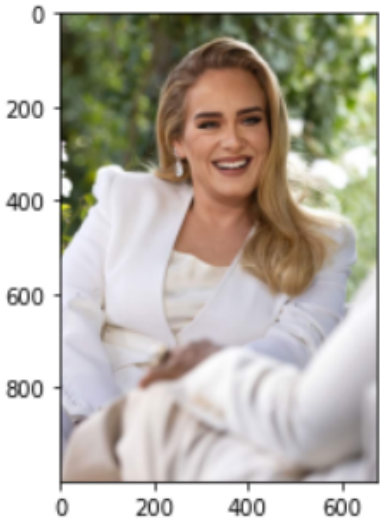

Fig. 18. rank2 image: Adele

Fig 19 is the 3rd closest image for Adele which is a correct match.

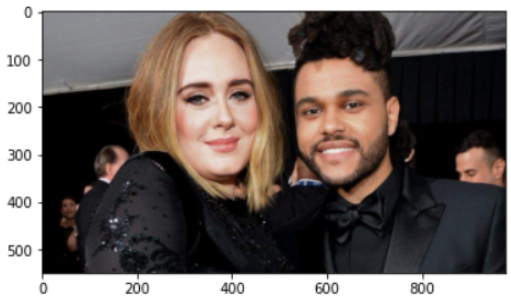

Fig. 19. rank3 image: Adele

Fig 20 is the 4th closest image for Adele which is a correct match.

Fig. 20. rank4 image: Adele

Fig 21 is the 3rd closest image for Adele which is an incorrect match as it matches Adele to Angelina Jolie.

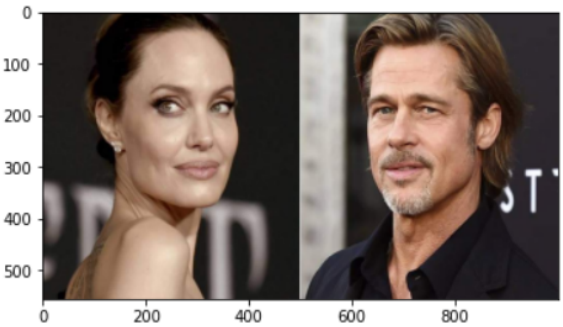

Fig. 21. rank5 image: Angeline Jolie

Fig 22 is the overall histogram for k=100 clusters.

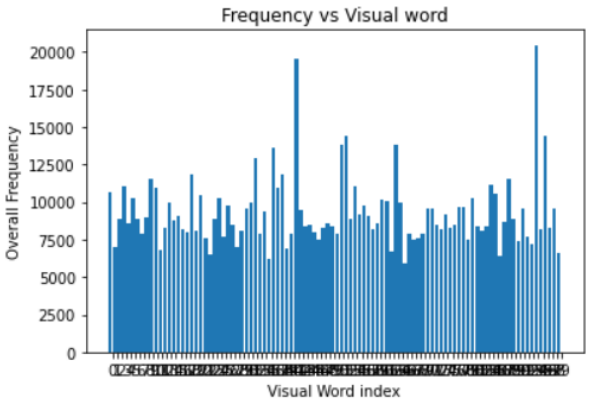

Fig. 22. Histogram

Fig 23 is the 1st closest image for Adele which is a correct match.

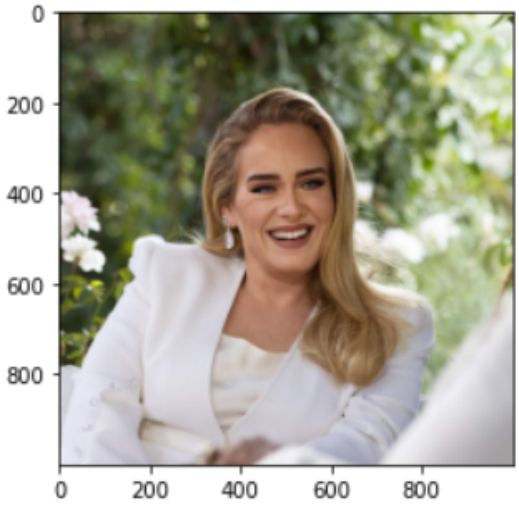

Fig. 23. rank1 image: Adele

Fig 24 is the 2nd closest image for Adele which is a correct match.

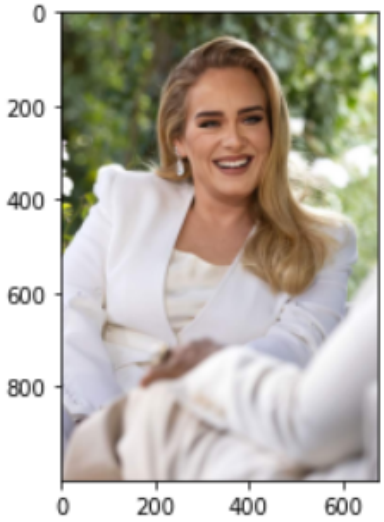

Fig. 24. rank2 image: Adele

Fig 25 is the 3rd closest image for Adele which is a correct match.

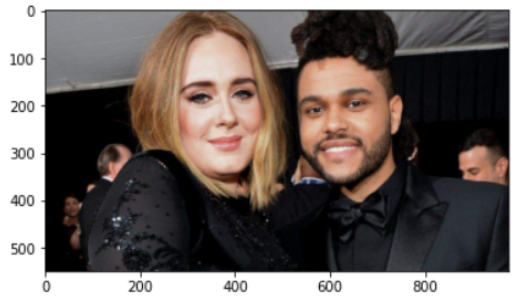

Fig. 25. rank3 image: Adele

Fig 26 is the 4th closest image for Adele which is a correct match.

Fig. 26. rank4 image: Adele

Fig 27 is the 5th closest image for Adele which is an incorrect match as it matches Adele to Angelina Jolie.

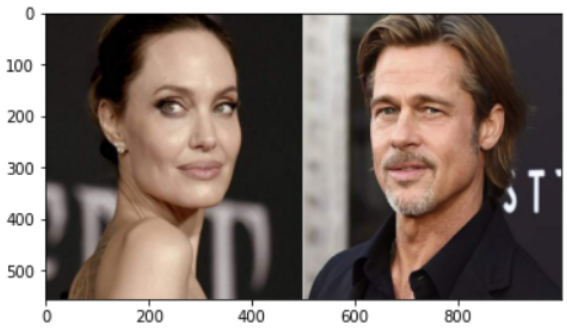

Fig. 27. rank5 image: Angelina Jolie

Fig 28 is the overall histogram for k=500 clusters.

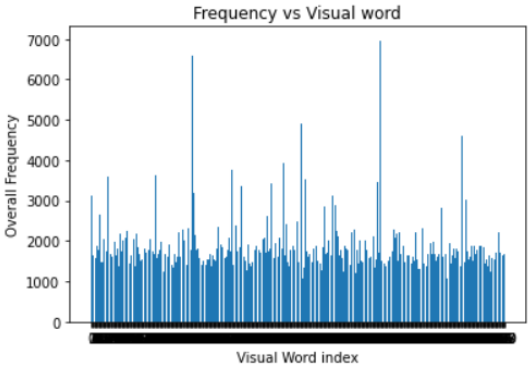

Fig. 28. Histogram

Fig 29 is the 1st closest image for Adele which is a correct match.

Fig. 29. rank1 image: Adele

Fig 30 is the 2nd closest image for Adele which is a correct match.

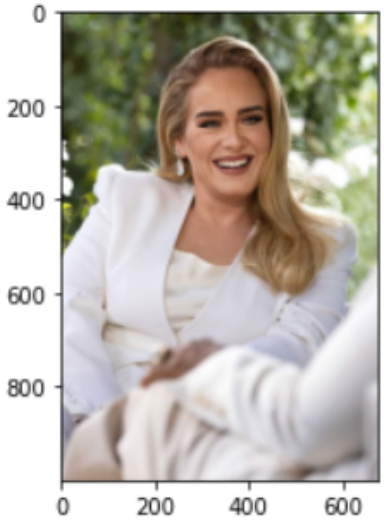

Fig. 30. rank2 image: Adele

Fig 31 is the 3rd closest image for Adele which is a correct match.

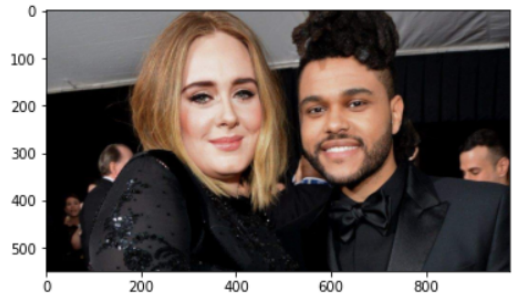

Fig. 31. rank3 image: Adele

Fig 32 is the 4th closest image for Adele which is an incorrect match as it matches Adele to Australia.

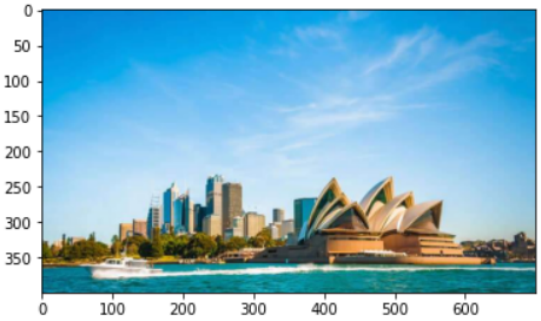

Fig. 32. rank4 image: Australia

Fig 33 is the 5th closest image for Adele which is an incorrect match as it matches Adele to Ariana grande.

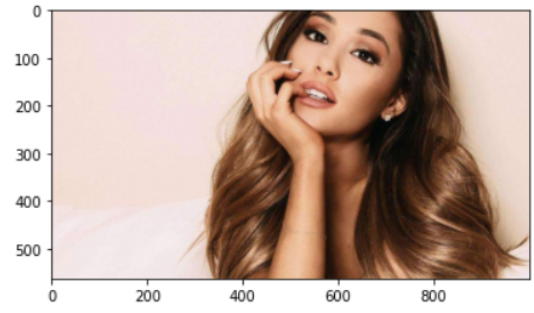

Fig. 33. rank5 image: Ariana Grande

Fig 34 is the 6th closest image for Adele which is a correct match.

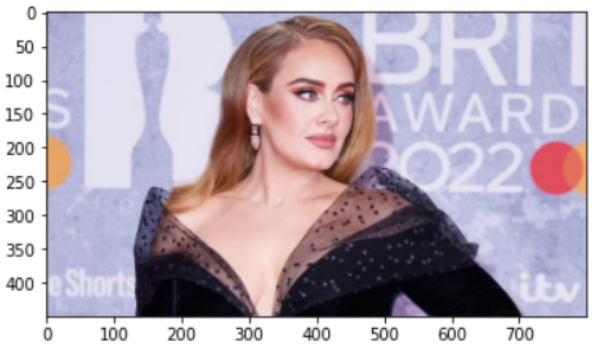

Fig. 34. rank6 image: Adele

VI. Analysis

A. SIFT

Time taken by the SIFT algorithm to find the closest matches for our query image = 1186.95s

When we ran our SIFT algorithm we observed that from the five closest images for our query image, three of them are correct matches while the last two matches are incorrect. Also we observed that we have a lot more matches in the first three images (each red line in the image represents a match) whereas the number of matches in the last two images are very low. Thus, one possible reason for the incorrect match is the low threshold we have chosen to decide if a given match is a good match. Thus, we should increase the number of matches threshold to get better matches for the closest images. Thus, our SIFT algorithm works well for the given query input and by varying the threshold we can fine tune it to give better performance.

B. Bag of Visual Words

Bag of visual words algorithm has been implemented to search the visually related image to the query image for k number of clusters.

Number of visual words	correct matches	incorrect matches	Time taken
5	2	1	0.989s
10	3	0	2.099s
50	4	1	2.347
100	4	1	2.519
500	4	2	2.99s

TABLE I
Comparison for different values of visual words

So, we can see on increasing the number of visual words, correct matches started increasing as for k=5, correct matches are 2 while for k=10,50, correct matches are 3 and 4 respectively and then incorrect matches started increasing afterwards as for k=50,100 incorrect matches are 1 while for k=500, incorrect matches are 2.

We got the best matches at k=50

So, we can see that with respect to number of visual words taken, first underfitting is happening till k=100, and then k=100

is best fit and then we get overfitting in case of k=500.
Here, from the table we can deduct that time taken by the code is increasing with respect to increasing value of number of visual words which is because size of histogram is increasing as increase in number of visual words.

VII. Conclusion

In this assignmemt we downloaded a dataset of top 100 wikipedia articles and the top 50 images corresponding to those 100 words. We ran the TF-IDF algorithm and the Word2Vec algorithm on the 100 words dataset by taking a query word as input and returning the ranked list of the closest words from our dataset as output.

We also ran the SIFT algorithm and the Bag of Visual Words (BoVW) algorithm on the image dataset by taking a query image as input and returning the closest ranked list of images from the database as output.

We also took a random subset of 20 words from our downloaded dataset and trained the above algorithm on the 20 words and their respective 50 images. Finally we tested our algorithms on a new test set of 30 words and the images for those words.

References

[1] 'Search Engine' (2022) Wikipedia. Available at: https://en.wikipedia.org/wiki/Search_engine (Accessed: 14 April 2022).

[2] 'tf–idf' (2022) Wikipedia. Available at: https://en.wikipedia.org/wiki/Tf-idf (Accessed: 14 April 2022).

[3] 'Word2vec' (2022) Wikipedia. Available at: https://en.wikipedia.org/wiki/Word2vec (Accessed: 14 April 2022).

[4] 'TF-IDF' (2022). Available at: https://towardsdatascience.com/tf-idf-explained-and-python-sklearn-implementation-b020c5e83275 (Accessed: 14 April 2022).

[5] 'Word2Vec' (2022) . Available at: https://www.analyticsvidhya.com/blog/2020/08/information-retrieval-using-word2vec-based-vector-space-model/ (Accessed: 14 April 2022).

[6] 'Scale-invariant feature transform' (2022) Wikipedia. Available at: https://en.wikipedia.org/wiki/Scale-invariant_feature_transform (Accessed: 14 April 2022).

[7] 'Bag-of-words model in computer vision ' (2021) Wikipedia. Available at: https://en.wikipedia.org/wiki/Bag-of-words_model_in_computer_vision (Accessed: 14 April 2022).

[8] 'Scale-invariant feature transform'. Available at:https://docs.opencv.org/3.4/d1/de0/tutorial_py_feature_homography.html (Accessed: 17 April 2022).

[9] 'Video Search Engine'.Available at:https://dzone.com/articles/4-steps-to-building-a-video-search-system (Accessed: 17 April 2022).

[10] 'Audio Search Engine'.Available at:https://pure.unic.ac.cy/en/publications/sensory-aware-multimodal-fusion-for-word-semantic-similarity-esti (Accessed: 17 April 2022).