RAMAKRISHNA MISSION RESIDENTIAL COLLEGE (AUTONOMOUS)
RAMAKRISHNA MISSION ASHRAMA
NARENDRAPUR, KOLKATA-700 103, W. B., INDIA
(A Branch Centre of Ramakrishna Mission, Belur Math, Howrah-711202)

Project Completion Certificate

Date: 03/06/2020

This is to certify that **Sri Aritra Raut** (ROLL NO: CSUG/205/17) a student of Department of Computer Science, Ramakrishna Mission Residential College(Autonomous), has undergone a Project work from January 15, 2020 to May 30, 2020 titled **"Image Steganography Using Different Methods".**

Department of Computer Science
Ramakrishna Mission Residential College
(Autonomous)
Narendrapur, Kolkata-700 103

Dr. Siddhartha Banerjee
(Head of the Department)
Department of Computer Science

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE (AUTONOMOUS)
RAMAKRISHNA MISSION ASHRAMA
NARENDRAPUR, KOLKATA-700 103, W. B., INDIA
(A Branch Centre of Ramakrishna Mission, Belur Math, Howrah-711202)

Project Completion Certificate

Date: 03/06/2020

This is to certify that **Sri Sukhomoy Debnath** (ROLL NO: CSUG/206/17) a student of Department of Computer Science, Ramakrishna Mission Residential College(Autonomous), has undergone a Project work from January 15, 2020 to May 30, 2020 titled "**Image Steganography Using Different Methods**".

Siddhartha Banerjee

Department of Computer Science
Ramakrisnha Mission Residential College
(Autonomous)
Narendrapur, Kolkata-700 103

Dr. Siddhartha Banerjee
(Head of the Department)
Department of Computer Science

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE(AUTONOMOUS),NARENDRAPUR

Image Steganography Using Different Methods

Sukhomoy Debnath
CSUG/206/17

Aritra Raut
CSUG/205/17

Department of Computer Science

2020

Image Steganography Using Different Methods

Abstract:

This report contains 3 different methods of image steganography. Here we will show you 3 methods from 3 different ages of the steganographic world.

As the first method we chose PVD (Pixel Value Differencing). According to the human vision, eyes can tolerate more changes in edge block than in smooth area block so more data can be embedded into the sharp edges than into smooth area . This method takes the difference value of two consecutive pixels, and then embeds k-bits of the data onto their LSB bits according to the difference.

As our second method we chose TPVD (Tri Pixel Value Differencing) method. It is nearly same as the previous method but here we take a 2*2 block of an image and then following some algorithm we decide in which pixel pair we should embed the data. This makes the distribution of the data more random, making it harder for the hackers, who try to break the code and steal the data.

And at last we chose integer to integer S WT (S-Wavelet transformation) to hide the data. Here we will break the image in four parts, as it happens in wavelet transformation. And then embed the data in one of them and then reform the original image from those four components, and then send the data to the receiver through the medium.

Methods:

1.A PVD approach proposed by Cheng-Hsing Yang, Chi-Yao Weng, Shiuh-Jeng Wang, Member, IEEE, and Hung-Min Sun:

a.Introduction:

The internet is becoming increasingly popular as a communication channel. However, message sending via internet have some problems. There are some people who try to steal your message while it is in the medium . Nowadays with increasing usage of the internet, the number of such people is also increasing. So, now it is so important to secure your data while it is in the medium and to bring security there are two methods, one is to modify the data (in terminology it is known as cryptography) and another is to hide the data(in terminology it is known as steganography).Researchers of Greek history came to know that this data modification and hiding system is used since ancient Greece. So we can say that requirement of the hiding and modifying the data are quiet old.

A well known and the most basic steganographic method is the least significant bit (LSB) substitution. This method embeds secret data by replacing k LSBs of a pixel with k bits directly. However, not all pixels can tolerate an equal amount of change. As a result, many new sophisticated LSB approaches have been proposed to improve this drawback . Some of these methods use the concept of human vision to increase the quality of the stegoimages .

In 2003, Wu and Tsai proposed a "pixel-value differencing" steganographic method that uses the difference value between two neighbor pixels to determine how many secret bits should be embedded . Here we discuss **Adaptive Data Hiding in Edge Ares of Images with Spatial LSB Domain Systems,** proposed by prof. Yang, Prof Weng and Prof. Wang. In this method the maximum amount of data is hidden in the edge areas of a grayscale cover image.

b.Method:

Embedding :

- The cover images used are of 256 gray values.
- Difference value d is computed for every non overlapping block with two consecutive pixels.
- Here we take three ranges for difference value of pixels R1=[0-15], R2=[16-31] and R3=[32-255].
- In addition, to succeed in the readjusting phase we apply the restrictions $l<=\log_2|R1|$, $m<=\log_2|R2|$, $h<=\log_2|R3|$ to the *l-m-h* division, where |R1|, |R2|, |R3| are the cardinality of R1, R2, and R3 respectively.

For each block, the detailed embedding steps for an *l-m-h* division are as follows-

1. Calculate the difference value for each block with two consecutive pixels, say p_i and p_{i+1}, using $d_i = |p_i - p_{i+1}|$.

2. From the *l-m-h* division, find out the level to which d_i belongs to. Let , $k = l$, m, and h, if d_i belongs to the lower level, middle level, and higher level, respectively.
3. By the *k*-bit LSB substitution method, embed *k* secret bits into p_i and *k* secret bits into p_{i+1}, respectively. Let p'_i and p'_{i+1} be the embedded results of p_i and p_{i+1}, respectively.
4. Apply the modified LSB substitution method to p_i and p_{i+1}.
5. Calculate the new difference value d_i by $d'_i= |p'_i - p'_{i+1}|$.
6. If d_i and d_{i+1} belong to different levels, execute the readjusting phases as follows.

6.1. d_i belongs to lower-level, d'_i does not belongs to lower level. If $p_i >= p_{i+1}$, readjust (p'_i ,p'_{i+1}) to being the better choice between $(p'_i , p'_{i+1}+2^k)$ and (p'_i-2^k, p'_{i+1}) ; otherwise, readjust (p'_i ,p'_{i+1}) to be the better choice between $(p'_i , p'_{i+1}-2^k)$ and (p'_i+2^k, p'_{i+1}).

6.2. d_i belongs to middle-level, d'_i belongs to lower level. If $p_i >= p_{i+1}$, readjust (p'_i ,p'_{i+1}) to being the better choice between $(p'_i +2^k, p'_{i+1})$ and $(p'_i, p'_{i+1}-2^k)$; otherwise, readjust (p'_i ,p'_{i+1}) to be the better choice between $(p'_i -2^k, p'_{i+1})$ and $(p'_i, p'_{i+1}+2^k)$.

6.3. d_i belongs to middle-level, d'_i belongs to higher level. If $p_i >= p_{i+1}$, readjust (p'_i ,p'_{i+1}) to being the better choice between $(p'_i , p'_{i+1}+2^k)$ and (p'_i-2^k, p'_{i+1}) ; otherwise, readjust (p'_i ,p'_{i+1}) to be the better choice between $(p'_i , p'_{i+1}-2^k)$ and (p'_i+2^k, p'_{i+1}).

6.4. d_i belongs to higher-level, d'_i does not belongs to higher level. If $p_i >= p_{i+1}$, readjust (p'_i ,p'_{i+1}) to being the better choice between $(p'_i , p'_{i+1}-2^k)$ and (p'_i+2^k, p'_{i+1}) ; otherwise, readjust (p'_i ,p'_{i+1}) to be the better choice between $(p'_i , p'_{i+1}+2^k)$ and (p'_i-2^k, p'_{i+1}).

<In Step 6> the better choice, say (x_i, x_{i+1}), means that it satisfies the conditions that $|x_i-x_{i+1}|$ and d_i belong to the same level.

Extraction:

The stego image is partitioned into non overlapping blocks with two consecutive pixels, and the process of extracting the embedded message is the same as the embedding process with the same traversing order of blocks.

For each block, the detailed steps of data extracting are as follows-

1) Calculate the difference value d_i for each block with two consecutive pixels, say p'_i and p'_{i+1}, using $d'_i = |p'_i - p'_{i+1}|$.
2) From the *l*-*m*-*h* division, find out the level to which d'_i belongs to. Let , $k=l$, m, and h, if d'_i belongs to the lower level, middle level, and higher level, respectively.
3) From the k-bit LSB of a pixel, extract k secret bits from p'_i and k secret bits from p'_{i+1} .

c.Results :

we applied this data on 'Cameraman.png' and after embedding we see it is impossible for our human eye to guess that the pictures are different.

: original cover image :

: stego image :

Then we applied the same method on 5 different images and compared those with the help of some conventional parameters-

Image Name	MSE	RMSE	PSNR	SSIM
Macao.png	$4.3130e^{-08}$	5.3513	133.5617	0.9999
Monkey.png	$8.6697e^{-09}$	9.3111	128.7508	0.9999
Cameraman.png	$4.3678e^{-08}$	0.0002	121.7281	0.9999
Lenna.png	$7.3051e^{-09}$	8.5470	129.4946	0.9999
Baboon.png	$7.8871e^{-09}$	8.88095	129.1616	0.9999

2.Image steganography using Tri-Way Pixel Value Differencing by Ko-Chin Chang,Chien-Ping Chang,Ping S Huang,Te-Ming Tu,2008:

a.Introduction:

This method is quiet similar to the previous one. But here we will choose different blocks of an image to embed the data and then determine the pixel pairs in which we should embed. We know Human eyes can tolerate more change in the edge areas than to the smooth areas. So based on this we can embed more bits in the edge areas in order to increase the capacity.

b.Method:

Embedding:

Step-1 > Divide the gray valued cover image 2×2 blocks and then take the 2 consecutive pixels in horizontal , vertical and diagonal directions.

P(x,y)	P(x+1,y)
P(x,y+1)	P(x+1,y+1)

Here x and y denotes the pixel positions.

Let, P(x,y) is starting pixel then 3 pixel pairs can be found as
P0=(P(x,y),P(x+1,y))
P1=(P(x,y),P(x,y+1))
P2=(P(x,y),P(x+1,y+1))

$P3=(P(x,y+1),P(x+1,y+1))$

Step-2 > Calculate pixel value difference of each pair as

$d0=P(x+1,y)-P(x,y)$

$d1=P(x,y+1)-P(x,y)$

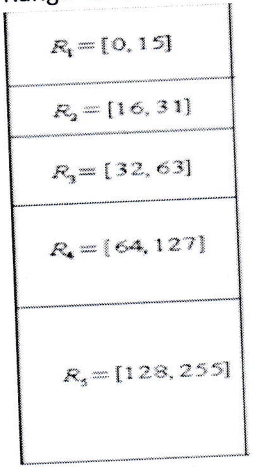
$d2=P(x+1,y+1)-P(x,y)$

$d3=P(x+1,y+1)-P(x,y+1)$

- The range of $|di|$ is in between 0 to 255.
- If $|di|<0$ then discard the pixel pairs.
- The small value of $|di|$ locates in the smooth area.
- The large value of $|di|$ locates in the sharp area.
- According to the human vision eyes can tolerate more changes in edge block than in smooth area block so more data can be embedded into the sharp edges than into smooth area .

Step-3 > Design a range table with R_k where $k=1,2,\ldots,n$.

- Range of the table is 0 to 255.



$R_1=[0,15]$
$R_2=[16,31]$
$R_3=[32,63]$
$R_4=[64,127]$
$R_5=[128,255]$

- The lower and upper boundary of R_k denoted by l_k and u_k.
- The width (w_k) of R_k is calculated as

$$W_k=u_k-l_k+1$$

- For each pair of pixel value difference ($|di|$) calculating the range as $j=\min(u_k-|d_i|)$ where $u_k \geq |d_i|$ for all $1 \leq k \leq n$.

Then R_j is the located range.

- Compute the amount of secret data bits(t) embedded into each pair of consecutive pixels by

$t=|\log_2 w_j|$ where w_j is width of R_j.

Step-4 >

After embedding t bits into consecutive pixel difference value, compute the new difference value.

- Read t bits from binary secret data and transform the bit sequence into a decimal value b.
- Calculate the new difference value di' as

$di'=lj+b$ if $di \geq 0$

Or $d_i' = -(l_j + b)$ if $d_i < 0$

Here lj is the lower limit of Rj.

Step-5 >

Computing the new pixel position as

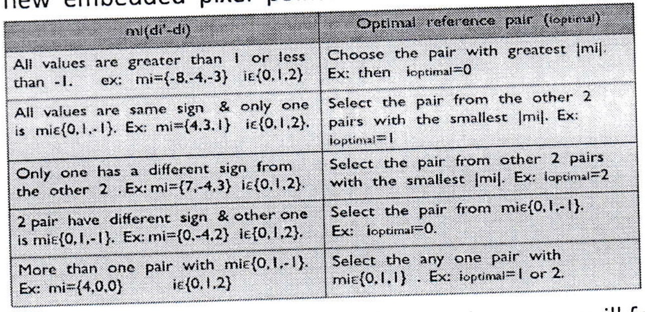
$P_i' = P_i - \lfloor (m/2) \rfloor$

$P_{i+1}' = P_{i+1} + \lfloor (m/2) \rfloor$

Here $m = d_i' - d_i$

The secret data is embedded into the pixel pair (Pi',Pi+1') is done by changing the values of Pi and Pi+1.

Step-6 > Here is an optimal selection approach to achieve minimum **Mean Square Error** of new embedded pixel points.



mi(di'-di)	Optimal reference pair (ioptimal)
All values are greater than 1 or less than -1. ex: mi={-8,-4,-3} i∈{0,1,2}	Choose the pair with greatest |mi|. Ex: then ioptimal=0
All values are same sign & only one is mi∈{0,1,-1}. Ex: mi={4,3,1} i∈{0,1,2}.	Select the pair from the other 2 pairs with the smallest |mi|. Ex: ioptimal=1
Only one has a different sign from the other 2 . Ex: mi={7,-4,3} i∈{0,1,2}.	Select the pair from other 2 pairs with the smallest |mi|. Ex: ioptimal=2
2 pair have different sign & other one is mi∈{0,1,-1}. Ex: mi={0,-4,2} i∈{0,1,2}.	Select the pair from mi∈{0,1,-1}. Ex: ioptimal=0.
More than one pair with mi∈{0,1,-1}. Ex: mi={4,0,0} i∈{0,1,2}	Select the any one pair with mi∈{0,1,1} . Ex: ioptimal=1 or 2.

Now to extract the data from this stego image we will follow some reverse steps as following,

Extraction:

Step-1 > Partition stego_image into 2×2 pixel blocks and then taking the 2 consecutive pixels in horizontal , vertical and diagonal directions.

Step-2 > Calculate the difference value di as

$d0 = P(x+1,y) - P(x,y)$

$d1 = P(x,y+1) - P(x,y)$

$d2 = P(x+1,y+1) - P(x,y)$

$d3 = P(x+1,y+1) - P(x,y+1)$

Discarded the pixel pairs whose $d_i < 0$.

Step-3 > For each pair of pixel value difference (|di|) calculating the range table with helping of step 3 of embedding phase as

$\mathbf{R_{K,i} = min(u_K - |d_i|)}$,where $u_K \geq |d_i|$ for all $1 \leq k \leq n$.

u_K is the upper limit of $R_{K,i}$.

- Compute the amount of embedding data bits(t) by

$\mathbf{T_i = \lfloor \log 2 w_{j,i} \rfloor}$,where $w_{j,i}$ is width of $R_{k,i}$.

Step-4 >

- Lower limit of $R_{j,i}$ $(l_{j,i})$ is subtracted from the selected $|d_i|$ and b_i is obtained as.

  $\mathbf{B_i = d_i - l_{i,j}}$

- Finally b_i is converted from decimal value into binary into a binary sequence with t_i bits.

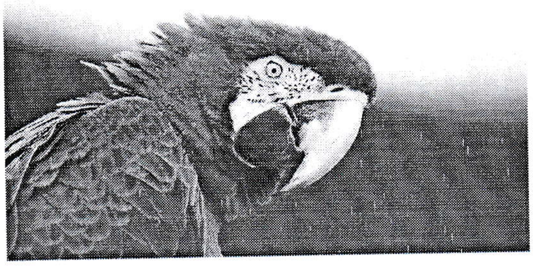
- The t_i bit stream is only one part of the secret data before embedding.

c.Result and Discussion:

We applied this method to 5 different images using the hash code(sha256 algorithm) of those respective images as the data to be encoded and have successfully retrieved the data back.

Image Name	MSE	RMSE	PSNR	SSIM
Macao.png	$1.2611e^{-09}$	$3.5511e^{-05}$	137.1233	0.9999
Monkey.png	$1.9746e^{-08}$	0.0001	125.1761	0.9999
Cameraman.png	$2.1648e^{-08}$	0.0001	124.7765	0.9999
Lenna.png	$7.3487e^{-09}$	$8.5724e^{-05}$	129.4687	0.9999
Baboon.png	$1.9743e^{-08}$	0.0002	125.1556	0.9999

We can clearly see that the secret data embedded in the stego_image is imperceptible for human vision while compared with the cover image.



: Cover Image(before embedding):

: Stego-Image(after embedding):

This method provides more robustness than the previous method,to avoid the data detection and thus it is a better approach than the previous one. Here we can make it more harder to detect the data by taking some specific pixel blocks from every two rows following the ap series.

3.Image steganography using Integer to Integer wavelet transformation:

a.Introduction:

Before starting any discussion , we have to know what wavelets are actually. Simply we can say , wavelets are mini waves. Rather than being a wave that goes on forever like sin and cos waves, wavelets are a short burst of wave that quickly die away.

Now, we know we transfer data from machine to machine as a form of signals. Now when these signals are in the medium, they can be noised in various ways and the receiver receives the signal not as sent from the sender side. Now to send the data properly we need to do some modulations in the signal. Wavelets are very useful tool in this field. Using wavelets we can modulate the signal and can send these. This is known as wavelet transformation.

Why it is used?

A signal is usually something you want to send or record. For example it could be a clip of voice record, image or video etc. Now we have to modulate this. Engineers find it useful to deal with a signal in frequency domain (frequency along x-axis and amplitude along the y-axis). Now to convert a signal from time domain to frequency domain, and do the modulation and then getting the modulated signal from the frequency domain to the time domain, we have a very useful tool called 'Fourier transformation'.

Now we all know about Heisenberg's uncertainty principle, which says that at any particular moment either you can determine the position of the molecule or the velocity of it. The same happens here, at any point of amplitude either you can determine the exact frequency of the signal (from frequency domain),or you can determine the exact time at which the amplitude occurred(from the time domain).

To overcome this resolution problem wavelet transformation is used to deconstruct the signal into a load of wavelets being added together. Wavelets are useful because they are limited in time and frequency. Instead of a wavelet lasting forever and having no limit in time, it dies quickly.

Here we discuss an integer wavelet transformation process. Basically it is an S-Transform using the idea of lifting scheme produced by Wim Sweldens . When we divide the picture in different components by applying some filtration on the pixel values we generally get some float values and then to plot them we convert them to integers by using round off operator. Now when we try to reform that Image from these components , we face some some minor error. That's why we are using this special operation to get integer values from those filtrations, which will remove the problem of errors and will help us to get the original image from those components.

b.Method:

A canonical case of lifting consists of three stages, which we refer to as: **split, predict,** and **update.** We here describe the basic idea behind each and later work out a concrete example. Assume we start with an abstract data set. We know this data set has some correlation structure and we would like to exploit it to obtain a more compact representation

Let us take an image pixel matrix as the data set-

191	187	206	198
171	151	186	186
130	106	116	168
112	120	136	140

Embedding:

At first we need to **split** the dataset into two parts(the re is no restriction in this splitting, thus it is known as lazy wavelets). Then we try to **predict** one part of the image from the second one(using any **prediction operator**).

Here we will apply high pass and low pass filter on every row of the image(size m*n) ,which will create two component of that image of size m*(n/2) . the low pass filter brings the approximation component of that image and the high pass filter brings the detail component of that image.

To do this we actually divide the image into pairs of pixels ,each pair made of one even and one odd pixel and then to bring the approximation component we **predict** the approximation pixel from two consecutive even and odd pixels(here we take the avg as prediction operator) and to bring the detail component we take difference(another prediction operator) of them. But in this prediction the average calculation will produce some fractions and then at the time of reformation it may predict the wrong pixel value, which will lead us to an error. This problem can be solved by the third stage of lifting scheme, which is known as **update.** In this stage we will update the values, which we get after prediction stage ,with another operator. The operations in details are shown below -

Step-1.1 >

Approx: a[n]=floor((x[2n]+x[2n+1])/2) , where floor(**updating operator**) is a function which gives the floor value of the

division

Detail: d[n]=x[2n]-x[2n+1]

Where ,x[n] is the pixel value of the image at position n and n=(length of each row)/2

For example we apply this on the above matrix-

X=[191,187,206,198]

Thus, for n=0,

a[0]=floor((191+187)/2)=189 ; d[0]=191-187

and for n=1,

a[1]=floor((206+198)/2) ; d[1]=206-198=8

doing this for every row we get the matrix-

189	202	4	8
161	186	20	0
118	142	24	-52
116	138	-8	-4

L H

Step-1.2 > now we apply the same operations as step 1.1 on every column of the matrix which we get after step-1.1 .

So, L produces submatrix LL and LH each of size (m/2)*(n/2)

And H produces submatrix HL and HH each of size (m/2)*(n/2)

After this operations the matrix looks like-

LL HL

175	194	12	4
117	140	8	-28
28	16	-16	8
2	4	32	-48

LH HH

We have now replaced the original data with wavelet representation. Given that the wavelet sets encode the difference with some predicted value based on a correlation model, this is likely to give a more compact representation.

Step-3 > now we take any of the components but LL, and then encode one bit the data at the LSBs

of each pixel values of that component (if the data length is less than the size of that component) .Suppose we take the binary data 101 and after encoding at each LSB th matrix will look like

175	194	13	4
117	140	9	-28
28	16	-16	8
2	4	32	-48

Here we encoded the data at the LH component.and after all this we will send this image through the medium.

Step-4 > now we have to reform the image from this components. For this reverse transformation we apply the undergiven mathematical formulae ,first column wise and then row wise-

x[2n]=a[n]+floor((d[n]+1)/2)

x[2n+1]=x[2n]-d[n]

look, here we are adding 1 to d[n] and then after dividing (d[n]+1) by 2 we are again taking the floor operator ,both as a part of **update** operation, in order to predict the original value.

After column operation-

189	202	5	8
161	186	21	0
118	142	25	-52
116	138	-7	-4

Now we apply same operation o each row of this matrix-

192	187	206	198
171	151	156	156
131	106	116	168
113	120	136	140

And now we will sed this image through the medium.

Extraction

Step-1 > we will do the same forward transformation on this image matrix in order to get the four components.

Step-2 > then we will take the LH component and extract the LSB bit from the pixels to get the desired data.

c.Result and discussuion:

There are some advantages of using this lifting scheme-

1. It allows a faster implementation of the wavelet transform. Traditionally, the fast wavelet transform is calculated with a two-band subband transform scheme. In each step the signal is split into a high pass and low pass band and then subsampled. Recursion occurs on the low pass band. The lifting scheme makes optimal use of similarities between the high and low pass lters to speed up the calculation. In some cases the number of operations can be reduced by a factor of two.

2. The lifting scheme allows a fully in-place calculation of the wavelet transform. In other words, no auxiliary memory is needed and the original signal (image) can be replaced with its wavelet transform.

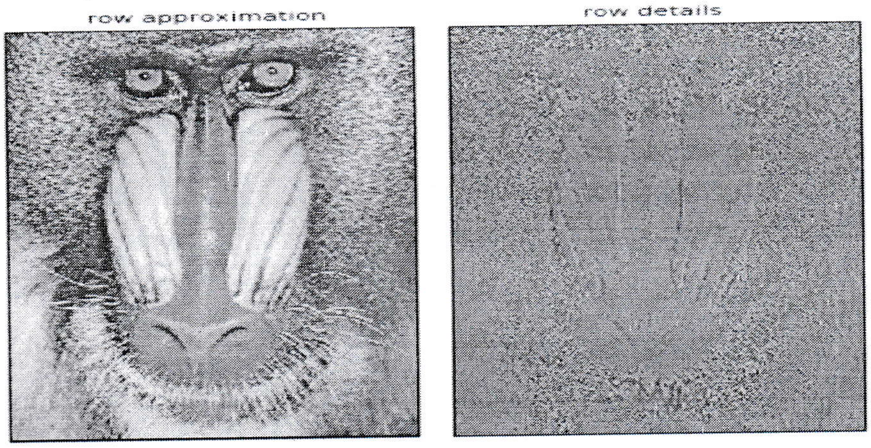
3. In the classical case, it is not immediately clear that the inverse wavelet transform actually is the inverse of the forward transform. Only with the Fourier transform one can convince oneself of the perfect reconstruction property. With the lifting scheme, the inverse wavelet transform can immediately be found by undoing the operations of the forward transform. In practise, this comes down to simply reversing the order of the operations and changing each + into a and vice versa.

4. The lifting scheme is a very natural way to introduce wavelets in a classroom. Indeed, since it does not rely on the Fourier transform, the properties of the wavelets and the wavelet transform do not appear as somehow “magical" to students who do not have a strong background in Fourier analysis.

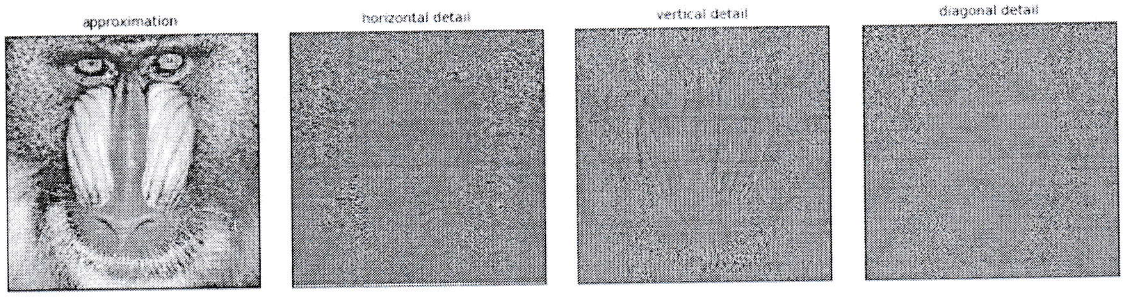
Now let us take a look on the results-

we take an image named ‘Baboon.png’ as the cover image-

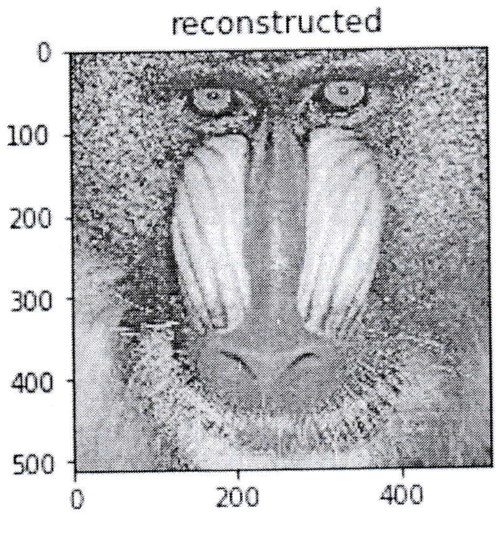
now we apllied forward transformation onto this, and after row operations we get-



and after column operation on this two components we get-



Then we embedded the hash code(sha256) of the original cover image in the LSBs of the LH(Horizontal detail)component and then reconstructed the image from this 4 components again-



Now finally we compared the original cover image and the stego image and the results are-

Image	MSE	RMSE	PSNR	SSIM
Macao.png	$2.8636e^{-09}$	5.3512	133.5617	0.9999
Monkey.png	$8.6697e^{-09}$	9.3111	128.7508	0.9999
Cameraman.png	$4.3678e^{-08}$	0.0002	121.7281	0.9999

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(O(i,j) - D(i,j))^2$$

- RMSE=Root Mean Square Error
- PSNR=Peak Signal Noise Ratio

$$PSNR = 10log_{10}(\frac{(L-1)^2}{MSE}) = 20log_{10}(\frac{L-1}{RMSE})$$

- SSIM=Structural Similarity Index Measure

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Conclusion:

After discussing these 3 methods briefly and applying them we cannot say that which one is better in every aspect. All of these 3 methods are good in some aspects and average in some aspects. And the performance depends on the images also, it varies according to the character of the image(no. of edge areas ,the brightness , the size of the image etc.).And using wavelet transformation on this field opened a new window to make our network channels more and more secure.

References:

- A steganographic method for images by pixel-value differencing by Da-Chun Wu , Wen-Hsiang Tsai,2003
- Fridrich, J., 1998. Image watermarking for tamper detection. In: Proc. IEEE Internat. Conf. on Image Processing, Vol. II, pp. 404–408.
- Anderson, R.J., Petitcolas, F.A.P., 1998. On the limits of steganography. IEEE J. Selected Areas Commun. 16, 474– 481.
- A novel image steganographic methodusing tri-way pixel value differencing by Ko-Chin Chang,Chien-Ping Chang,Ping S Huang,Te-Ming Tu,2008.
- Lifting scheme: a new philosophy in biorthogonal wavelet constructions by W Sweldens – Wavelet applications in signal and image, 1995

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE (AUTONOMOUS)
RAMAKRISHNA MISSION ASHRAMA
NARENDRAPUR, KOLKATA-700 103, W. B., INDIA
(A Branch Centre of Ramakrishna Mission, Belur Math, Howrah-711202)

Project Completion Certificate

Date: 03/06/2020

This is to certify that **Sri Hiranmoy Roy** (ROLL NO: CSUG/161/17) a student of Department of Computer Science. Ramakrishna Mission Residential College(Autonomous), has undergone a Project work from January 15, 2020 to May 30, 2020 titled "**Identification of Brain Tumor using Image Processing Techniques**".

Siddhartha Banerjee

Department of Computer Science
Ramakrisnha Mission Residential College
(Autonomous)
Narendrapur, Kolkata-700 103

Dr. Siddhartha Banerjee
(Head of the Department)
Department of Computer Science

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE (AUTONOMOUS)
RAMAKRISHNA MISSION ASHRAMA
NARENDRAPUR, KOLKATA-700 103, W. B., INDIA
(A Branch Centre of Ramakrishna Mission, Belur Math, Howrah-711202)

Project Completion Certificate

Date: 03/06/2020

This is to certify that **Sri Arunava Adhikari** (ROLL NO: CSUG/174/17) a student of Department of Computer Science, Ramakrishna Mission Residential College(Autonomous), has undergone a Project work from January 15, 2020 to May 30, 2020 titled "**Identification of Brain Tumor using Image Processing Techniques**".

Department of Computer Science
Ramakrishna Mission Residential College
(Autonomous)
Narendrapur, Kolkata-700 103

Dr. Siddhartha Banerjee
(Head of the Department)
Department of Computer Science

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE(AUTONOMOUS),NARENDRAPUR

Identification of Brain Tumor using Image Processing Techniques

NAME: ARUNAVA ADHIKARI

ROLL NO: CSUG/174/17

NAME : HIRANMOY ROY

ROLL NO : CSUG/161/17

Department of Computer Science

2020

Identification of Brain Tumor using Image Processing Techniques

Abstract –

At present, processing of medical images is a developing and important field. It includes many different types of imaging methods. Some of them are Computed Tomography scans (CT scans), Ultrasound, X-rays and Magnetic Resonance Imaging (MRI) etc. These technologies allow us to detect even the smallest defects in the human body. Abnormal growth of tissues in the brain which affect proper brain functions is considered as a brain tumor. The main goal of medical image processing is to identify accurate and meaningful information using images with the minimum error possible. MRI is mainly used as it offers better difference concern of various cancerous tissues of human body because of its high resolution and better quality images compared with other imaging technologies. MRI imaging plays an important role in brain tumor for analysis, diagnosis and treatment planning. It's helpful for doctors to determine the previous steps of brain tumor. Brain tumor identifications through MRI images are a challenging task because of the complex structure of the brain. MRI images can be processed and the brain tumor can be segmented. The segmentation, detection, and extraction of infected tumor area from magnetic resonance (MR) images are a primary concern but an exhausting and time-consuming process performed by radiologists or clinical experts, and their accuracy depends on their experience only. So, the use of computer aided automated brain tumour segmentation and analysis technology has thus gained much attention in recent years. However, the existing segmentation techniques do not meet the requirements of real-time use due to limitations posed by poor image quality and image complexity. Nowadays, these tumors can be segmented using various image segmentation techniques. The process of identifying brain tumors through MRI images can be categorized into four different sections; pre-processing, image segmentation, feature extraction and image classification.

Keywords –

Brain Tumor (BT), Image Processing (IP), MRI-Images, CT, X-Ray, Pre-Processing, Segmentation, Feature Extraction

I. INTRODUCTION –

Human body is made up of several types of cells. Brain is a highly specialized and sensitive organ of human body. Brain tumor is a very harmful disease for human beings. It is an intracranial mass made up by abnormal growth of tissues in or around the brain. It can disrupt proper brain functions and be life-threatening. Two types of brain tumors have been identified as Benign (non-cancerous) tumors and Malignant (cancerous) tumors. Benign tumors look relatively normal, grow slowly, and do not spread to other sides of the body. These tumors can still be serious and even life threatening if they are in the vital areas of the brain. Malignant tumors can be further divided into two categories: primary and secondary tumors. Primary tumors start within the brain, and secondary tumors spread from some other parts of the body to the brain.

According to the World Health Organizations, brain tumors can be classified into grade I–IV. Generally, grades I and II are treated as low-grade tumors (benign), and grades III and IV as high-grade tumors (malignant). If a low-grade tumor is not treated properly, it is likely to develop into high-grade tumor, i.e. malignant tumor. Brain tumor diagnosis is quite difficult because of diverse shape, size, characteristics, location and appearance of tumor in brain. Detection at the beginning stage is very hard because it can't find the accurate measurement of tumor. Once the brain tumor is clinically suspected, radiologic assessment (manual operation) is required to decide the position, area, and extent of the tumor, as well as its relationship with the neighbouring structures. This information is very essential and crucial in the planning of further treatment.

Medical imaging techniques are used to create visual representation of interior of the human body for medical purposes. The various types of medical imaging technologies based on non-invasive approach like: MRI, CT scan, Ultrasound, SPECT, PET and X-ray. Among these MRI provides greater contrast images of the brain and cancerous tissues. In MRI-scan there is a powerful magnetic field's component to determine the radio frequency pulses and to produce the detailed pictures of organs, soft tissues, bones and other internal structures of the human body. So, the MRI-Technique is the most effective for brain tumor detection. This paper focuses on the identification of brain tumor using image processing techniques.

In image processing various tools are used to improve the quality of images. The contrast adjustment and threshold techniques are used for highlighting the features of MRI images. The Edge detection, Histogram, Segmentation and Morphological operations play a vital role for classification and detecting the tumor of brain. The main objective of this paper is too studied and reviewed the different research papers to find the various filters and segmentation techniques, algorithms to brain tumor detection.

II. *BACKGROUND*

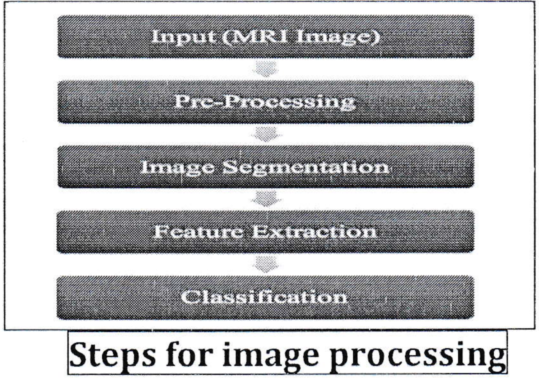
Brain Tumor is described as abnormal development of tissues in the brain. Nowadays the prevalence of tumors is growing fast. In 2016, an estimated 23,800 adults (10,350 women and 13,450 men) in the US were identified with the harmful tumors of brain as well as spinal cord. Analysis of brain tumors is somewhat problematic as the varied shape, size, tumor location and the presence and appearance of tumor in brain. It's hard to detect brain tumors in beginning stage because the accurate measurement of tumor can't be found. But once the brain tumor is identified at the very beginning, the proper treatments can be done and it may be curable. At present, visual representation of the interior of the body is processed using medical imaging technique for clinical analysis and medical researches. MRI is the most effective and extensively used technique for brain tumor detection. Current diagnosis techniques are performed using the conventional methods based on human experience and this increases the possibility of false detection when identifying brain tumors. Present tools and methods to analyse tumors and their behaviour have become more prevalent. Image processing technique can be used to identify brain tumors. Image processing methods converts images into digital and do operations on them, in order to get better and enhanced images. This study will focus how to identify brain tumors using image processing techniques.

III. *RELATED WORK*

In recent years, image processing has applied to process images in medical stream, in cooperating cell detection. In 2012, 'S. Mokhled' introduced several identification steps, including segmenting images to extract the object from the background through the threshold. This feature was introduced with the 'Gabor filter' in order to do more classification into cancer cells. In 2013, 'H. G. Zadeh' proposed further steps, which is image extraction and segmentation of images for diagnosing cancer cells. The Gaussian smoothing concept was introduced as a filtering purpose, previous to applying the 'Fast Fourier Transform' (FFT). Machine Learning for tumor detection: 'NN', 'Fuzzy C-mean' algorithms was introduced for the identification of tumorous cells. This takes lower computational time but the accuracy is also lower. In 2014, 'X. Chen' introduces gene counting technology. But this technology is appropriate only for the complex formation of gene selection. From the above-mentioned techniques and using other technologies, in this research paper we focus on the identification of brain tumor using image processing techniques.

IV. *METHODOLOGY –*

According to the following steps, Brain tumors can be detected using Image Processing techniques.



Steps for image processing

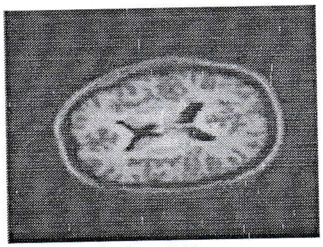
4.1. *Image Pre-Processing*

It is very difficult to process an image. Before any image is processed, it is very significant to remove unnecessary items it may hold. After removing unnecessary artifacts, the image can be processed successfully. The initial step of image processing is Image Pre-Processing. The objective of the pre-processing stage is to enhance the quality of brain images and make them suitable for further processing by human or machine vision system. This stage consists of two sub-stages, i.e. noise removal and skull stripping.

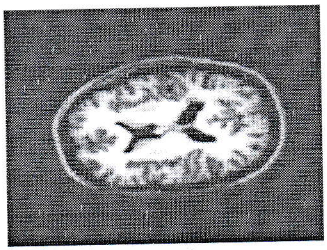
Pre-Processing involves processes like conversion to greyscale image, noise removal and image reconstruction. Conversion to greyscale image is the most common pre-processing practice. After the image is converted to greyscale, then remove excess noise using different filtering methods.

Image Enhancement

Poor contrast is one of the defects found in acquired image. The effect of that defect has great impact on the contrast of image. When contrast is poor the contrast enhancement method plays an important role. In this case the gray level of each pixel is scaled to improve the contrast. Contrast enhancements improve the visualization of the MRI images. Contrast enhancement technique is used to enhance the MRI image as shown in the figure.



Original Image



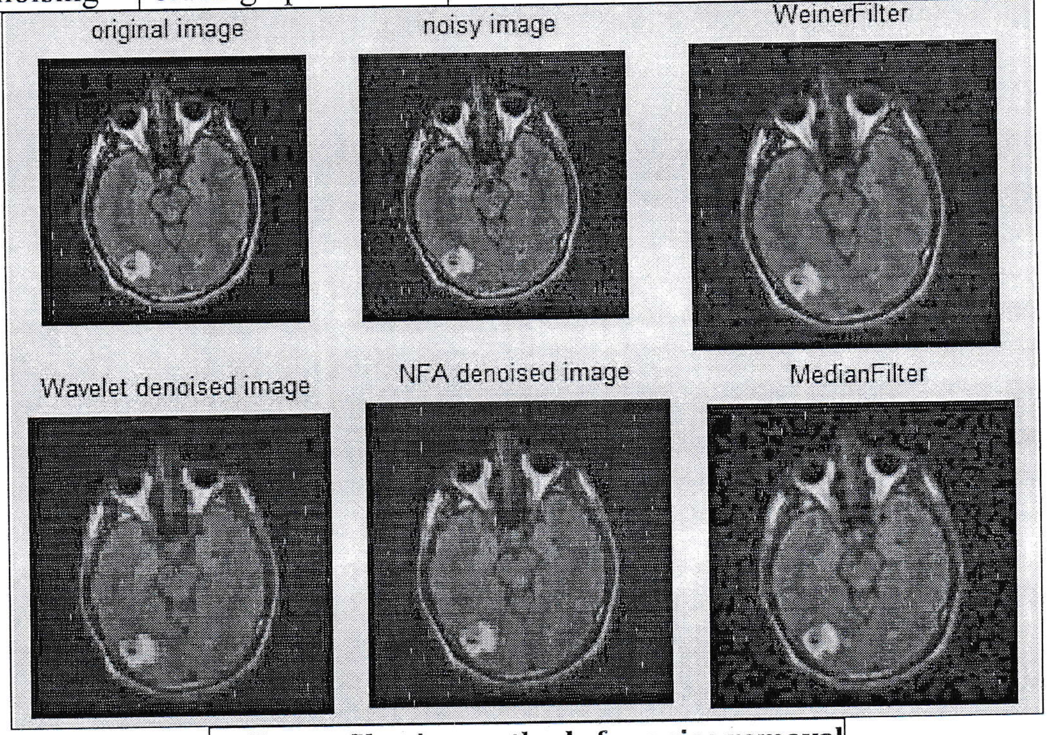
Enhanced Image

4.1.1. *Noise Removal*

The brain image contains impulsive noise and is to be reduced before segmentation to improve its quality. Most of the impulse noise-diminishing filters blur the image boundaries. Thus, an appropriate filter that reduces the impulsive noise and at the same time preserves edges and sharp details of the image should be selected. We have studied the various filtering techniques in digital image processing that are shown in table.

Various Filters	Working Principle	Advantages	Disadvantages
1. Median Filter	Based on the average value of pixels. It replaces the value of the centre pixel with the median of the intensity values in the neighbourhood of that pixel.	Efficient for reducing salt & pepper noise, speckle noise. Boundaries and edges are preserved.	Complex and time consuming as compared to mean filter.
2. Mean Filter	Based on average value of pixels	Reduces Gaussian noise. Response time is fast	Results with distorted boundaries and edges
3. Wiener Filter	Based on inverse filtering in frequency domain	Efficient for removing blurring effects from images	Due to working in frequency domain, its speed is slow. Not suitable for speckle noise.
4. Hybrid Filter	Combination of median and wiener filter	Removes speckle noise, impulse noise and blurring effects From images	Complex and time consuming
5.Modified hybrid median Filter	Combination of mean and median Filter	Efficient for removing speckle, salt and pepper and Gaussian noise	Computation time is more as compared to simple median filter

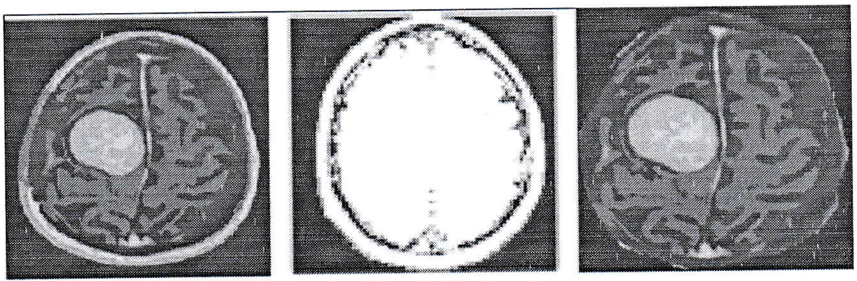
6. Morphology Based De-noising	Based on Morphological opening and closing operations	Efficient and produces better results as compared to other filters	



Different filtering methods for noise removal

4.2.2. *Skull Stripping*

Skull stripping is an important preliminary step in brain imaging where non-cerebral tissues like the skull, scalp and vein are to be removed. This is due to the fact that non-cerebral tissue does not usually contain any useful information but increases the execution time of detection. Therefore, to remove the skull from brain images, a threshold-based method is adopted in this study. It involves three steps. In the first step, the original medical image is processed and converted into grey scale equivalent as shown in Figure A, and then into a binary equivalent image as illustrated in Figure B. In the second step, the threshold value is set and the input binary image is processed by retaining the fixed threshold level and discarding other pixel values. This value is called mask. Finally, in the third stage, the skull-stripped image is obtained by multiplying the mask value with the input image, as shown in Figure 3C. The output of the third stage is a noise-free image that contains only the human brain.



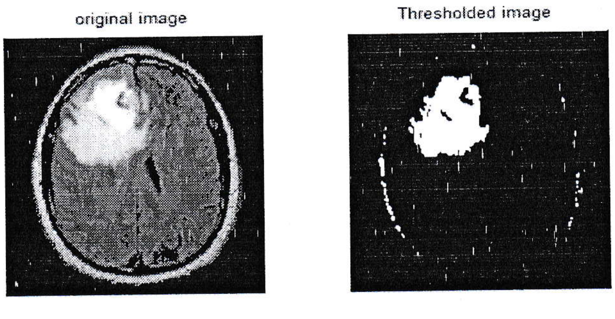
Brain Skull Stripping.
(A) Real grey brain image, (B) binary image, and (C) skull-stripped image.

4.2. *Image Segmentation*

'Image Segmentation' is the technique that has been introduced to divide a digital image into minor segments. It creates several sets of pixels and set of super pixels within same image. Objectives to be accomplished through the process of segmentation are simplifying and changing the format of representation of a digital image in a way that it will become more detailed, meaningful and easier to further analyse and recognize important information. Locating objects and boundaries in images such as lines, curves can be performed through Image segmentation. Throughout the procedure of image segmentation, every pixel in an image is assigned a label and the pixels consisting of similar label share certain visual features. Each pixel in the region is similar in relation to some features or computed properties, such as colour, intensity or texture. Adjoining regions are particularly different in regard to the same features. Here we will discuss different types of segmentation techniques.

A. *Threshold Segmentation*

Thresholding methodology is a simple, effective, way of partitioning an image into a foreground and background. This image analysis technique is a type of image segmentation that involves a threshold value that is used for converting a gray-scale featured image to a binary image. It uses gradient magnitude to find the potential edge pixels. Image threshold is most effective in images with high levels of contrast and is hard to be used for images with poor contrast. The major advantage of this method is selecting the threshold value to be used. Threshold technique is applied on the input MRI image by changing the threshold value and the result is shown in figure.



B. *Morphological Based Segmentation*

'Morphology' refers to describing the properties of the shape and structure of any entity. Binary images may comprise many defects. Particularly, the binary regions constructed by simple thresholding are deformed by texture and noise. Morphological image processing seeks to achieve the goals of eliminating these defects by accounting for image shape and structure. Generally, this denotes recognizing objects or boundaries within the image. Morphological operations are logical conversions based on comparison of pixel neighbourhoods with a pattern. It correctly separates regions according to the similarity of properties. Usually, morphological operations are implemented on binary images under the pixel values; 0 or 1. Many of the morphological operations target on binary images. Noise may lead to quality of final result.

C. *K-Means and Fuzzy C-Means Clustering Algorithm*

C.1. *K-Means Clustering algorithm*

The *K*-means clustering algorithm is a very simple unsupervised learning algorithm. It provides a very easy way to classify a given data set into a certain number of clusters i.e. a set of data such as $x_1, x_2, x_3, ..., x_n$ are grouped into *K* clusters. The major idea behind this algorithm is to define *K* centers, one for each cluster [39]. The *K* cluster centers should be selected randomly. Distance measure plays a very important rule on the performance of this algorithm. Different distance measure techniques are available for this algorithm such as Euclidean distance, Manhattan distance and Chebychev distance etc. But, choosing a proper technique for distance calculation is entirely dependent on the type of the data that we are going to cluster. However, we will use Euclidean distance as the distance metric because it is fast, robust and easier to understand [40,41]. Step by step conventional *K*-means clustering algorithm [40] is described as follows:

Algorithm 1 *K*-means clustering algorithm.

```
Assume that, $X = x_1, x_2, x_3, ...., x_n$ be the set of data points and $V = v_1, v_2, v_3, ..., v_c$ be the set of centers.
1: Define number of clusters $'K'$.
2: Randomly, define cluster centers $'c'$.
3: Calculate the distance between each data point and cluster centers.
4: Data point is assigned to the cluster center whose distance from the cluster center is minimum of
   all the cluster centers.
5: Then , new cluster center is recalculated as follows.

       $V_i = \frac{1}{c_i} \sum_{j=1}^{c_i} x_i$        (1)

   where $'c_i'$ is the number of data points in $i$-th cluster.
6: Recalculate the distance between each data point and newly acquired cluster centers.
7: If no data point was reassigned then stop, otherwise repeat steps from 3 to 6.
```

The distance which is called Euclidean distance are calculated between each pixel to each cluster centers. All the pixels are compared individually to all cluster centers using the distance function [42]. The pixel is lead to one of the clusters which is shorter in distance among all. Then, the center is recalculated. Then, every pixel is compared to all centroids again. This process continuous until the center converges and the convergence is evaluated through a maximum number of iterations. The quality of clustering of this algorithm is optimized through repetition of K-means several several times with different initialization in order to identify best centroids.

It provides improved computational efficiency and supports multidimensional vectors [43]. So, this algorithm intent to diminish an objective function which is known as squared error function given by:

$$J_v = \sum_{i=1}^{c} \sum_{j=1}^{c_i} (||x_i - v_j||)^2 \tag{2}$$

where $||x_i - v_j||$ is the Euclidean distance between x_i and v_j, c_i is the number of data points in ith cluster and c is the number of cluster centers.

C.2. *Fuzzy C-Means Clustering Algorithm*

FCM clustering algorithm is introduced by Bezdek that is a technique of clustering where proceeds each pixel of data to belong to two or more clusters. The more the data is near to the cluster center more is its membership towards the particular cluster center [44,45]. It depends on reducing an objective function regarding to fuzzy membership set U of cluster centroids V.

$$J_m(U,V) = \sum_{j=1}^{N} \sum_{i=1}^{C} (u_{ij})^m (||x_j - v_i||)^2; \quad 1 \leq m \leq \infty \tag{3}$$

where $X = x_1, x_2, ..., x_j, ..., x_n$ is a $P \times N$ data matrix in Equation (3) and m is any real number greater than 1. P, N and C denotes the dimension of each x_j 'feature' vectors, the number of feature vectors (pixel numbers in the image) and the number of clusters, respectively.

$U_{ij} \subseteq U(P \times N \times C)$ is called the membership function of vector x_j to the i-th cluster, which satisfies $U_{ij} \;\epsilon\; [0 \quad 1]$ and $\sum_{U_{ij}} = 1$, $j = 1, 2, ..., N$. The membership function can be expressed as follows:

$$U_{ij} = \sum_{k=1}^{c} \left(\frac{(||x_j - v_i||)}{(||x_j - v_k||)} \right)^{\left(\frac{-2}{m-1}\right)} \tag{4}$$

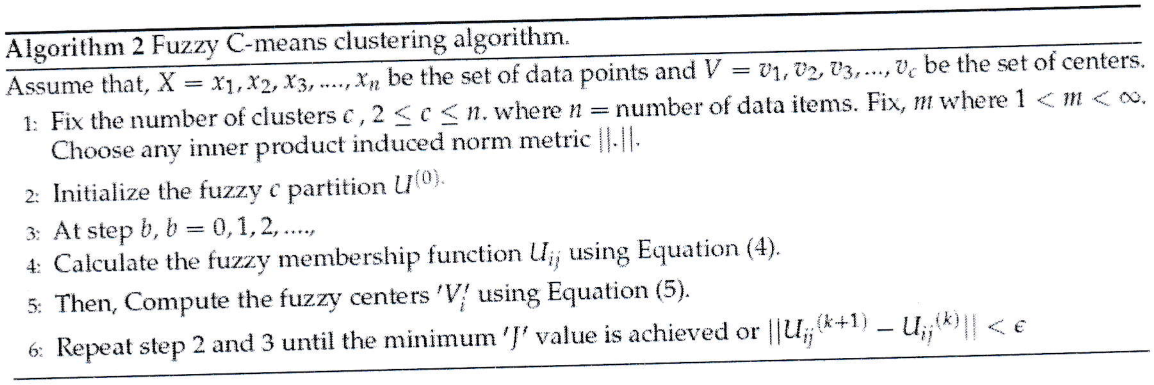
where $V = v_1, v_2, v_3, ..., v_i, ..., v_c$ which is a $P \times C$ matrix. Now we calculate the i-th cluster feature center as follows:

$$V_i = \frac{\sum_{j=1}^{N} (U_{ij})^m \times J}{\sum_{j=1}^{N} U_{ij}{}^m} \tag{5}$$

where m is any real number that is greater than 1, controls the degree of fuzziness $d^2(x_j, v_i)$. It is a measurement of similarity between x_j and v_i that is defined as follows:

$$d^2(x_j, v_i) = ||x_j - V_i||^2 \tag{6}$$

Here, $||\,.\,||$ can be denoted as either a straightforward Euclidean distance or its generalization like Mahalanobis distance. The feature vector X in MR image presents the pixel intensity $P = k$. The FCM algorithm repetitively optimizes $J_m(U, V)$ with the continuous update of U and V, until $||(U_{ij})^{(k)} - (U_{ij})^{(k+1)}|| \leq \epsilon$, ϵ 0 to 1, where k is the number of iterations. The step by step conventional fuzzy C-means clustering algorithm is demonstrated as follows:



Algorithm 2 Fuzzy C-means clustering algorithm.

```
Assume that, $X = x_1, x_2, x_3, ...., x_n$ be the set of data points and $V = v_1, v_2, v_3, ..., v_c$ be the set of centers.
1: Fix the number of clusters $c$, $2 \leq c \leq n$. where $n =$ number of data items. Fix, $m$ where $1 < m < \infty$.
   Choose any inner product induced norm metric $||.||$.
2: Initialize the fuzzy $c$ partition $U^{(0)}$.
3: At step $b$, $b = 0, 1, 2, ....,$
4: Calculate the fuzzy membership function $U_{ij}$ using Equation (4).
5: Then, Compute the fuzzy centers $'V_i'$ using Equation (5).
6: Repeat step 2 and 3 until the minimum $'J'$ value is achieved or $||U_{ij}{}^{(k+1)} - U_{ij}{}^{(k)}|| < \epsilon$
```

4.3. *Feature Extraction*

Accurate tumor extraction is a critical task in the case of brain tumor due to the complex structure of the brain. There are some criterions that are being considered to extract features. Feature extraction is the process of collecting higher level information of an image such as form (shape), size, texture, colour and contrast. In fact, texture analysis is an important parameter of human visual perception and machine learning system. It is used effectively to improve the accuracy of diagnosis system by selecting prominent features. Haralick et al. introduced one of the most widely used image analysis applications of Gray level Co-occurrence Matrix (GLCM) and texture feature. This technique follows two steps for feature extraction from the medical images. In the first step, the GLCM is computed, and in the other step, the texture features based on the GLCM are calculated. Due to the intricate structure of diversified tissues such as WM, GM, and CSF in the brain MR images, extraction of relevant features is an essential task. Textural findings and analysis could improve the diagnosis, different stages of the tumor (tumor staging), and therapy response assessment. With respect to the results retrieved from extract features the process of tumor classification is performed.

A. *Edge Detection*

An edge happens when there is a sudden and unexpected intensity modification of the image. Whenever it is detected an abrupt modification or a change in the intensity of a certain image, the associated pixel would be treated as an edge pixel. Edge detection is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness. Edge detection is used for image segmentation and data extraction in areas such as image processing, computer vision, and machine vision. The algorithm that has been put forward for the detection of edge pixel supports in identifying the quality of the edge. Common edge detection algorithms include methods like Prewitt, Sobel, Canny, Log, and Zero cross. Edge detection methods are used for finding object boundaries from MRI images. But sometimes these edges are not displayed in the final result. Hence the algorithms are adjusted to determine the edges.

1. *"Prewitt" edge detection*

The "Prewitt Mask" is considered as a distinct differentiation operation. Accordingly, approximated derivative values in both the directions, such that horizontal and vertical, are calculated using two 3 × 3 masks. Prewitt masks approximates both horizontal derivative and the vertical derivative.

2. *"Robert edge" detection*

Through the "Robert edge" detection operation, the image gradient is estimated via distinct differentiation. In addition, "Robert Mask" is a matrix and the regions of high spatial frequency are highlighted, that often correspond to edges in the image.

3. *"Sobel edge" detection*

The "Sobel Mask" is mostly worked as the "Prewitt mask". It can only be distinguished as the Sobel operator has values: '2' and '-2' which are allocated in the centre of 1st and the 3rd columns of the horizontal mask and 1st and 3rd rows of the vertical mask. Hence it provides high edge intensity.

4. *Canny Edge Detection*

Canny edge detection is used to obtain a high-contrast binary mask of segmented image. Canny edge detection works on the principle of calculating a gradient value of contrast between the segmented image and the background image.

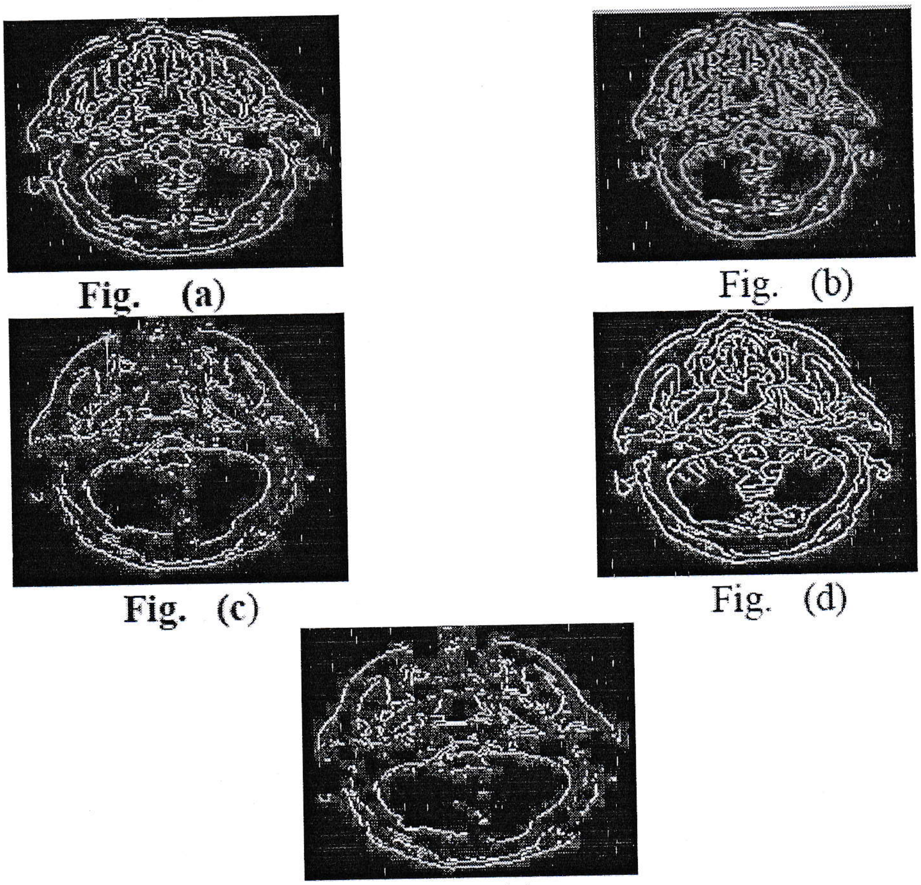


Fig. (a)

Fig. (b)

Fig. (c)

Fig. (d)

Fig. (e)

Fig. (a) Log operator (b) Sobel operator (c) Canny operator (d) Prewitt operator (e) Zero cross operator

B. *"Histogram of Oriented Gradient" Feature Extraction*

The extraction process of the "Histogram of Oriented Gradient" (HOG) is having following calculations. First, the pre-processed cell image will be distributed into "32 × 32" pixels. The intensity of each pixel is '0' or '1'. Then the result will be added to "HOG". Following figure shows the architecture of "HOG" feature. Then the image will be distributed into "8 × 8" pixels that is called box. Here, the box will be already added into a single block. Again each box will be distributed into 9 bins which is "3 × 3". Pixel gradient is used for the creation of the feature in each and every bin. Therefore there are 9 features, it will lead to "9 × 4" characteristics for each block. In the all "32 × 32" pixels, "HOG" feature extraction allows to create '9 blocks' and finally, it will having "9 × 9 × 4" features in single dimension or "1 × 324" in the vector image.

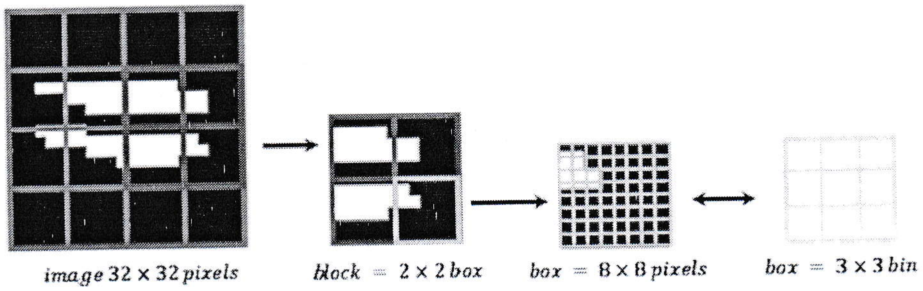


Figure : Architecture of HOG features (elongated cell)

Histogram

Histogram is nothing but the graphical representation of an image. The histogram of a digital image with gray levels in the range [0, L-1] is a discrete function. The histogram of an image mostly represents the comparative frequency of the various gray levels in the image. Histogram techniques is applied on input MRI image and result is shown in figure.

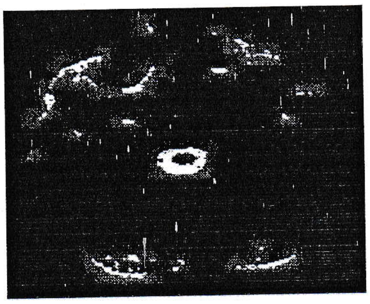


Fig.(a). Histogram applied image

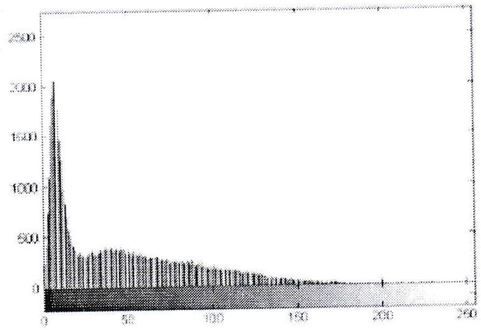


Fig.(b). Histogram of Fig.(a)

V. *RESULTS AND DISCUSSION*

The most significant thing in image processing is image segmentation, while diagnosing brain tumor from a digital image. Following table represents different segmentation methods with different characteristics such as accuracy and complexity.

Segmentation Method	Complexity (algorithm)	Accuracy (%)
Seed Region Growing	10	92.5
Threshold Segmentation	8.22	91
Watershed	5.67	88.5
Fuzzy C-Mean	5.29	85
Histogram Thresholding	7.61	81

TABLE: DIFFERENT TUMOR SEGMENTATION METHOD

Main goal of Pre-Processing is the edge preservation of the image. Among the edge detection mechanisms, Sobel is the best option, then both the Gaussian and the Median filter. The different edge detection algorithms like 'Sobel', 'Prewitt' and 'Robert' are shown in the following table with different characteristics such as computation time in seconds and computation number of flip-flops used.

	Sobel	Robert	Prewitt
Advantages	Simplicity	Better noise suppression	Mask simpler as compared to Sobel
Disadvantages	Discontinuity in edges	Not accurate results	Discontinuity in edges
Computation time in sec.	0.3	0.2	0.4
Number used as Flip Flops	343	219	339
Number used as logic	450	322	450

TABLE: PERFORMANCE OF EDGE DETECTION

The following figures show the steps of brain tumor detection using image processing techniques. That is original MRI image, grayscale image, filtering image using Median filter, segmenting using threshold method, morphological operation applied image and finally diagnosed tumor from MRI image.

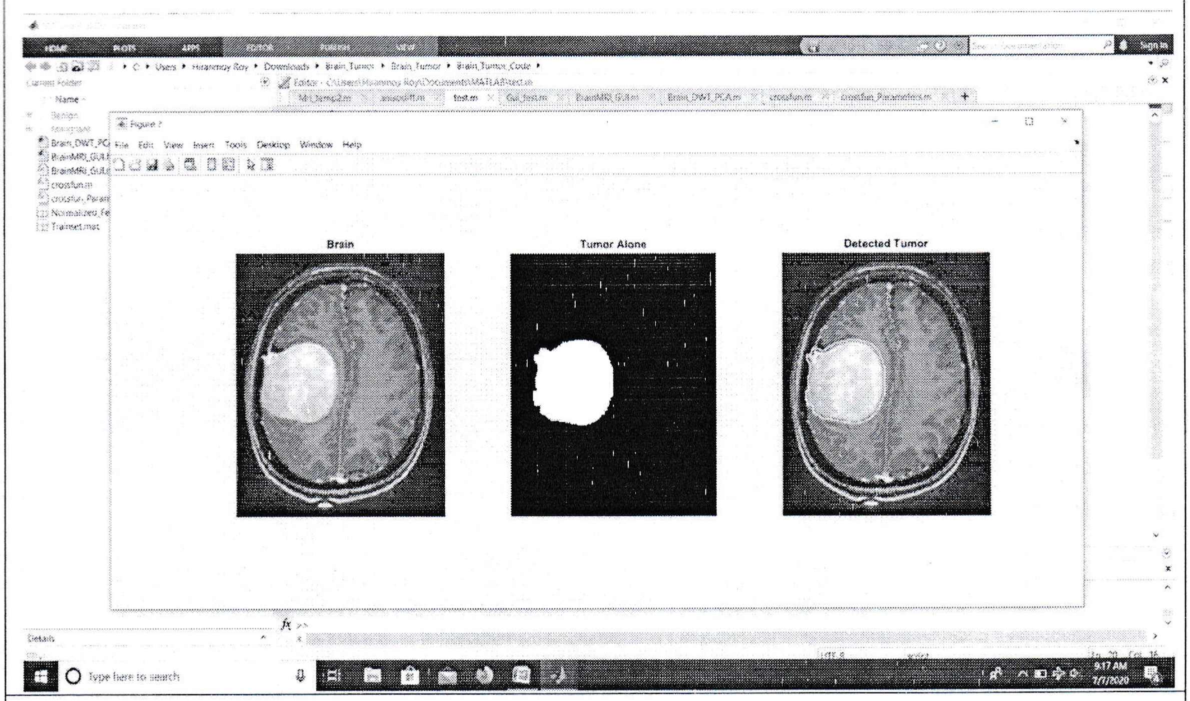


Figure: Steps of brain tumor detection

VI. *CONCLUSION*

Abnormal growth of tissues in the brain which affect proper brain functions is considered as a brain tumor. The main goal of medical image processing is to identify accurate and meaningful information using images with the minimum error possible. Brain tumor identifications through MRI images is a difficult task because of the complexity of the brain. These tumors can be segmented using various image segmentation techniques. The process of identifying brain tumors through MRI images can be categorized into four different sections; pre-processing, image segmentation, feature extraction and image classification. Median filter is the most commonly used filtering technique among various filtering techniques. Less complexity and the efficiency in eliminating 'Salt and Pepper noise' are the main advantages of median filter. Not like Gaussian filter, it is a non-linear filter, Median filter is an edge preserving filter. Also, Gaussian filter is a low pass filter hence the edge information will be lost and edges getting displaced and blurred. Although, less complexity and the cheapness to implement than the Median filter are the main advantages of Gaussian filter. Another advantage is the Gaussian filter is very applicable in smoothening Gaussian noise. Thresholding is the best and easiest approach among image segmentation techniques. It easy to implement and widely used these days. When the contrast between foreground object and background object is comparatively high, threshold technique works well.

VII.ACKNOWLEDGEMENT

I would like to take this great opportunity to thank **Dr. Siddhartha Banerjee**, **Shri Bibek Ranjan Ghosh** and **Shri Surajit Giri** for their advices and valuable guidance during my independent study. I should also thank the Department of Computer Science for giving me the opportunity and also thank all those who gave me the valuable advices that allowed me to make this independent study a success.

VIII.REFERENCES

Vipin Y. Borole, Sunil S. Nimbhore, Dr. Seema S. Kawthekar ,"Image Processing Techniques for Brain Tumor Detection: A Review" , Volume 4, Issue 5(2), ISSN 2278-6856 , September - October 2015.

Deepak C. Dhanwani, Mahip M. Bartere, "Survey on various techniques of brain tumour detection from MRI images", IJCER, Vol.04, issue.1, Issn 22503005, January 2014, pg. 24-26.

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE (AUTONOMOUS)
RAMAKRISHNA MISSION ASHRAMA
NARENDRAPUR, KOLKATA-700 103, W. B., INDIA
(A Branch Centre of Ramakrishna Mission, Belur Math, Howrah-711202)

Project Completion Certificate

Date: 03/06/2020

This is to certify that **Sri Aritra Mazumdar** (ROLL NO: CSUG/172/17) a student of Department of Computer Science, Ramakrishna Mission Residential College(Autonomous), has undergone a Project work from January 15, 2020 to May 30, 2020 titled "**Title Prediction of Essays Using NLP**".

Department of Computer Science
Ramakrishna Mission Residential College
(Autonomous)
Narendrapur, Kolkata-700 103

Dr. Siddhartha Banerjee
(Head of the Department)
Department of Computer Science

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE (AUTONOMOUS)
RAMAKRISHNA MISSION ASHRAMA
NARENDRAPUR, KOLKATA-700 103, W. B., INDIA
(A Branch Centre of Ramakrishna Mission, Belur Math, Howrah-711202)

Project Completion Certificate

Date: 03/06/2020

This is to certify that **Sri Ayush Chakraborty** (ROLL NO: CSUG/223/17) a student of Department of Computer Science, Ramakrishna Mission Residential College(Autonomous), has undergone a Project work from January 15, 2020 to May 30, 2020 titled **"Title Prediction of Essays Using NLP".**

Department of Computer Science
Ramakrishna Mission Residential College
(Autonomous)
Narendrapur, Kolkata-700 103

Dr. Siddhartha Banerjee
(Head of the Department)
Department of Computer Science

RAMAKRISHNA MISSION RESIDENTIAL
COLLEGE(AUTONOMOUS),NARENDRAPUR

Title Prediction of Essays Using NLP

Ayush Chakraborty
Roll. No.: CSUG/223/17

Aritra Mazumdar
Roll. No.: CSUG/172/17

Department of Computer Science

2020

INTRODUCTION

Have you ever wrote an essay your teacher gave as homework or maybe after writing a brilliant blog on internet and then thought hard on what should be the best title that would match your work? As per the current statistics, people get better response when they use a good title. The title of your manuscript is usually the first introduction readers have to your published work. Therefore, you must select a title that grabs attention, accurately describes the contents of your manuscript, and makes people want to read further. Therefore increasing the viewership of the essay and also leading to more traffic and organic search on your website or webpage. We often find ourselves confused while searching for a perfect title that would justify our content fully so that the write-up gets the attention it deserves. *This project aims at helping the people in search of a perfect title. The project aims at predicting a good and relevant title for your write-up.* We have used Natural Language Processing [NLP] in Python3 to implement the project. We are also aiming at convenient and easy interaction of the program with the users. The users have to submit the manuscript or write-up and the program will try to predict the most relevant titles that will justify the write-up.

Task Definition

The project aims at returning a good and relevant title for a given write-up. Here, the project takes an essay or a piece of write up as input. It then continues to works on what the write-up is about, picking out the important words from the document and organising them by rating them as per their importance in the document. Thus we get a set of words which can be used for forming a perfect title for the submitted write-up. We are using Python 3 for the project and the concepts of Natural Language Processing to work and breakdown the text data. The basic idea is to search for features which can model the attributes like vocabulary, structure, content etc. We have approached the problem without neural network and have achieved better results in terms of prediction.

FEATURE EXTRACTION

As we know, for any text based projects feature extraction is very important and helps in forming the basic structure of the given data into a simple ready to use form. We have used NLTK[Natural Language ToolKit] package in Python3 for feature extraction.

At present, our model is using the following set of features extracted from the ASCII text of the essays:

1. **Tokenization:** Tokenization is one of the most common tasks when it comes to working with text data. *Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens.* NLTK contains a module called *tokenize()* which further classifies into two sub-categories:

   Word tokenize: We use the word_tokenize() method to split a sentence into tokens or words.

   Example:

```
from nltk.tokenize import word_tokenize
text = """Founded in 2002, SpaceX’s mission is to
enable humans to become a spacefaring civilization and a
multi-planet species by building a self-sustaining city
on Mars. In 2008, SpaceX’s Falcon 1 became the first
privately developed liquid-fuel launch vehicle to orbit
the Earth."""
word_tokenize(text)
```

Output:

```
['Founded', 'in', '2002', ',', 'SpaceX', '’',
's', 'mission', 'is', 'to', 'enable', 'humans', 'to',
'become', 'a', 'spacefaring', 'civilization', 'and', 'a',
'multi-planet', 'species', 'by', 'building', 'a', 'self-
sustaining', 'city', 'on', 'Mars', '.', 'In', '2008',
',', 'SpaceX', '’', 's', 'Falcon', '1', 'became', 'the',
'first', 'privately', 'developed', 'liquid-fuel',
'launch', 'vehicle', 'to', 'orbit', 'the', 'Earth', '.']
```

Sentence tokenize: We use the sent_tokenize() method to split a document or paragraph into sentences.

Example:

```
from nltk.tokenize import word_tokenize
text = """Founded in 2002, SpaceX’s mission is to
enable humans to become a spacefaring civilization and a
multi-planet species by building a self-sustaining city
on Mars. In 2008, SpaceX’s Falcon 1 became the first
privately developed liquid-fuel launch vehicle to orbit
the Earth."""
sent_tokenize(text)
```

Output: ['Founded in 2002, SpaceX’s mission is to enable humans to become a spacefaring civilization and a multi-planet species by building a self-sustaining city on Mars.',
'In 2008, SpaceX’s Falcon 1 became the first privately developed liquid-fuel launch vehicle to orbit the Earth.']

2. **Stop words Removal:** One of the major forms of pre-processing is to filter out useless data. In natural language processing, useless words (data), are referred to as stop words. A stop word is a commonly used word such as “the”, “a”, “an”, “in”, etc. We would not want these words to take up space in our database, or taking up valuable processing time. NLTK (Natural Language Toolkit) in python has a list of stop words stored in 16 different languages.

   To check the list of stopwords you can type the following commands in the python shell.

```
import nltk
from nltk.corpus import stopwords
print(stopwords.words('english'))
```

```
Output: {‘ourselves’, ‘hers’, ‘between’, ‘yourself’,
‘but’, ‘again’, ‘there’, ‘about’, ‘once’, ‘during’,
‘out’, ‘very’, ‘having’, ‘with’, ‘they’, ‘own’, ‘an’,
‘be’, ‘some’, ‘for’, ……, ‘it’, ‘how’, ‘further’, ‘was’,
‘here’, ‘than’}
```

3. **Part-Of-Speech (POS) Tagging:** Parts of speech Tagging is responsible for reading the text in a language and assigning some specific token (Parts of Speech) to each word. POS tagger is used to assign grammatical information of each word of the sentence. Here we had to apply the *pos_tag* to the before mentioned set of tokenised words by using *nltk.pos_tag(tokenize_text).*
4. **Removing the verbs:** Statistically speaking, for title prediction the use of verbs is very negligible as a title mostly consists of nouns and adjectives. So, by removing the verbs results into a more efficient and simpler database to work on since many useless verbs are removed from the essay. Here we implement it by deleting the words with the POS tag of verb ('*VB*') attached to them. Hence we get a set of nouns and adjectives which gives us our pre-processing dataset.

Thus by following these steps and implementing these functions to an essay, we get a set of nouns and adjectives which will be helpful to build an efficient dictionary which will further help us in choosing the better title for the write-up.

APPROACH, EVALUATION AND RESULTS

Once we are done with the task of extracting features, we use the dataset (The set of nouns and adjectives we got) to create a dictionary of bigrams which will further enhance the prediction of the set of words for the prediction of a perfect title.

We followed the below mentioned steps to achieve the results:

Generating Bigrams: Some English words occur together more frequently. For example - Sky High, best performance, heavy rain, etc. So, in a text document we may need to identify such pair of words which will help in sentiment analysis. First, we need to generate such word pairs from the existing sentence maintain their current sequences. Such pairs are called bigrams. Python has a bigram function as part of NLTK library which helps us generate these pairs: nltk.bigrams.

Example:

```
import nltk
word_data = "The best performance can bring in sky high success."
nltk_tokens = nltk.word_tokenize(word_data)
print(list(nltk.bigrams(nltk_tokens)))
```

Output:

```
[('The', 'best'), ('best', 'performance'), ('performance',
'can'), ('can', 'bring'), ('bring', 'in'), ('in', 'sky'),
('sky', 'high'), ('high', 'success'), ('success', '.')]
```

This result can be used in statistical findings on the frequency of such pairs in a given text. That will correlate to the general sentiment of the descriptions present in the body of the text. Now we will concentrate on creating the vocabulary or dictionary of bigrams.

Creating a Vocabulary/Dictionary: Statistical algorithms work with numbers. However, natural languages contain data in the form of text. Therefore, a mechanism is needed to convert words to numbers. Similarly, after applying different types of processes on the numbers, we need to convert numbers back to text.

One way to achieve this type of functionality is to create a dictionary that assigns a numeric ID to every unique word in the document. The dictionary can then be used to find the numeric equivalent of a word and vice versa.

We have used GENSIM package to create the dictionary by using

```
bigram_dictionary=gensim.corpora.Dictionary(bigram_list)
```

Bigram: After the creation of the dictionary bigram we will find the bigram frequency values for each and every bigram in the dictionary. We take most frequent bigram as the most important bigram which represents the context of the text as a whole. Therefore, we take the most frequent bigram in the document as the most important bigram which gives us a skeleton form for the title which will be further enhanced.

Giving a Basic Structure to the Title: We have already chosen the one Bigram which defines the essay best among the other bigrams. Now we search for the bigrams a level forward and a level backward i.e. we take the first word from the selected bigram and search for the bigrams which has that word as its second part and concatenate to get a better end result. Similarly we check the bigram list for the bigram which has the second word of the previously selected bigram as its first word and concatenate them. We try and find all possible combination and choose the one best suited to be used for title.

Example: Let's consider some selected bigrams we found from one of our article from a test set. In this case the most important bigram among all of them is

```
['neural', 'networks'].
```

And then after searching for all the bigrams a level forward and backward we get a set of bigrams:

```
['multiresolution', 'recurrent']
['recurrent', 'neural']
['networks', 'an']
['an', 'application']
['application', 'dialogue']
['dialogue', 'response']
['response', 'generation']
```

Now we just need to concatenate the bigrams as per the previously mentioned method to get some possible results as the title of the article:

- Recurrent neural networks
- Recurrent neural networks an application
- Multiresolutional recurrent neural networks
- Multiresolutional recurrent neural networks an application
- Multiresolutional recurrent neural networks an application dialogue response generation

FUTURE WORKS

Our project mainly focuses on predicting a perfect title, but there are some flaws which in further research and in future works might be reviewed and will be overcome. Some of them are discussed below.

Implementing Sentence formation algorithms to the words in the title words set that was predicted. Here we have successfully predicted the words which have the most probability to be in the title. Also we have predicted the sequence of how some words must be placed in the title to make complete sense. Implementation of sentence formation on the chosen set of words would bring in the depth and make the title readable and user friendly.

Implementing Sentiment Analysis algorithms would bring in the perfection in title generation. Sentiment analysis is the interpretation and classification of emotions (positive, negative and neutral) within text data using text analysis techniques. Implementing sentence formation will give us a set of possible sentences or phrases which may be used as the title to our article. It is because a group of words can be rearranged to form multiple meaningful sentences. Sentiment analysis would help us choose which sentence among those will be best for describing the key features of the article thus, increasing the efficiency of the article.

BIBLIOGRAPHY

Packages used:

- *Python NLTK Package.*
- *Python GENSIM Package.*
- *Python MatPlotLib Package.*

Book- Information Retrieval By Christopher Manning.

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE (AUTONOMOUS)
RAMAKRISHNA MISSION ASHRAMA
NARENDRAPUR, KOLKATA-700 103, W. B., INDIA
(A Branch Centre of Ramakrishna Mission, Belur Math, Howrah-711202)

Project Completion Certificate

Date: 03/06/2020

This is to certify that **Sri Manohar Mondal** (ROLL NO: CSUG/035/17) a student of Department of Computer Science, Ramakrishna Mission Residential College(Autonomous), has undergone a Project work from January 15, 2020 to May 30, 2020 titled **"Newspaper article classification using NLP".**

Department of Computer Science
Ramakrisnha Mission Residential College
(Autonomous)
Narendrapur, Kolkata-700 103

Dr. Siddhartha Banerjee
(Head of the Department)
Department of Computer Science

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE (AUTONOMOUS)
RAMAKRISHNA MISSION ASHRAMA
NARENDRAPUR, KOLKATA-700 103, W. B., INDIA
(A Branch Centre of Ramakrishna Mission, Belur Math, Howrah-711202)

Project Completion Certificate

Date: 03/06/2020

This is to certify that **Sri Suryadeep Dasgupta** (ROLL NO: CSUG/160/17) a student of Department of Computer Science, Ramakrishna Mission Residential College(Autonomous), has undergone a Project work from January 15, 2020 to May 30, 2020 titled **"Newspaper article classification using NLP"**.

Department of Computer Science
Ramakrishna Mission Residential College
(Autonomous)
Narendrapur, Kolkata 700 103

Dr. Siddhartha Banerjee
(Head of the Department)
Department of Computer Science

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE(AUTONOMOUS),NARENDRAPUR

NEWSPAPER ARTICLE CLASSIFICATION USING NATURAL LANGUAGE PROCESSING (NLP)

NAME : MANOHAR MONDAL
ROLL NO : CSUG/035/16

NAME : SURYADEEP DASGUPTA
ROLL NO : CSUG/160/17

Department of Computer Science

2020

NEWSPAPER ARTICLE CLASSIFICATION USING NATURAL LANGUAGE PROCESSING (NLP)

ABSTRACT- Here, we consider the problem to classify newspaper articles into various categories. We first take input a dataset and filter the text to make it easier to classify then we remove all the stop words which are not required. We then try to classify the articles using various classifiers. We then try to better that algorithm using hyper parameter tuning. Here, we also use tf-idf weighting scheme and word count vectors scheme to see how human brain can classify article topics.

TEXT CLASSIFICATION:

It is also known as text tagging or text categorization is the process of categorizing text into organized groups.

By using NLP, text classifiers can automatically analyze text and then assign a pre-defined tags or categories based on its contents.

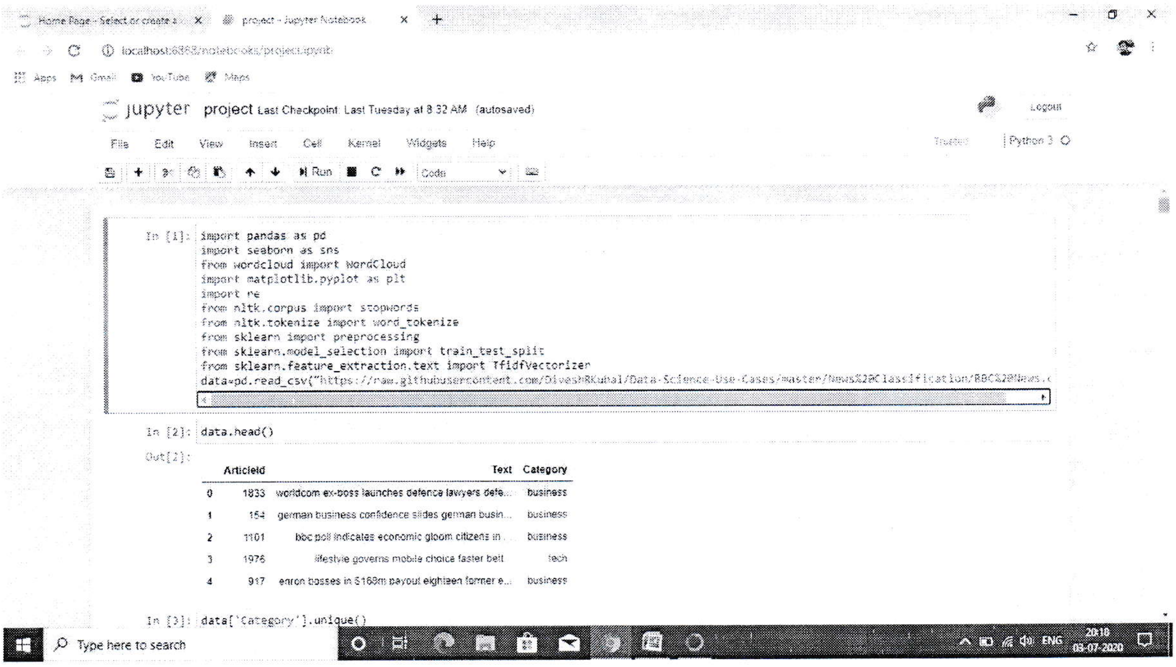
APPLICATIONS:

1. Sentiment analysis.
2. Topic detection.
3. Language detection.
4. Differentiate between legitimate messages and spam mail.
5. News portal.

THE DATA:

The data is taken by using the url: https://raw.githubusercontent.com/DiveshRKubal/Data-Science-Use-Cases/master/News%20Classification/BBC%20News.csv

The data contains 1490 rows and 3 columns. The columns are ArticleId, Text, Category.



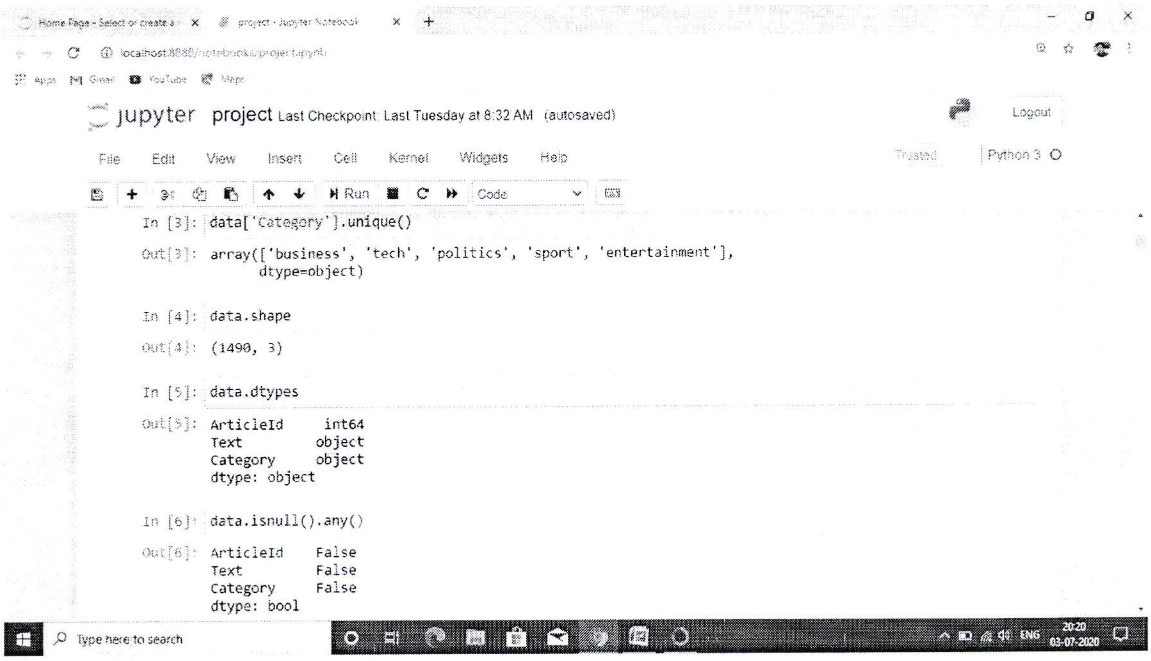
Home Page - Select or create a
project - Jupyter Notebook
localhost:8888/notebooks/project.ipynb
Apps
Gmail
YouTube
Maps
jupyter
project Last Checkpoint: Last Tuesday at 8:32 AM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Type here to search
ENG
20:18
03-07-2020

```
In [1]: import pandas as pd
        import seaborn as sns
        from wordcloud import WordCloud
        import matplotlib.pyplot as plt
        import re
        from nltk.corpus import stopwords
        from nltk.tokenize import word_tokenize
        from sklearn import preprocessing
        from sklearn.model_selection import train_test_split
        from sklearn.feature_extraction.text import TfidfVectorizer
        data=pd.read_csv("https://raw.githubusercontent.com/DiveshRKubal/Data-Science-Use-Cases/master/News%20Classification/BBC%20News.c
```

```
In [2]: data.head()

Out[2]:
```

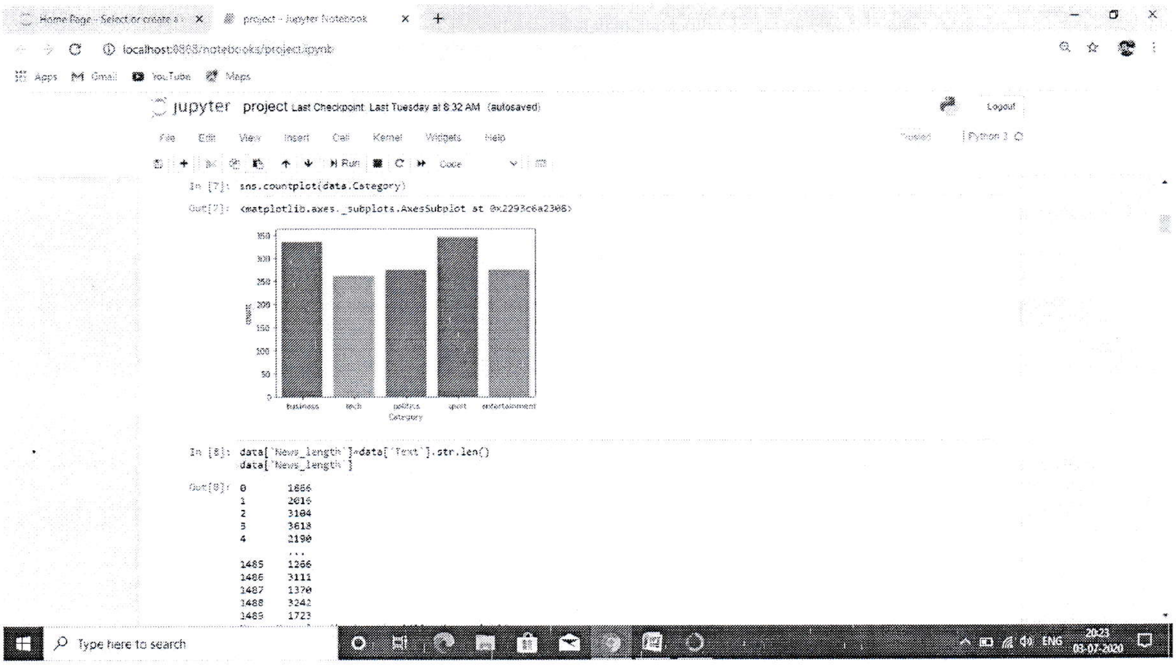
	ArticleId	Text	Category
0	1833	worldcom ex-boss launches defence lawyers defe...	business
1	154	german business confidence slides german busin...	business
2	1101	bbc poll indicates economic gloom citizens in ...	business
3	1976	lifestyle governs mobile choice faster bett	tech
4	917	enron bosses in $168m payout eighteen former e...	business

```
In [3]: data['Category'].unique()
```



Home Page - Select or create a
project - Jupyter Notebook
localhost:8888/notebooks/project.ipynb
Apps
Gmail
YouTube
Maps
jupyter
project Last Checkpoint: Last Tuesday at 8:32 AM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Type here to search
ENG
20:20
03-07-2020

```
In [3]: data['Category'].unique()

Out[3]: array(['business', 'tech', 'politics', 'sport', 'entertainment'],
              dtype=object)

In [4]: data.shape

Out[4]: (1490, 3)

In [5]: data.dtypes

Out[5]: ArticleId     int64
        Text         object
        Category     object
        dtype: object

In [6]: data.isnull().any()

Out[6]: ArticleId    False
        Text         False
        Category     False
        dtype: bool
```

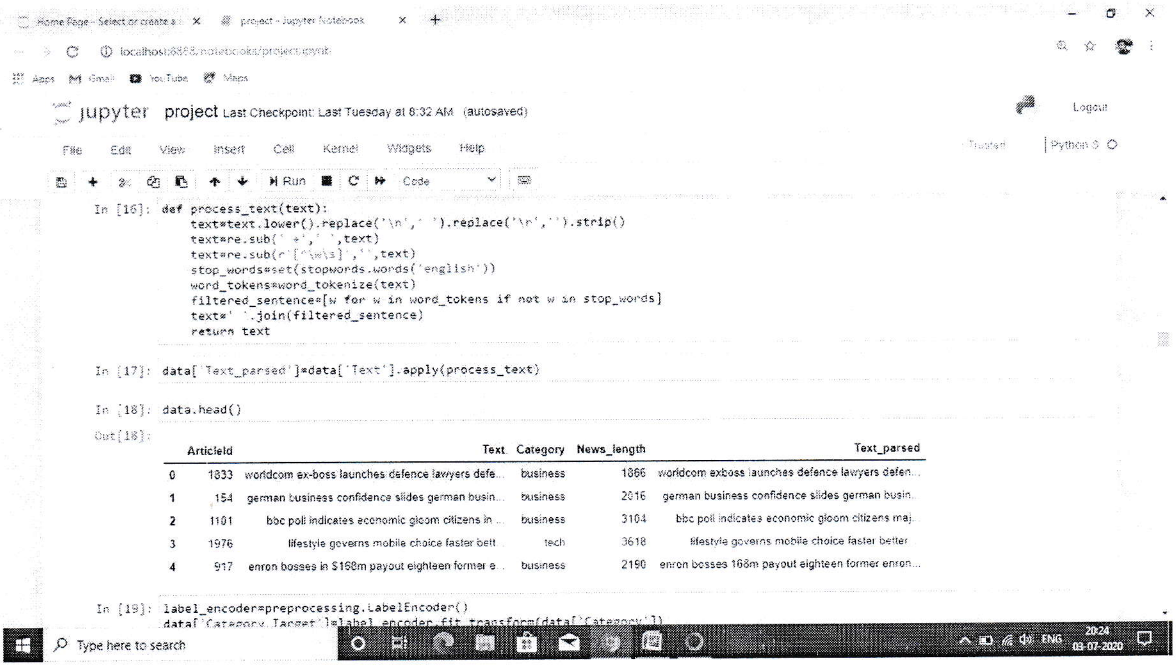


```
In [7]: sns.countplot(data.Category)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x2293c6a2308>
```

```
In [8]: data['News_length']=data['Text'].str.len()
        data['News_length']

Out[8]: 0        1866
        1        2016
        2        3104
        3        3618
        4        2190
                 ...
        1485     1266
        1486     3111
        1487     1370
        1488     3242
        1489     1723
```

FILTERED SENTENCE:

We define a function which is used to filter a sentence. Here we first convert the whole text to lower case, replacing new line with a single space and \r with space. Then replace multiple spaces with a single space and removing all the special characters. We also remove all the stop words from the data using stop words and word tokens.

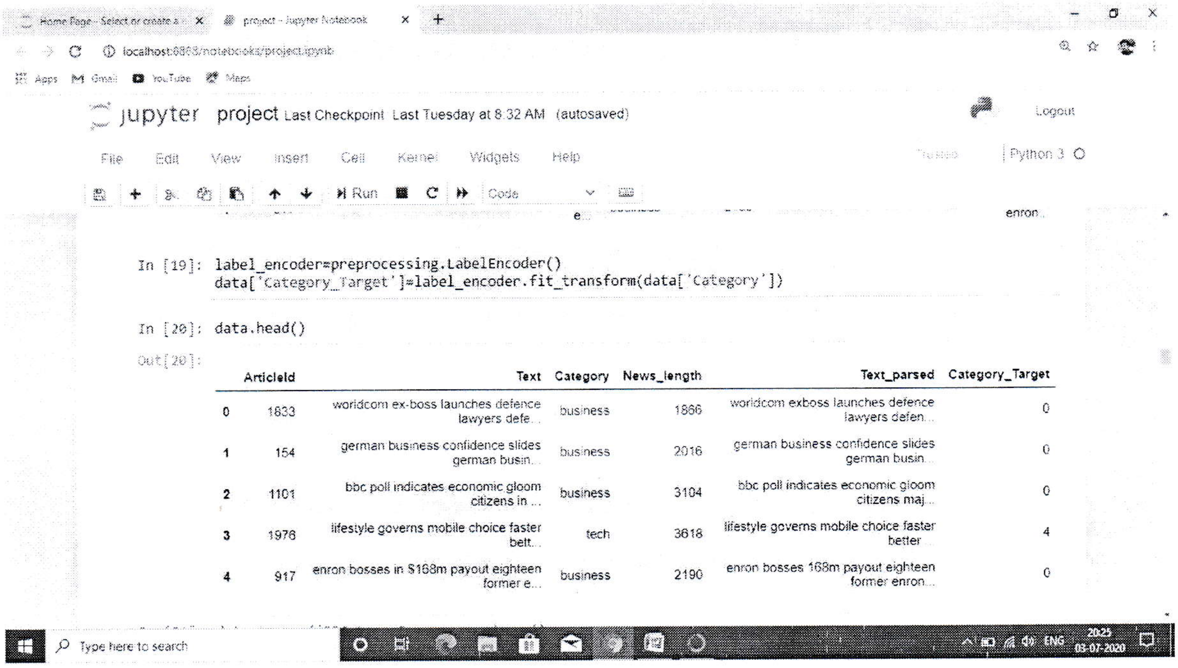

Home Page - Select or create a
project - Jupyter Notebook
localhost:8888/notebooks/project.ipynb
Apps
Gmail
YouTube
Maps
jupyter
project Last Checkpoint: Last Tuesday at 8:32 AM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Type here to search
ENG
20:24
03-07-2020

```
In [16]: def process_text(text):
             text=text.lower().replace('\n',' ').replace('\r','').strip()
             text=re.sub(' +',' ',text)
             text=re.sub(r'[^\w\s]','',text)
             stop_words=set(stopwords.words('english'))
             word_tokens=word_tokenize(text)
             filtered_sentence=[w for w in word_tokens if not w in stop_words]
             text=' '.join(filtered_sentence)
             return text

In [17]: data['Text_parsed']=data['Text'].apply(process_text)

In [18]: data.head()

Out[18]:
     ArticleId                                         Text  Category  News_length                                   Text_parsed
  0       1833  worldcom ex-boss launches defence lawyers defe...  business         1866  worldcom exboss launches defence lawyers defen...
  1        154  german business confidence slides german busin...  business         2016  german business confidence slides german busin...
  2       1101      bbc poll indicates economic gloom citizens in ...  business         3104      bbc poll indicates economic gloom citizens maj...
  3       1976          lifestyle governs mobile choice faster bett...      tech         3618          lifestyle governs mobile choice faster better...
  4        917  enron bosses in $168m payout eighteen former e...  business         2190  enron bosses 168m payout eighteen former enron...

In [19]: label_encoder=preprocessing.LabelEncoder()
         data['Category_Target']=label_encoder.fit_transform(data['Category'])
```

ENCODING THE DATA:

The data which we have taken input is then encoded on the basis of its categories in the following order:

Category	ENCODED_CATEGORY
BUSINESS	0
ENTERTAINMENT	1
POLITICS	2
SPORT	3
TECH	4

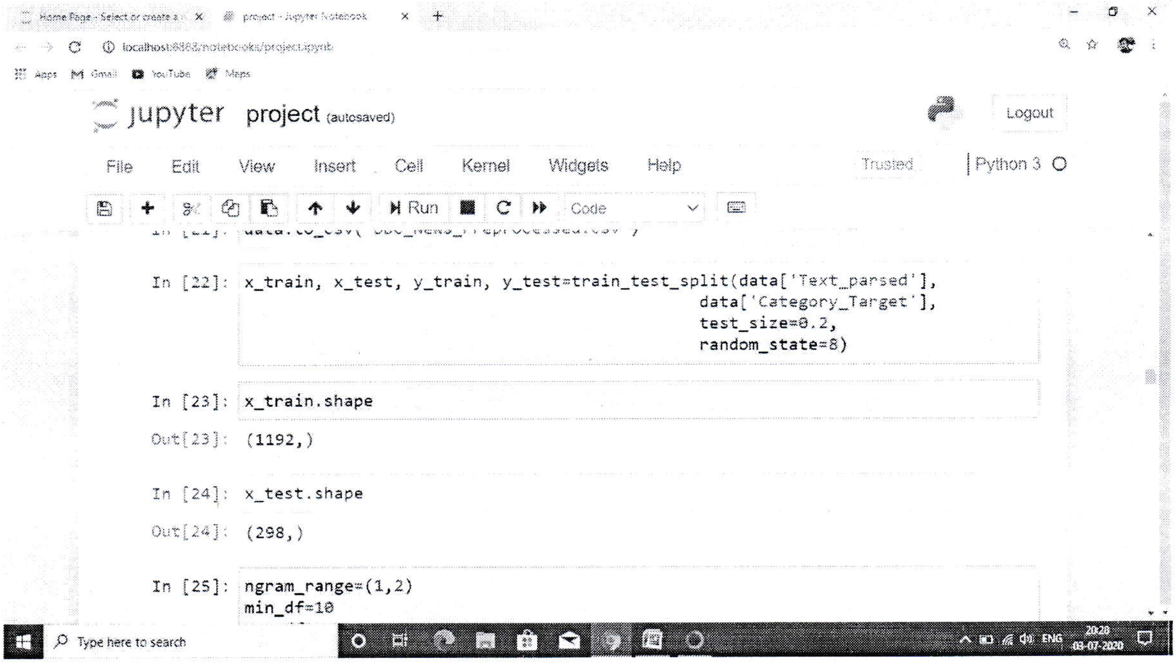

jupyter project Last Checkpoint Last Tuesday at 8 32 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Python 3
Run
Code

```
In [19]: label_encoder=preprocessing.LabelEncoder()
         data['Category_Target']=label_encoder.fit_transform(data['Category'])

In [20]: data.head()

Out[20]:
```

	ArticleId	Text	Category	News_length	Text_parsed	Category_Target
0	1833	worldcom ex-boss launches defence lawyers defe...	business	1866	worldcom exboss launches defence lawyers defen...	0
1	154	german business confidence slides german busin...	business	2016	german business confidence slides german busin...	0
2	1101	bbc poll indicates economic gloom citizens in ...	business	3104	bbc poll indicates economic gloom citizens maj...	0
3	1976	lifestyle governs mobile choice faster belt...	tech	3618	lifestyle governs mobile choice faster better ...	4
4	917	enron bosses in $168m payout eighteen former e...	business	2190	enron bosses 168m payout eighteen former enron...	0

TESTING AND TRAINING:

The data we get is divided into two categories: Training and Testing. Here, we take two columns for testing and training namely: filtered data and the encoded_category. The number of training set is considered about 80% of the total data set where as the testing dataset is about 20% of the data set.


jupyter project (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search

```
In [22]: x_train, x_test, y_train, y_test=train_test_split(data['Text_parsed'],
                                                        data['Category_Target'],
                                                        test_size=0.2,
                                                        random_state=8)

In [23]: x_train.shape

Out[23]: (1192,)

In [24]: x_test.shape

Out[24]: (298,)

In [25]: ngram_range=(1,2)
         min_df=10
```

TERM FREQUENCY INVERSE DOCUMENT FREQUENCY MODEL:

We are looking for high informative words. Term frequency- inverse document frequency (tf-idf), a numerical statistic that reflects how important a word is to a document in a collection or corpus.

The first element is the term frequency (tf), it is the number of times that a token occurs in an article, summed across all the articles in a particular class.

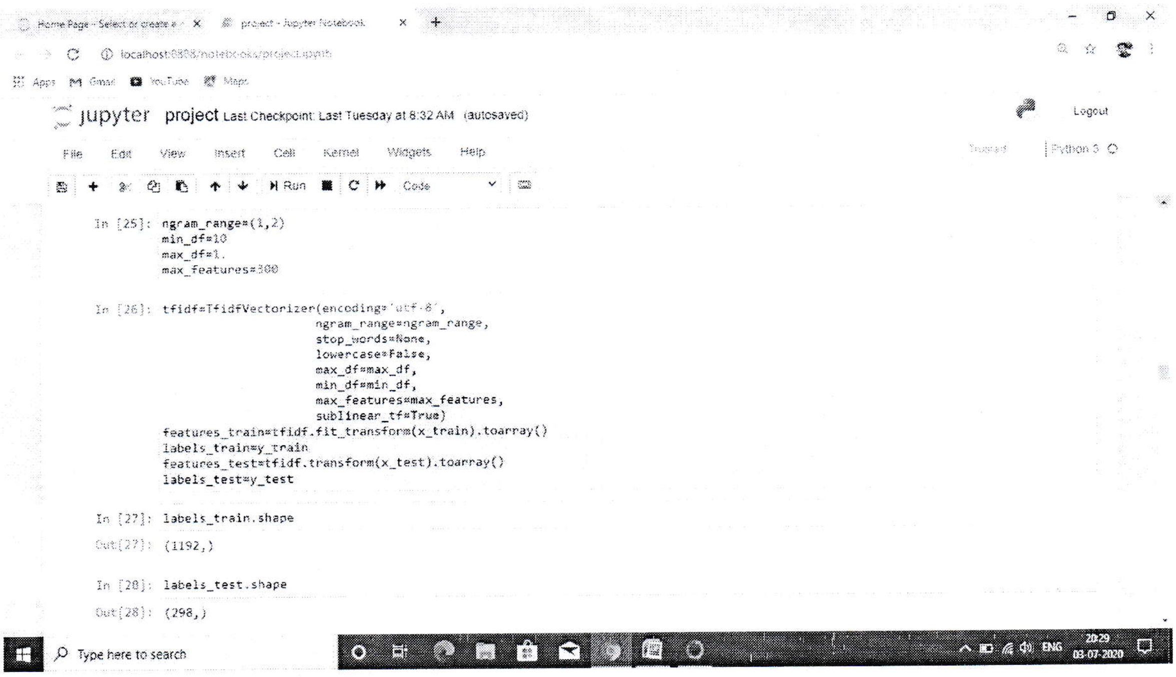
The inverse document frequency (idf) is obtained by dividing the total number of words in the corpus by the count of the instances of the particular word in the data, and then taking the logarithm of the quotient.

When creating the features with this method, we can choose some parameters:

- N-gram range: we are able to consider unigrams, bigrams, trigrams...
- Maximum/Minimum Document Frequency: when building the vocabulary, we can ignore terms that have a document frequency strictly higher/lower than the given threshold.

- Maximum features: we can choose the top *N* features ordered by term frequency across the corpus.

We expect that bigrams help to improve our model performance by taking into consideration words that tend to appear together in the documents. We have chosen a value of Minimum DF equal to 10 to get rid of extremely rare words that don't appear in more than 10 documents, and a Maximum DF equal to 100% to not ignore any other words. The election of 300 as maximum number of features has been made because we want to avoid possible overfitting, often arising from a large number of features compared to the number of training observations.



Home Page - Select or create a
project - Jupyter Notebook
localhost:8888/notebooks/project.ipynb
Apps
Gmail
YouTube
Maps
jupyter
project Last Checkpoint: Last Tuesday at 8:32 AM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Python 3
Run
Code
Type here to search
ENG
20:29
03-07-2020

```
In [25]: ngram_range=(1,2)
         min_df=10
         max_df=1.
         max_features=300

In [26]: tfidf=TfidfVectorizer(encoding='utf-8',
                               ngram_range=ngram_range,
                               stop_words=None,
                               lowercase=False,
                               max_df=max_df,
                               min_df=min_df,
                               max_features=max_features,
                               sublinear_tf=True)
         features_train=tfidf.fit_transform(x_train).toarray()
         labels_train=y_train
         features_test=tfidf.transform(x_test).toarray()
         labels_test=y_test

In [27]: labels_train.shape

Out[27]: (1192,)

In [28]: labels_test.shape

Out[28]: (298,)
```

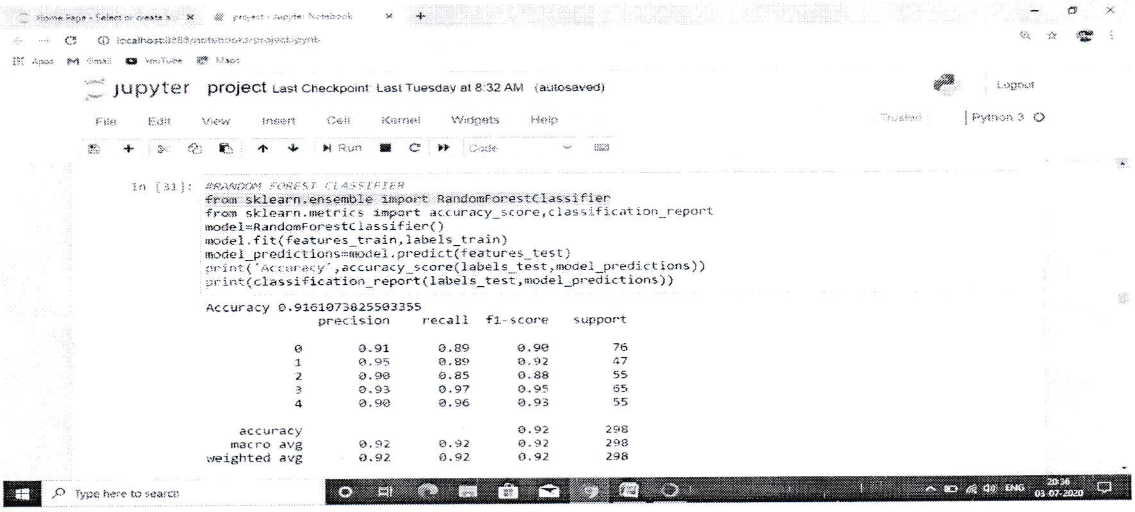
CLASSIFIERS:

We use different types of classifiers to classify the parsed_text to the embedded categories such as:

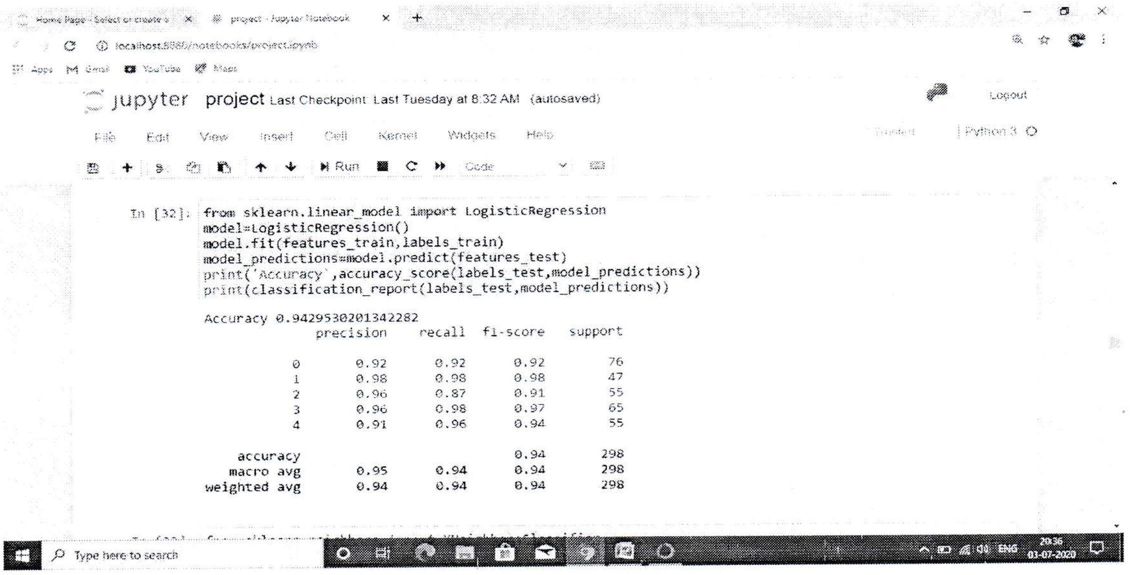
- RANDOM FOREST CLASSIFIER
- LOGISTIC REGRESSION

- KNEIGHBORS CLASSIFIER
- DECISION TREE CLASSIFIER
- GAUSSIAN NAIVE_BAYES

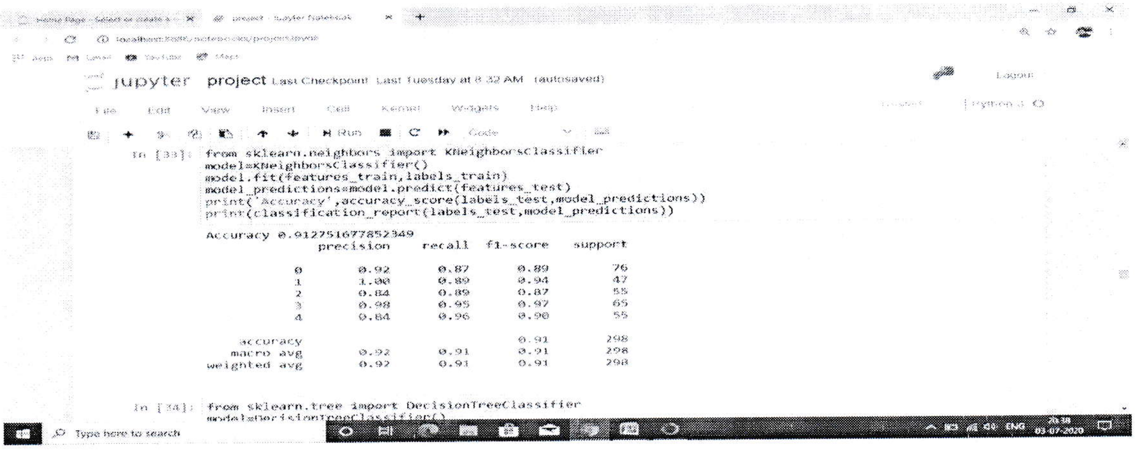
Out of all the classifier we can see that the Logistic Regression provides us with the highest accuracy. Hence we are going to use the Logistic Regression.



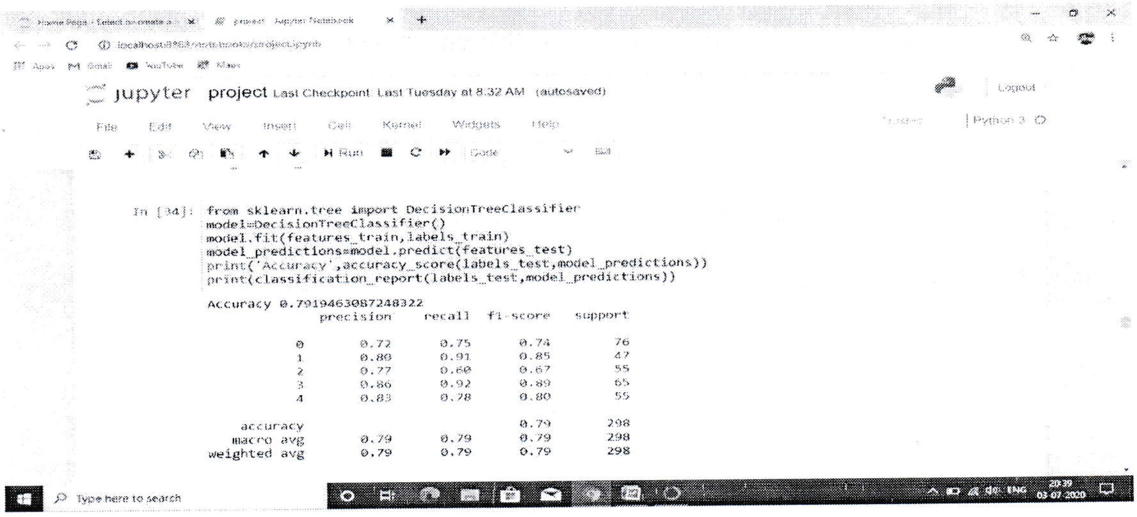
```
In [31]: #RANDOM FOREST CLASSIFIER
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.metrics import accuracy_score,classification_report
         model=RandomForestClassifier()
         model.fit(features_train,labels_train)
         model_predictions=model.predict(features_test)
         print('Accuracy',accuracy_score(labels_test,model_predictions))
         print(classification_report(labels_test,model_predictions))

         Accuracy 0.9161073825503355
                       precision    recall  f1-score   support

                    0       0.91      0.89      0.90        76
                    1       0.95      0.89      0.92        47
                    2       0.90      0.85      0.88        55
                    3       0.93      0.97      0.95        65
                    4       0.90      0.96      0.93        55

             accuracy                           0.92       298
            macro avg       0.92      0.92      0.92       298
         weighted avg       0.92      0.92      0.92       298
```



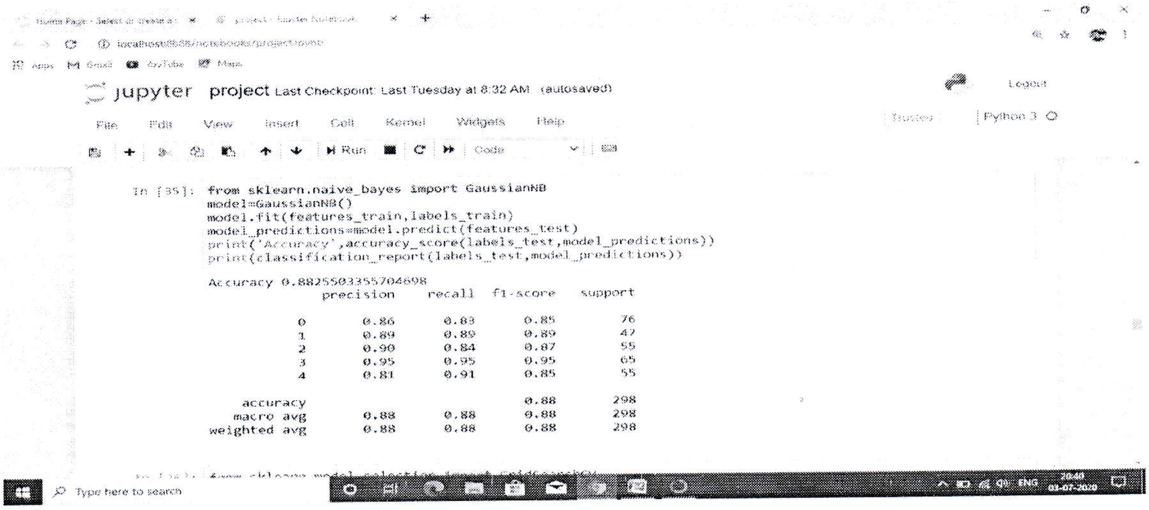
```
In [32]: from sklearn.linear_model import LogisticRegression
         model=LogisticRegression()
         model.fit(features_train,labels_train)
         model_predictions=model.predict(features_test)
         print('Accuracy',accuracy_score(labels_test,model_predictions))
         print(classification_report(labels_test,model_predictions))

         Accuracy 0.9429530201342282
                       precision    recall  f1-score   support

                    0       0.92      0.92      0.92        76
                    1       0.98      0.98      0.98        47
                    2       0.96      0.87      0.91        55
                    3       0.96      0.98      0.97        65
                    4       0.91      0.96      0.94        55

             accuracy                           0.94       298
            macro avg       0.95      0.94      0.94       298
         weighted avg       0.94      0.94      0.94       298
```



```
In [33]: from sklearn.neighbors import KNeighborsClassifier
         model=KNeighborsClassifier()
         model.fit(features_train,labels_train)
         model_predictions=model.predict(features_test)
         print('Accuracy',accuracy_score(labels_test,model_predictions))
         print(classification_report(labels_test,model_predictions))

         Accuracy 0.912751677852349
                       precision    recall  f1-score   support

                    0       0.92      0.87      0.89        76
                    1       1.00      0.89      0.94        47
                    2       0.84      0.89      0.87        55
                    3       0.98      0.95      0.97        65
                    4       0.84      0.96      0.90        55

             accuracy                           0.91       298
            macro avg       0.92      0.91      0.91       298
         weighted avg       0.92      0.91      0.91       298

In [34]: from sklearn.tree import DecisionTreeClassifier
         model=DecisionTreeClassifier()
```



```
In [34]: from sklearn.tree import DecisionTreeClassifier
         model=DecisionTreeClassifier()
         model.fit(features_train,labels_train)
         model_predictions=model.predict(features_test)
         print('Accuracy',accuracy_score(labels_test,model_predictions))
         print(classification_report(labels_test,model_predictions))

         Accuracy 0.7919463087248322
                       precision    recall  f1-score   support

                    0       0.72      0.75      0.74        76
                    1       0.80      0.91      0.85        47
                    2       0.77      0.60      0.67        55
                    3       0.86      0.92      0.89        65
                    4       0.83      0.78      0.80        55

             accuracy                           0.79       298
            macro avg       0.79      0.79      0.79       298
         weighted avg       0.79      0.79      0.79       298
```



```
In [35]: from sklearn.naive_bayes import GaussianNB
         model=GaussianNB()
         model.fit(features_train,labels_train)
         model_predictions=model.predict(features_test)
         print('Accuracy',accuracy_score(labels_test,model_predictions))
         print(classification_report(labels_test,model_predictions))

         Accuracy 0.8825503355704698
                       precision    recall  f1-score   support

                    0       0.86      0.83      0.85        76
                    1       0.89      0.89      0.89        47
                    2       0.90      0.84      0.87        55
                    3       0.95      0.95      0.95        65
                    4       0.81      0.91      0.85        55

             accuracy                           0.88       298
            macro avg       0.88      0.88      0.88       298
         weighted avg       0.88      0.88      0.88       298
```

PERFORMANCE MEASUREMENT:

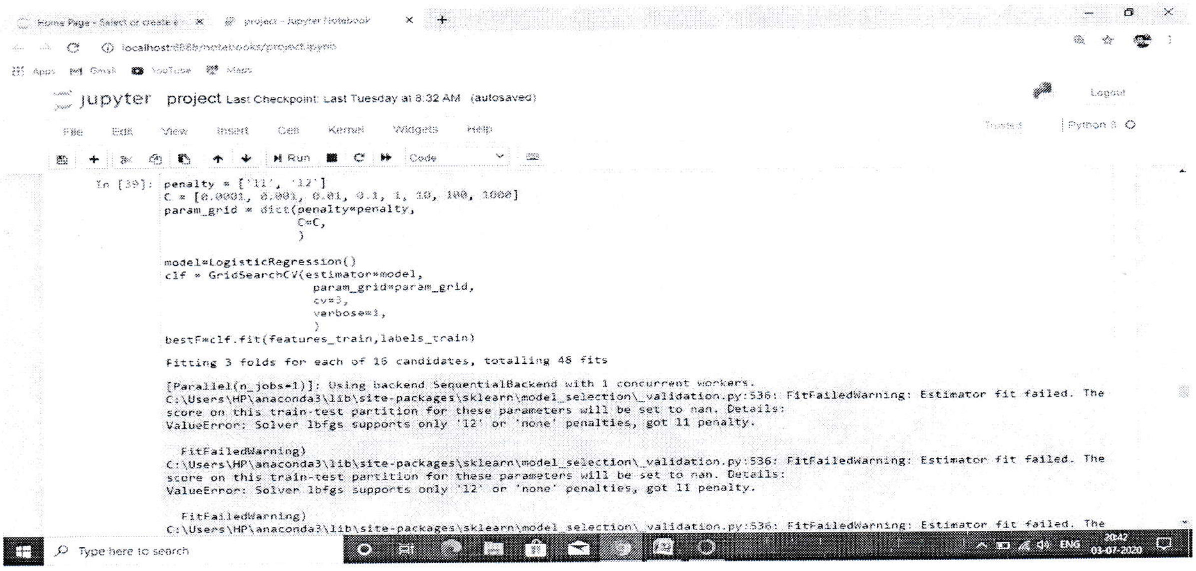
The classifications are measured on various parameters. The data is first made to be fit for the training data set and the model is cross validated using the testing data set. The parameters to check the insights of how the models are working are:

- **Accuracy**: the accuracy metric measures the ratio of correct predictions over the total number of instances evaluated.
- **Precision**: precision is used to measure the positive patterns that are correctly predicted from the total predicted patterns in a positive class.
- **Recall**: recall is used to measure the fraction of positive patterns that are correctly classified
- **F1-Score**: this metric represents the harmonic mean between recall and precision values

HYPER-PARAMETERS:

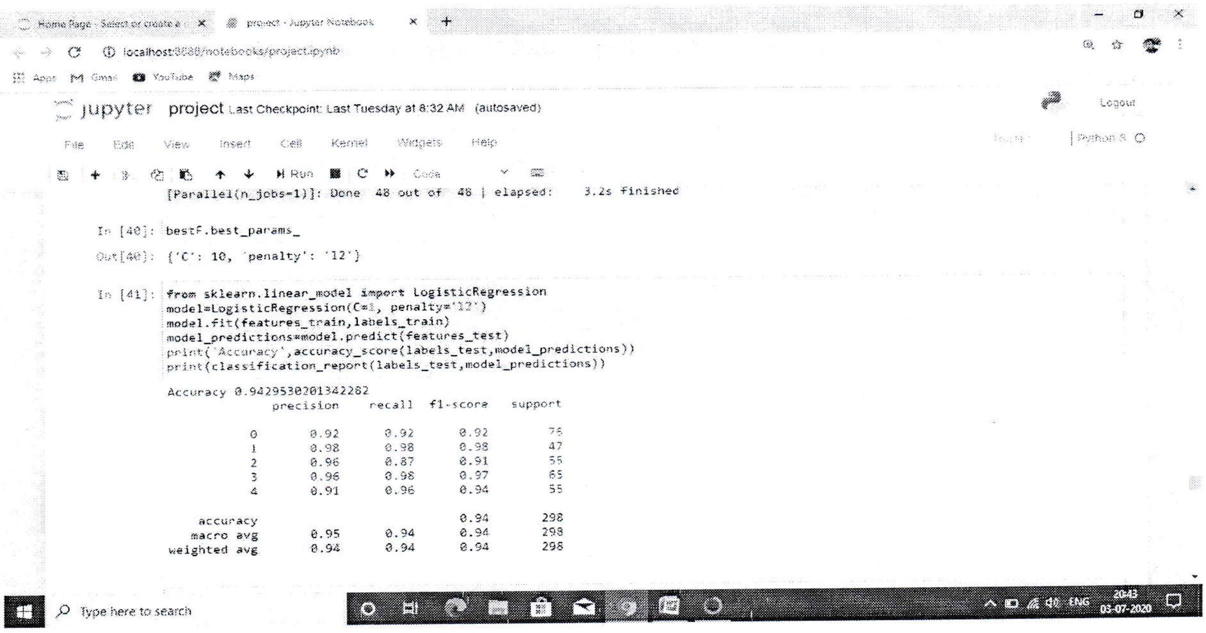
Each of the classifier has multiple parameters that also need to be tuned. Firstly, we have to decide the best hyper parameter which can get the best fitting model to classify.

After performing the hyper parameter tuning process with the training data via cross validation and fitting the model to this training data, we need to evaluate its performance on totally unseen data (the test set). Here in the Logistic Regression we find out the best fit hyper-parameters using GridSearchCV(). Then we use that parameter to get the best possible model for that classifier.



jupyter project Last Checkpoint: Last Tuesday at 8:32 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
20:42
03-07-2020

```
In [39]: penalty = ['l1', 'l2']
         C = [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000]
         param_grid = dict(penalty=penalty,
                           C=C,
                           )

         model=LogisticRegression()
         clf = GridSearchCV(estimator=model,
                            param_grid=param_grid,
                            cv=3,
                            verbose=1,
                            )
         bestF=clf.fit(features_train,labels_train)

         Fitting 3 folds for each of 16 candidates, totalling 48 fits

         [Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
         C:\Users\HP\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator fit failed. The
         score on this train-test partition for these parameters will be set to nan. Details:
         ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

           FitFailedWarning)
         C:\Users\HP\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator fit failed. The
         score on this train-test partition for these parameters will be set to nan. Details:
         ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

           FitFailedWarning)
         C:\Users\HP\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator fit failed. The
```



jupyter project Last Checkpoint: Last Tuesday at 8:32 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
20:43
03-07-2020

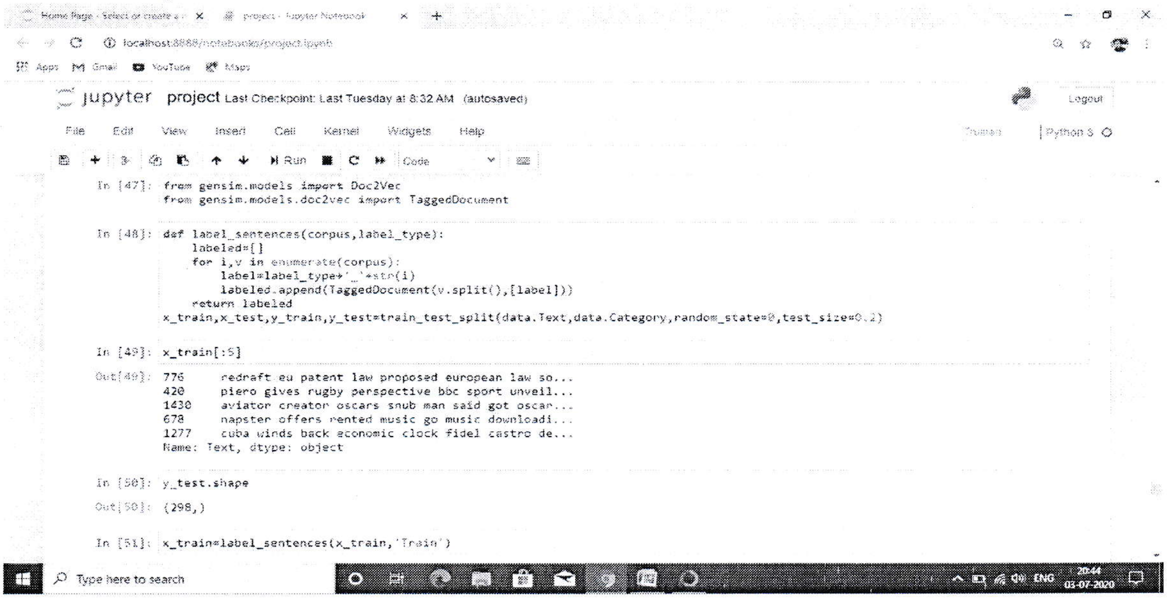
```
         [Parallel(n_jobs=1)]: Done  48 out of  48 | elapsed:    3.2s finished

In [40]: bestF.best_params_

Out[40]: {'C': 10, 'penalty': 'l2'}

In [41]: from sklearn.linear_model import LogisticRegression
         model=LogisticRegression(C=1, penalty='l2')
         model.fit(features_train,labels_train)
         model_predictions=model.predict(features_test)
         print('Accuracy',accuracy_score(labels_test,model_predictions))
         print(classification_report(labels_test,model_predictions))

         Accuracy 0.9429530201342282
                       precision    recall  f1-score   support

                    0       0.92      0.92      0.92        76
                    1       0.98      0.98      0.98        47
                    2       0.96      0.87      0.91        55
                    3       0.96      0.98      0.97        65
                    4       0.91      0.96      0.94        55

             accuracy                           0.94       298
            macro avg       0.95      0.94      0.94       298
         weighted avg       0.94      0.94      0.94       298
```

DOC2VEC:

The input of texts (i.e. word) per document can be various while the output is fixed-length vectors. Paragraph vector and word vectors are initialized. Paragraph vector is unique among all documents while word vectors are shared among all documents such that word vector can be learnt from different document.

During training phase, word vectors will be trained while paragraph will be thrown away after that. During the prediction phase the paragraph phase will be initialized randomly and compute with word vectors.

After carrying out the Doc2Vec operation we have carried the logistic regression on the Doc2Vec model we find that the accuracy of the logistic regression model has certainly increased.



Home Page - Select or create a
project - Jupyter Notebook
localhost:8888/notebooks/project.ipynb
Apps
Gmail
YouTube
Maps
jupyter
project Last Checkpoint: Last Tuesday at 8:32 AM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Type here to search
ENG
20:44
03-07-2020

```
In [47]: from gensim.models import Doc2Vec
         from gensim.models.doc2vec import TaggedDocument

In [48]: def label_sentences(corpus,label_type):
             labeled=[]
             for i,v in enumerate(corpus):
                 label=label_type+'_'+str(i)
                 labeled.append(TaggedDocument(v.split(),[label]))
             return labeled
         x_train,x_test,y_train,y_test=train_test_split(data.Text,data.Category,random_state=0,test_size=0.2)

In [49]: x_train[:5]

Out[49]: 776     redraft eu patent law proposed european law so...
         420     piero gives rugby perspective bbc sport unveil...
         1430    aviator creator oscars snub man said got oscar...
         678     napster offers rented music go music downloadi...
         1277    cuba winds back economic clock fidel castro de...
         Name: Text, dtype: object

In [50]: y_test.shape

Out[50]: (298,)

In [51]: x_train=label_sentences(x_train,'Train')
```

What is a Bag-of-Words?

A bag-of-words model, or BoW for short, is a way of extracting features from text for use in modeling, such as with machine learning algorithms.

The approach is very simple and flexible, and can be used in a myriad of ways for extracting features from documents.

A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things:

1. A vocabulary of known words.
2. A measure of the presence of known words.

It is called a *"bag"* of words, because any information about the order or structure of words in the document is discarded. The model is only concerned with whether known words occur in the document, not where in the document.

Example of the Bag-of-Words Model

Let’s make the bag-of-words model concrete with a worked example.

Step 1: Collect Data

Below is a snippet of the first few lines of text from the book “A Tale of Two Cities” by Charles Dickens, taken from Project Gutenberg.
It was the best of times,
it was the worst of times,
it was the age of wisdom,
it was the age of foolishness,

For this small example, let’s treat each line as a separate “document” and the 4 lines as our entire corpus of documents.

Step 2: Design the Vocabulary

Now we can make a list of all of the words in our model vocabulary.
The unique words here (ignoring case and punctuation) are:

- “it”
- “was”
- “the”
- “best”
- “of”
- “times”
- “worst”
- “age”
- “wisdom”
- “foolishness”

That is a vocabulary of 10 words from a corpus containing 24 words.

Step 3: Create Document Vectors

The next step is to score the words in each document.

The objective is to turn each document of free text into a vector that we can use as input or output for a machine learning model.

Because we know the vocabulary has 10 words, we can use a fixed-length document representation of 10, with one position in the vector to score each word.

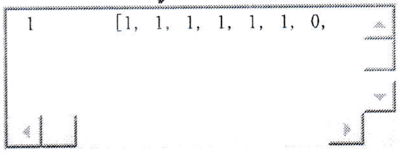
The simplest scoring method is to mark the presence of words as a boolean value, 0 for absent, 1 for present.

Using the arbitrary ordering of words listed above in our vocabulary, we can step through the first document ("*It was the best of times*") and convert it into a binary vector.

The scoring of the document would look as follows:

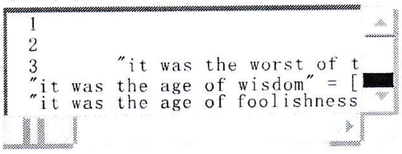
- "it" = 1
- "was" = 1
- "the" = 1
- "best" = 1
- "of" = 1
- "times" = 1
- "worst" = 0
- "age" = 0
- "wisdom" = 0
- "foolishness" = 0

As a binary vector, this would look as follows:



```
        [1, 1, 1, 1, 1, 1, 0,
```

The other three documents would look as follows:



```
        "it was the worst of t
"it was the age of wisdom" = [
"it was the age of foolishness
```

All ordering of the words is nominally discarded and we have a consistent way of extracting features from any document in our corpus, ready for use in modeling.

New documents that overlap with the vocabulary of known words, but may contain words outside of the vocabulary, can still be encoded, where only the occurrence of known words are scored and unknown words are ignored.

You can see how this might naturally scale to large vocabularies and larger documents.

Managing Vocabulary

As the vocabulary size increases, so does the vector representation of documents.

In the previous example, the length of the document vector is equal to the number of known words.

You can imagine that for a very large corpus, such as thousands of books, that the length of the vector might be thousands or millions of positions. Further, each document may contain very few of the known words in the vocabulary.

This results in a vector with lots of zero scores, called a sparse vector or sparse representation.

Sparse vectors require more memory and computational resources when modeling and the vast number of positions or dimensions can make the modeling process very challenging for traditional algorithms.

As such, there is pressure to decrease the size of the vocabulary when using a bag-of-words model.

There are simple text cleaning techniques that can be used as a first step, such as:

- Ignoring case
- Ignoring punctuation
- Ignoring frequent words that don't contain much information, called stop words, like "a," "of," etc.
- Fixing misspelled words.
- Reducing words to their stem (e.g. "play" from "playing") using stemming algorithms.

A more sophisticated approach is to create a vocabulary of grouped words. This both changes the scope of the vocabulary and allows the bag-of-words to capture a little bit more meaning from the document.

In this approach, each word or token is called a "gram". Creating a vocabulary of two-word pairs is, in turn, called a bigram model. Again, only the bigrams that appear in the corpus are modeled, not all possible bigrams.

An N-gram is an N-token sequence of words: a 2-gram (more commonly called a bigram) is a two-word sequence of words like "please turn", "turn your", or "your homework", and a 3-gram (more commonly called a trigram) *is a three-word sequence of words like “please turn your”, or “turn your homework”.*

For example, the bigrams in the first line of text in the previous section: “It was the best of times” are as follows:

- “it was”
- “was the”
- “the best”
- “best of”
- “of times”

A vocabulary then tracks triplets of words is called a trigram model and the general approach is called the n-gram model, where n refers to the number of grouped words.

Often a simple bigram approach is better than a 1-gram bag-of-words model for tasks like documentation classification.

a bag-of-bigrams representation is much more powerful than bag-of-words, and in many cases proves very hard to beat.

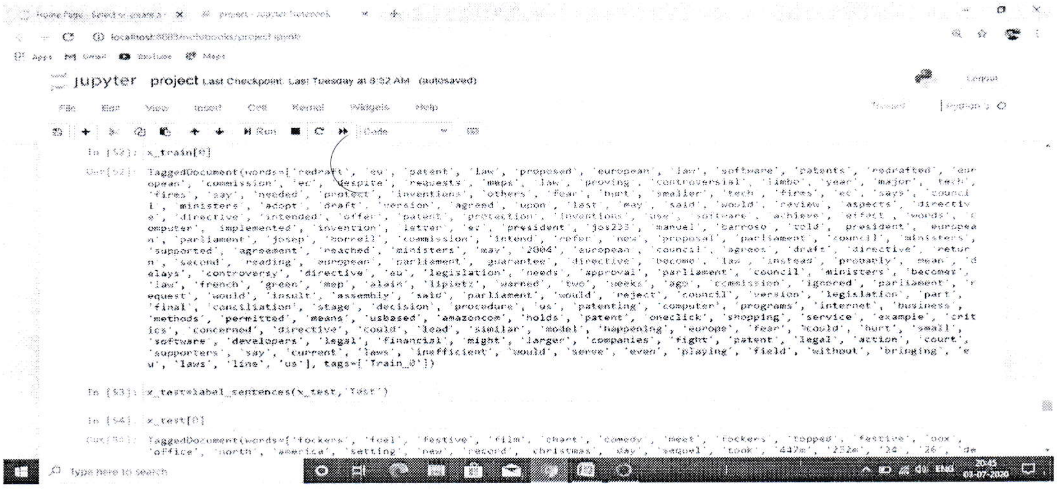
Scoring Words

Once a vocabulary has been chosen, the occurrence of words in example documents needs to be scored.

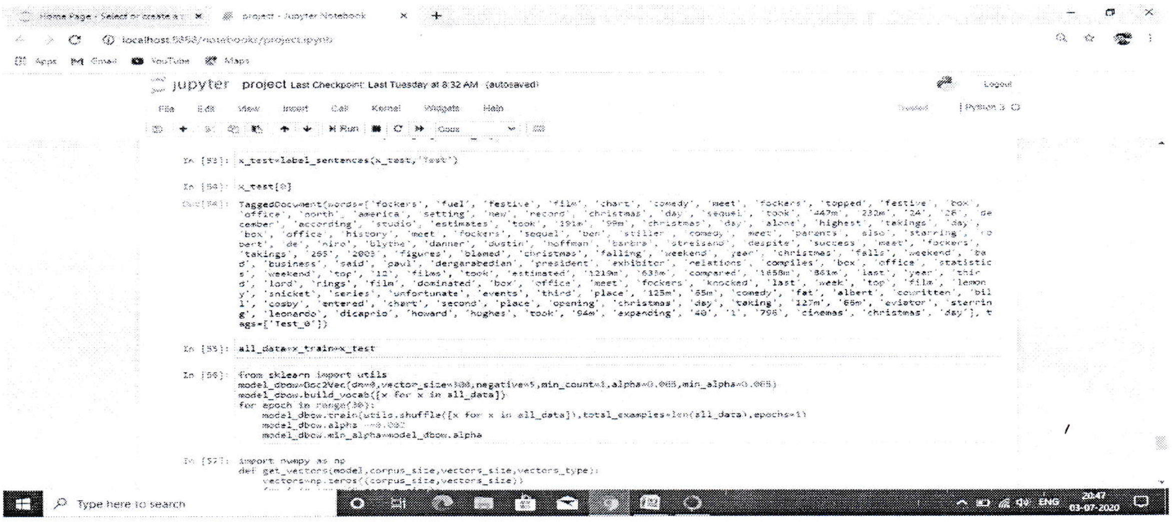
In the worked example, we have already seen one very simple approach to scoring: a binary scoring of the presence or absence of words.

Some additional simple scoring methods include:

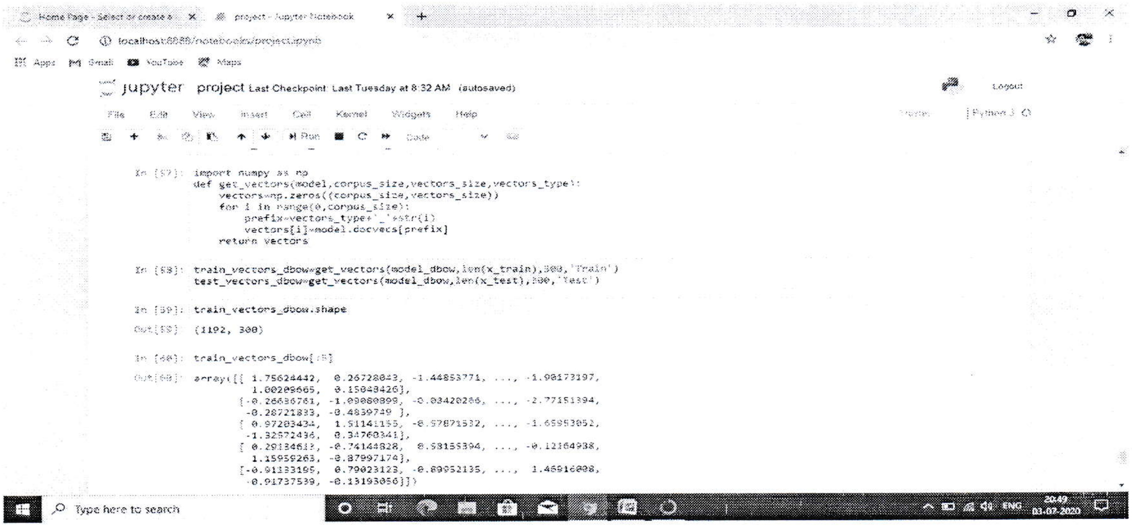
- **Counts**. Count the number of times each word appears in a document.
- **Frequencies**. Calculate the frequency that each word appears in a document out of all the words in the document.



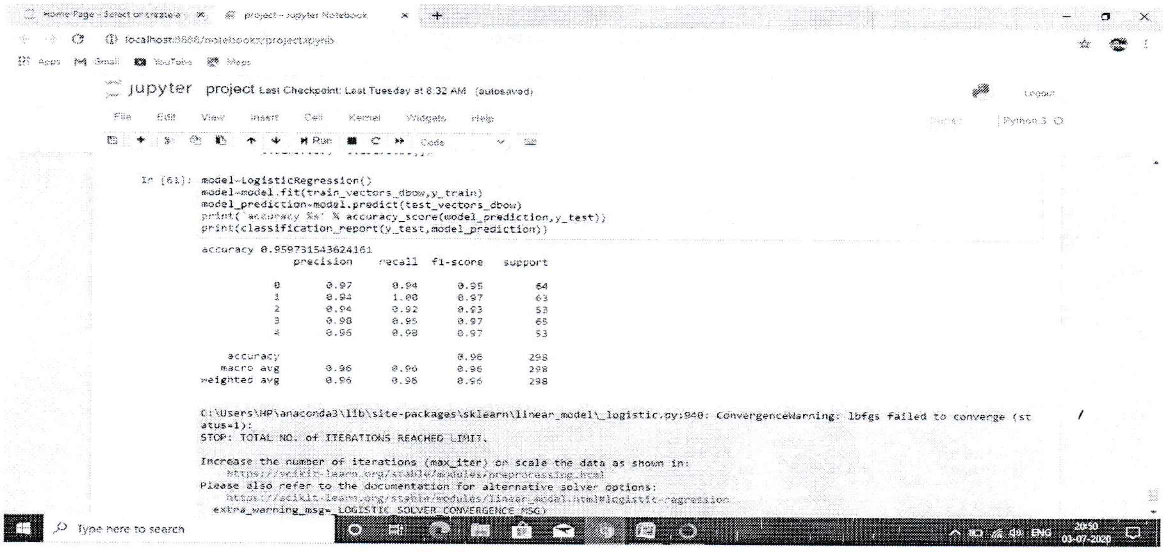
```
In [52]: x_train[0]

Out[52]: TaggedDocument(words=['redraft', 'eu', 'patent', 'law', 'proposed', 'european', 'law', 'software', 'patents', 'redrafted', 'eur
         opean', 'commission', 'ec', 'despite', 'requests', 'meps', 'law', 'proving', 'controversial', 'limbo', 'year', 'major', 'tech',
         'firms', 'say', 'needed', 'protect', 'inventions', 'others', 'fear', 'hurt', 'smaller', 'tech', 'firms', 'ec', 'says', 'counci
         l', 'ministers', 'adopt', 'draft', 'version', 'agreed', 'upon', 'last', 'may', 'said', 'would', 'review', 'aspects', 'directiv
         e', 'directive', 'intended', 'offer', 'patent', 'protection', 'inventions', 'use', 'software', 'achieve', 'effect', 'words', 'c
         omputer', 'implemented', 'invention', 'letter', 'ec', 'president', 'jos233', 'manuel', 'barroso', 'told', 'president', 'europea
         n', 'parliament', 'josep', 'borrell', 'commission', 'intend', 'refer', 'new', 'proposal', 'parliament', 'council', 'ministers',
         'supported', 'agreement', 'reached', 'ministers', 'may', '2004', 'european', 'council', 'agrees', 'draft', 'directive', 'retur
         n', 'second', 'reading', 'european', 'parliament', 'guarantee', 'directive', 'become', 'law', 'instead', 'probably', 'mean', 'd
         elays', 'controversy', 'directive', 'eu', 'legislation', 'needs', 'approval', 'parliament', 'council', 'ministers', 'becomes',
         'law', 'french', 'green', 'mep', 'alain', 'lipietz', 'warned', 'two', 'weeks', 'ago', 'commission', 'ignored', 'parliament', 'r
         equest', 'would', 'insult', 'assembly', 'said', 'parliament', 'would', 'reject', 'council', 'version', 'legislation', 'part',
         'final', 'conciliation', 'stage', 'decision', 'procedure', 'us', 'patenting', 'computer', 'programs', 'internet', 'business',
         'methods', 'permitted', 'means', 'usbased', 'amazoncom', 'holds', 'patent', 'oneclick', 'shopping', 'service', 'example', 'crit
         ics', 'concerned', 'directive', 'could', 'lead', 'similar', 'model', 'happening', 'europe', 'fear', 'could', 'hurt', 'small',
         'software', 'developers', 'legal', 'financial', 'might', 'larger', 'companies', 'fight', 'patent', 'legal', 'action', 'court',
         'supporters', 'say', 'current', 'laws', 'inefficient', 'would', 'serve', 'even', 'playing', 'field', 'without', 'bringing', 'e
         u', 'laws', 'line', 'us'], tags=['Train_0'])

In [53]: x_test=label_sentences(x_test,'Test')

In [54]: x_test[0]

Out[54]: TaggedDocument(words=['fockers', 'fuel', 'festive', 'film', 'chart', 'comedy', 'meet', 'fockers', 'topped', 'festive', 'box',
         'office', 'north', 'america', 'setting', 'new', 'record', 'christmas', 'day', 'sequel', 'took', '447m', '252m', '24', '26', 'de
```


Home Page - Select or create a
project - Jupyter Notebook
localhost:8888/notebooks/project.ipynb
Apps
Gmail
YouTube
Maps
jupyter project Last Checkpoint: Last Tuesday at 8:32 AM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Type here to search
ENG
20:47
03-07-2020

```
In [53]: x_test=label_sentences(x_test,'Test')

In [54]: x_test[0]

Out[54]: TaggedDocument(words=['fockers', 'fuel', 'festive', 'film', 'chart', 'comedy', 'meet', 'fockers', 'topped', 'festive', 'box',
         'office', 'north', 'america', 'setting', 'new', 'record', 'christmas', 'day', 'sequel', 'took', '447m', '232m', '24', '26', 'de
         cember', 'according', 'studio', 'estimates', 'took', '191m', '99m', 'christmas', 'day', 'alone', 'highest', 'takings', 'day',
         'box', 'office', 'history', 'meet', 'fockers', 'sequel', 'ben', 'stiller', 'comedy', 'meet', 'parents', 'also', 'starring', 'ro
         bert', 'de', 'niro', 'blythe', 'danner', 'dustin', 'hoffman', 'barbra', 'streisand', 'despite', 'success', 'meet', 'fockers',
         'takings', '265', '2003', 'figures', 'blamed', 'christmas', 'falling', 'weekend', 'year', 'christmas', 'falls', 'weekend', 'ba
         d', 'business', 'said', 'paul', 'dergarabedian', 'president', 'exhibitor', 'relations', 'compiles', 'box', 'office', 'statistic
         s', 'weekend', 'top', '12', 'films', 'took', 'estimated', '1219m', '633m', 'compared', '1658m', '861m', 'last', 'year', 'thir
         d', 'lord', 'rings', 'film', 'dominated', 'box', 'office', 'meet', 'fockers', 'knocked', 'last', 'week', 'top', 'film', 'lemon
         y', 'snicket', 'series', 'unfortunate', 'events', 'third', 'place', '125m', '65m', 'comedy', 'fat', 'albert', 'cowritten', 'bil
         l', 'cosby', 'entered', 'chart', 'second', 'place', 'opening', 'christmas', 'day', 'taking', '127m', '66m', 'aviator', 'starrin
         g', 'leonardo', 'dicaprio', 'howard', 'hughes', 'took', '94m', 'expanding', '40', '1', '796', 'cinemas', 'christmas', 'day'], t
         ags=['Test_0'])

In [55]: all_data=x_train+x_test

In [56]: from sklearn import utils
         model_dbow=Doc2Vec(dm=0,vector_size=300,negative=5,min_count=1,alpha=0.065,min_alpha=0.065)
         model_dbow.build_vocab([x for x in all_data])
         for epoch in range(30):
             model_dbow.train(utils.shuffle([x for x in all_data]),total_examples=len(all_data),epochs=1)
             model_dbow.alpha -=0.002
             model_dbow.min_alpha=model_dbow.alpha

In [57]: import numpy as np
         def get_vectors(model,corpus_size,vectors_size,vectors_type):
             vectors=np.zeros((corpus_size,vectors_size))
```


Home Page - Select or create a
project - Jupyter Notebook
localhost:8888/notebooks/project.ipynb
Apps
Gmail
YouTube
Maps
jupyter project Last Checkpoint: Last Tuesday at 8:32 AM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Type here to search
ENG
20:49
03-07-2020

```
In [57]: import numpy as np
         def get_vectors(model,corpus_size,vectors_size,vectors_type):
             vectors=np.zeros((corpus_size,vectors_size))
             for i in range(0,corpus_size):
                 prefix=vectors_type+'_'+str(i)
                 vectors[i]=model.docvecs[prefix]
             return vectors

In [58]: train_vectors_dbow=get_vectors(model_dbow,len(x_train),300,'Train')
         test_vectors_dbow=get_vectors(model_dbow,len(x_test),300,'Test')

In [59]: train_vectors_dbow.shape

Out[59]: (1192, 300)

In [60]: train_vectors_dbow[:5]

Out[60]: array([[ 1.75624442,  0.26728043, -1.44853771, ..., -1.98173197,
                  1.00209665,  0.15048426],
                [-0.26636761, -1.09080899, -0.03420266, ..., -2.77151394,
                 -0.28721833, -0.4839749 ],
                [ 0.97203434,  1.51141155, -0.57871532, ..., -1.65953052,
                 -1.32572436,  0.34760341],
                [ 0.29134613, -0.74144828,  0.58155394, ..., -0.12164938,
                  1.15959263, -0.87997174],
                [-0.91133195,  0.79023123, -0.89952135, ...,  1.46916008,
                 -0.91737539, -0.13193056]])
```


Home Page - Select or create a
project - Jupyter Notebook
localhost:8686/notebooks/project.ipynb
Apps
Gmail
YouTube
Maps
jupyter
project Last Checkpoint: Last Tuesday at 8:32 AM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Python 3
Run
Code
Type here to search
ENG
20:50
03-07-2020

```
In [61]: model=LogisticRegression()
         model=model.fit(train_vectors_dbow,y_train)
         model_prediction=model.predict(test_vectors_dbow)
         print('accuracy %s' % accuracy_score(model_prediction,y_test))
         print(classification_report(y_test,model_prediction))
```

```
accuracy 0.959731543624161
              precision    recall  f1-score   support

           0       0.97      0.94      0.95        64
           1       0.94      1.00      0.97        63
           2       0.94      0.92      0.93        53
           3       0.98      0.95      0.97        65
           4       0.96      0.98      0.97        53

    accuracy                           0.96       298
   macro avg       0.96      0.96      0.96       298
weighted avg       0.96      0.96      0.96       298

C:\Users\HP\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed to converge (st
atus=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

CONCLUSION:

There are many other methods to classify the newspaper articles. We find that whatever model we use, the accuracy is never 100%. However the machine learning process is getting better and better. Today it is used in highly sophisticated calculations. The work of classification can be used in various fields such as movie reviews, sentiment analysis, topic detection. Machine learning is also used in medical field- such as detection of malignant tumor detection, etc.

Reference :

1. http://www.iosrjournals.org/iosr-jce/papers/Vol18-issue1/Version-3/D018132226.pdf
2. https://www.ijser.in/archives/v5i2/IJSER151243.pdf
3. https://towardsdatascience.com/text-classification-in-python-dd95d264c802
4. https://github.com/DiveshRKubal/Data-Science-Use-Cases/tree/master/News%20Classification

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE (AUTONOMOUS)
RAMAKRISHNA MISSION ASHRAMA
NARENDRAPUR, KOLKATA-700 103, W. B., INDIA
(A Branch Centre of Ramakrishna Mission, Belur Math, Howrah-711202)

Project Completion Certificate

Date: 03/06/2020

This is to certify that **Sri Rik Dutta** (ROLL NO: CSUG/158/17) a student of Department of Computer Science, Ramakrishna Mission Residential College(Autonomous), has undergone a Project work from January 15, 2020 to May 30, 2020 titled "**Predicting stock market prices with LSTM and GRU**".

Department of Computer Science
Ramakrisnha Mission Residential College
(Autonomous)
Narendrapur, Kolkata-700 103

Dr. Siddhartha Banerjee
(Head of the Department)
Department of Computer Science

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE(AUTONOMOUS),NARENDRAPUR

Predicting stock market prices with LSTM and GRU

Rik Dutta

CSUG/158/17

Department of Computer Science

2020

Predicting stock market prices with LSTM and GRU

Rik Dutta

Ramakrishna Mission Residential College (Autonomous), Narendrapur, Kolkata - 700103, West Bengal, India.

Abstract- The most interesting and yielding part of modern computer science-oriented development is the concept of artificial intelligence in a way that it almost mimics the ways our brains function. In this paper the predictive capabilities of such a concept of AI has been used to generate a predictive set of values of a certain feature of a certain dataset. The algorithm uses a perceptron-based model to predict values. LSTMs are based on a modified version of Recurrent Neural Networks (RNN). This paper drops the need for using the basic if-else construct to find what feature of scalability will give the best result for our experiment. This paper also compares the effect of scaled down samples against non-scaled sample value for LSTMs and GRU in the format of graphs fir visualization and mean squared errors for metric comparisons. This is a time-series forecasting problem which is also a statistical tool. The event outcome seems to be dependent on a variety of factors but in the dataset used for this paper the outcome generally improved with increased randomness, hence our apparent problem on time series analysis merely becomes a problem of regression analysis.

Keywords- LSTM, GRU, time-series, prediction, AI.

I. INTRODUCTION

Neuro computing has come a long way since its inception in its simplest form as a perception. To put it lightly is to say that responds in a way that animals do. Humans think of an instance of anything maybe a lot of instances but we assign some different instances to them.

An example of such an activity would be to remember a group of people. Any arbitrary group, you may know them you may not. Chances are to create such a diverse group of two types of people you have taken types, your relatives and strangers. Now an activity would be that if you were given a chance of giving someone from amongst that group a million dollars and told to cite your preferences in order of who would most likely receive this gift from you. Most likely it would be your parents or your siblings and then a close relative and at the end of that list you would place a complete and random stranger. In a way all animals assign weights to certain things in life. Long Short Term Memory and Gated Recurrent Units do exactly that.

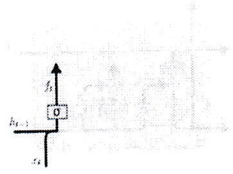
II. Problem Statement

The recurrent neural network has its own problem. It is called the vanishing gradient problem where weight of each sample is not modified or changed enough that makes the neural network more susceptible to errors in prediction. To overcome this problem, we use LSTM and GRU.

With a few minor tweaks these modifications upon RNN give surprisingly accurate results. As the weights of the samples become smaller and smaller for earlier layers of the neural network the learning rate is lowered. The vanishing gradient problem can be countered by using different neural network architectures. Two of these architectures are LSTM and GRU.

Most important LSTM formulae are as follows:

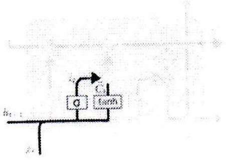
1) Deciding what information to pass through the cell state and what we give out. Denoted by the sigmoid layer.



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

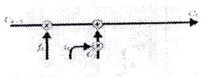
h_{t-1} denotes cell output from the previous state. W_f denotes the weight assigned to that set of previous h_{t-1} and x_t which is the input for the present state or the sample value in this state.

2) b_i is just the bias associated it helps keep the gradient always robust and save it from the vanishing gradient pitfall. This has two parts. First, a sigmoid layer called the "input gate layer" decides which values we'll update. Next, a 'tanh' layer creates a vector of new candidate values, C_t, that could be added to the state. In the next step, we'll combine these two to create an update to the state.



$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

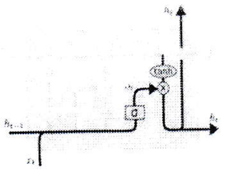
3) We multiply the old state by f_t, forgetting the things we decided to forget earlier. Then we add i_t*C_t. This is the new candidate values, scaled by how much we decided to update each state value.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

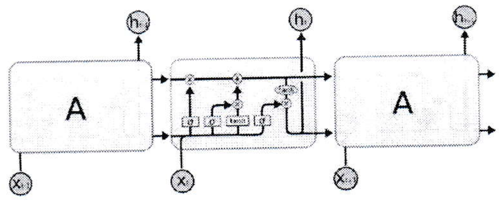
4) The output is decided by what we want to show. First, we run a sigmoid layer which decides what parts of the cell state we're going to output. Then, we put the cell state through tanh (to

push the values to be between −1 and 1) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.

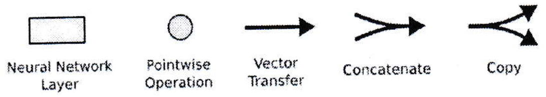


$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

The complete cell would be as-



By symbols:



III. Problem formulation

Gates are the flow of control that regulate how much of what value to send to the next step. This helps the algorithm decide how much informational value is passed on.

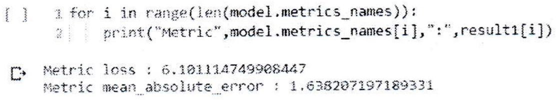
The gate functions used in LSTMs are of to types- tanh and sigmoid. The tanh layer regulates the gradient problem. When multiple values are operated as done above the multiplicative result will increase in value while other values in that input sample will not be so great. Hence only the vales with higher numerical points will be taken into account (W_f). Hence the learning rate of this algorithm diminishes. This is called the vanishing gradient problem. LSTM uses the tanh and sigmoid layers to regulate the values. The tanh will reduce the values to a threshold between -1 and 1. This is a number compression technique. Other versions of LSTM use certain other compression technique. Even logarithmic or inverse exponential functions can be used to reduce the values of multiplied values with weights. The sigmoid layer only filters what value to keep. For any value 'x' the sigmoid layer multiplies 'x' to a csonstant that regulates its actual importance in the following manner:

a) multiplied with 1 if we want to completely keep the value
b) multiplied with 0 if we throw that value away.

Hence we use the sigmoid layer as specified in accordance to the need to use the value of that input vector. Weights are generally used at random in the initial stages as the model tries to adjust these weights or constant values. But as errors seem to pile up a variety of techniques are used to reduce the errors and settle on a definitive set of weights as a result of which the back propagation shows its colours and gives a definite set of weights necessary for prediction. The weights will be multiplied for making changes to cell state. This cell state is generally hidden, which means that the value will be thrown off and this value has a low weight multiplied to it.

III. Python implementation

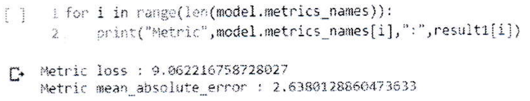
Output given by the code is as follows:



```
[ ]  for i in range(len(model.metrics_names)):
         print("Metric",model.metrics_names[i],":",result1[i])

     Metric loss : 6.101114749908447
     Metric mean_absolute_error : 1.638207197189331
```

Figure 1- metrics for non-randomized samples.

Using a randomized selection technique to select a set of samples we see the metric loss and mean absolute error slightly increase in magnitude (Figure 2):



```
[ ]  for i in range(len(model.metrics_names)):
         print("Metric",model.metrics_names[i],":",result1[i])

     Metric loss : 9.062216758728027
     Metric mean_absolute_error : 2.6380128860473633
```

Figure 2-metrics for randomized samples.

This happens usually in this case. Although it looks like a regression problem at first glance, it is a time series problem. The trend in previous sales prices affects the trend towards future prices the graphs included below will explain further.

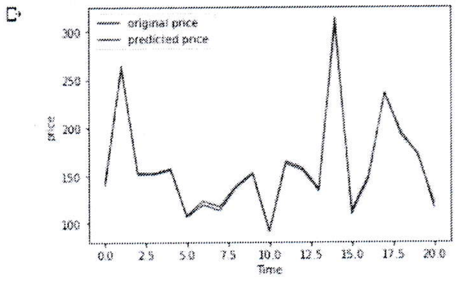


Figure 3-shows the effectiveness in our prediction but due to the random nature of selecting the sample values the long-term predictions will not be very effective

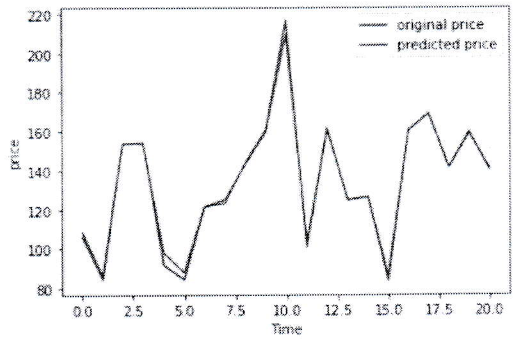
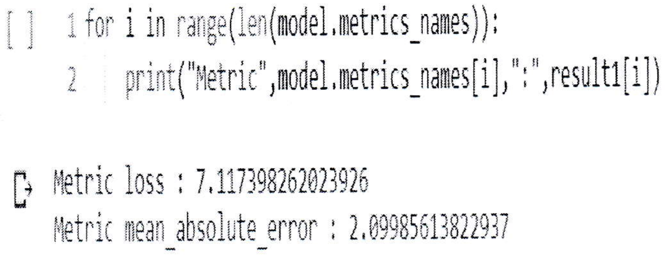


Figure 4- shows that some higher and lower peaks are slightly faulty, but this method is more effective for long term prediction of time series modelling.

IV. Difference between LSTM and GRU

Gated Recurrent Units remove the usage of the cell state part instead use the hidden state to transfer information. This hidden state transfers information as well as is used for prediction. Hence, GRU cell is more simple to understand than LSTM cell. It only has two gates namely reset and update gates. Hence LSTM help us set more hyperparameters that make the development more fluid and helps us fix values better.

The metrics are as follows using GRU-



```
[ ]  for i in range(len(model.metrics_names)):
         print("Metric",model.metrics_names[i],":",result1[i])

     Metric loss : 7.117398262023926
     Metric mean_absolute_error : 2.09985613822937
```

Figure 5-GRU metrics parameter values

The output prediction graph for GRU is :

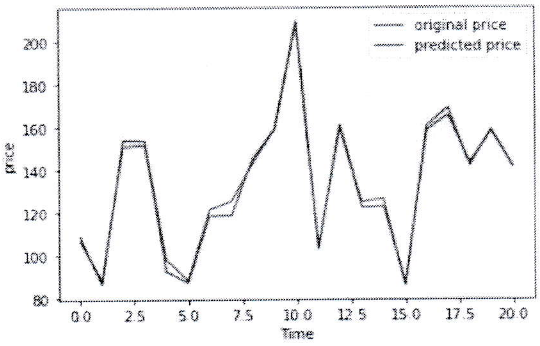


Figure 6-GRU actual prices versus prediction values

It is only natural that since GRU have less number of gates that also compute faster than LSTM. But it is only a matter of trial and error about which model is better. Average time elapsed between computations of the two models is:

LSTM	GRU
403 seconds for 200 epochs with batch size 16 and an average 932us/step of evaluation.	600 seconds for 200 epochs with batch size 16 and an average 1ms/step of evaluation.

The visible differences are not that substantial to classify either as the better.

The figure next shows the difference in predictions:

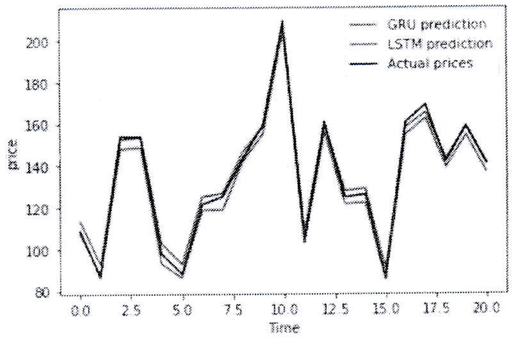


Figure 7-Complete comparison graph Actual price vs. LSTM prediction vs. GRU prediction.

The differences are not that substantial although a few minor tweaks have been shown with higher epoch numbers:

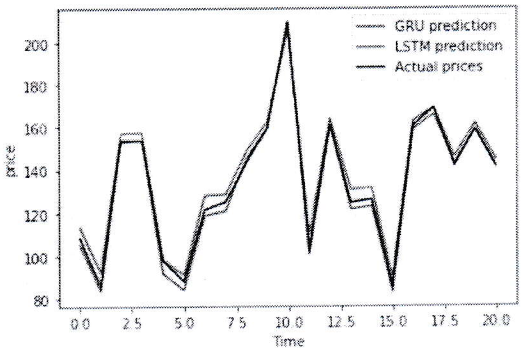


Figure 8-For 300 epochs the prediction comparison stands at GRUs being better than LSTMs

As visible there has been a slight improvement in predictions but that is natural when we increase the epoch numbers, moreover the GRU architecture tends to give better outputs in this case. This fallacy in the neural architecture might cause disturbance for noisy data when we have to try more combinations of batch and epoch numbers for finding the correct and satisfying results, as shown below:

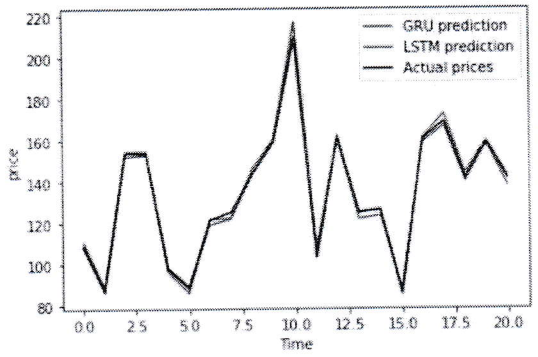
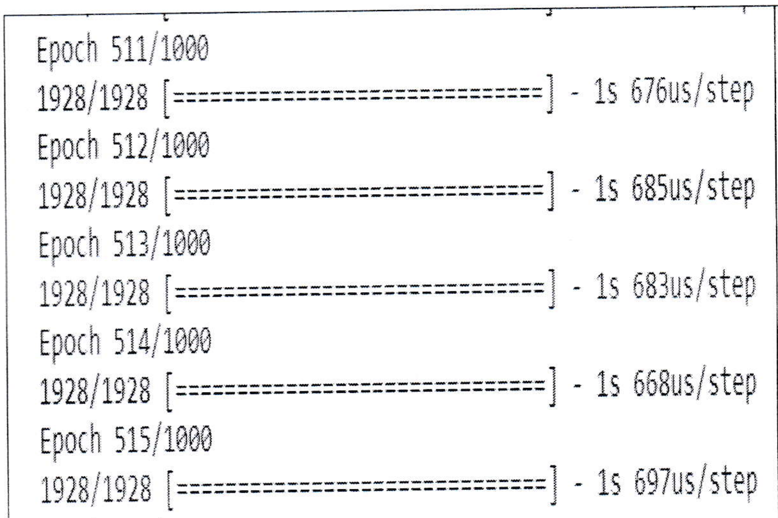


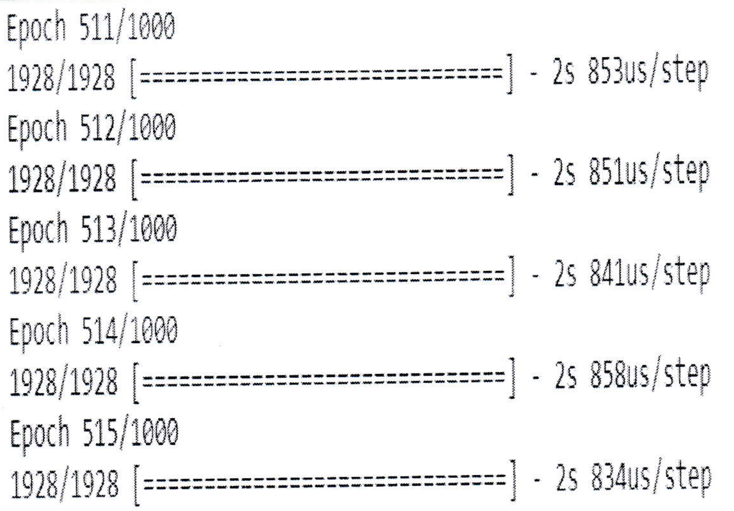
Figure 9-LSTM PREDICTION vs GRU PREDICTON vs actual prices for 1000 epochs and 32 batches each. We can see a marked improvement over the previous ones where we used lower values for hyperparameters.

Following shows the time needed to execute each:



```
Epoch 511/1000
1928/1928 [==============================] - 1s 676us/step
Epoch 512/1000
1928/1928 [==============================] - 1s 685us/step
Epoch 513/1000
1928/1928 [==============================] - 1s 683us/step
Epoch 514/1000
1928/1928 [==============================] - 1s 668us/step
Epoch 515/1000
1928/1928 [==============================] - 1s 697us/step
```

Figure 10-AVERAGE TIME OF LSTM EXECUTION FOR EACH EPOCH.



```
Epoch 511/1000
1928/1928 [==============================] - 2s 853us/step
Epoch 512/1000
1928/1928 [==============================] - 2s 851us/step
Epoch 513/1000
1928/1928 [==============================] - 2s 841us/step
Epoch 514/1000
1928/1928 [==============================] - 2s 858us/step
Epoch 515/1000
1928/1928 [==============================] - 2s 834us/step
```

Figure 11-AVERAGE TIME OF GRU EXECUTION FOR EACH EPOCH.

V. **Conclusion**

There is much difference between the execution times of GRU and LSTM. As far as time-series are concerned, hence the time saving capabilities of LSTM are far more advantageous than the less computations required in GRUs.

Notebook available at google colab:
https://drive.google.com/file/d/1mzyx-oRVhpcF90UaHWmoQk5RuusX-g3Fs/view?usp=sharing

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE (AUTONOMOUS)
RAMAKRISHNA MISSION ASHRAMA
NARENDRAPUR, KOLKATA-700 103, W. B., INDIA
(A Branch Centre of Ramakrishna Mission, Belur Math, Howrah-711202)

Project Completion Certificate

Date: 03/06/2020

This is to certify that **Sri Sabyasachi Chatterjee** (ROLL NO: CSUG/007/17) a student of Department of Computer Science, Ramakrishna Mission Residential College(Autonomous), has undergone a Project work from January 15, 2020 to May 30, 2020 titled "**Optical Character Recognition**".

Department of Computer Science
Ramakrishna Mission Residential College
(Autonomous)
Narendrapur, Kolkata-700 103

Dr. Siddhartha Banerjee
(Head of the Department)
Department of Computer Science

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE (AUTONOMOUS)
RAMAKRISHNA MISSION ASHRAMA
NARENDRAPUR, KOLKATA-700 103, W. B., INDIA
(A Branch Centre of Ramakrishna Mission, Belur Math, Howrah-711202)

Project Completion Certificate

Date: 03/06/2020

This is to certify that **Sri Sourjya Chatterjee** (ROLL NO: CSUG/185/17) a student of Department of Computer Science, Ramakrishna Mission Residential College(Autonomous), has undergone a Project work from January 15, 2020 to May 30, 2020 titled **"Optical Character Recognition".**

Department of Computer Sc
Ramakrishna Mission Residen
(Autonomous)
Narendrapur, Kolkata-700 1

Dr. Siddhartha Banerjee
(Head of the Department)
Department of Computer Science

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE(AUTONOMOUS),NARENDRAPUR

OPTICAL CHARACTER RECOGNITION

NAME : Sobyasachi Chatterjee

Roll no : csug/007/17

NAME : Sourjya Chatterjee

Roll no : csug/185/17

Department of Computer Science

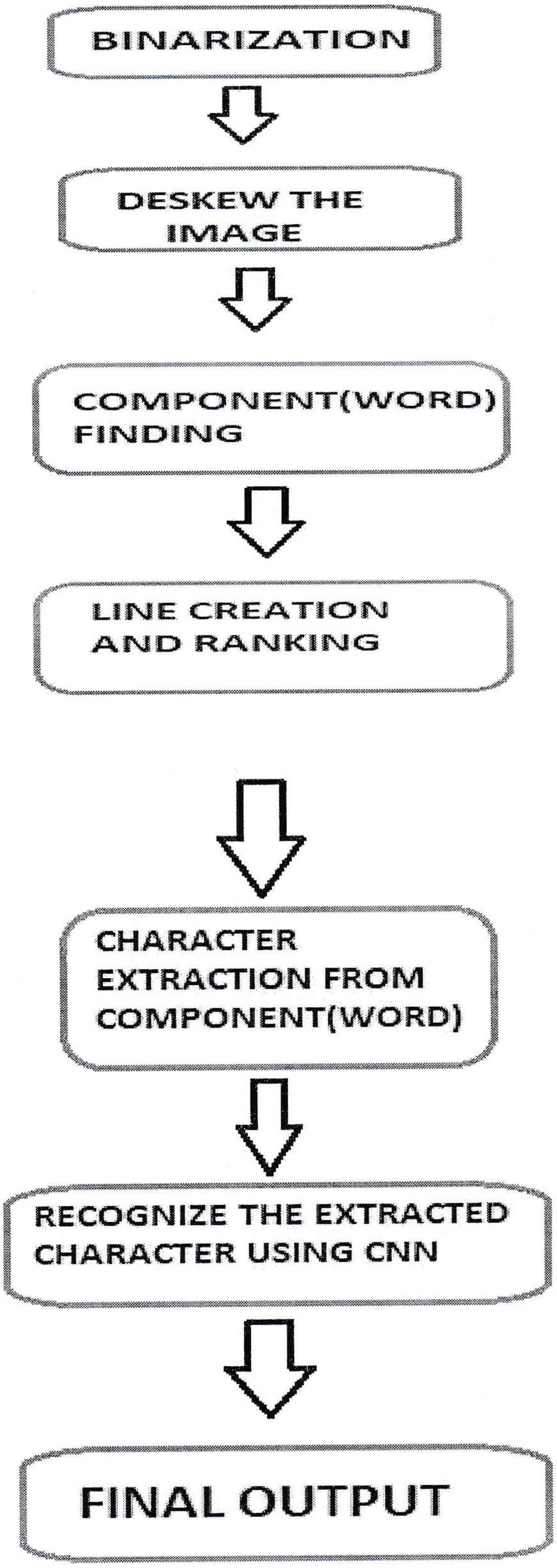
2020

OPTICAL CHARACTER RECOGNITION

Abstraction:- Machine replication of human functions, like reading, is an ancient dream. However, over the last five decades, machine reading has grown from a dream to reality. Optical character recognition has become one of the most successful applications of technology in the field of pattern recognition and artificial intelligence. Optical character recognition (OCR) is one of the latest technologies adopted in a lot of areas such as management, business, criminal and social networks. It consists of recognizing image-based characters and transforming them to real digital character that can be editing, written and displayed. The **aim** of this project is to recognize an **English** text from a photo like JPG,JEPG,PNG .

Step Wise Methodology:- This project is done in several steps. We try to solve this practical problem very carefully and we experienced too many real life problems to solve this project which aren't found generally in theoretical world.

We use **Python** Programming Language in ***Jupyter lab*** environment to implement our idea. We use some **Image Processing** concepts for preprocessng the captured Image and also use **Convolutional Neural Network** to recognize the characters from the captured Image.


BINARIZATION
DESKEW THE IMAGE
COMPONENT(WORD) FINDING
LINE CREATION AND RANKING
CHARACTER EXTRACTION FROM COMPONENT(WORD)
RECOGNIZE THE EXTRACTED CHARACTER USING CNN
FINAL OUTPUT

Binarization:- In this step we covert the image to gray then binarise that.We divide the image into various matrices .

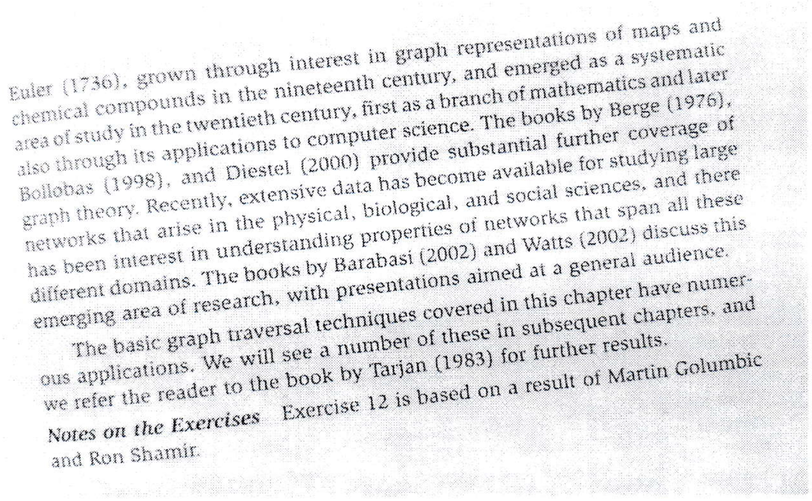
We use OTSU to binarise each of the divided matrices

Problem comes when the divided part has no part of foreground image (text written)

So we use neural network on number of black points and number of black points along the edges of matrices

So we whiten that divided matrix when tested by neural network returns 0.

INPUT:

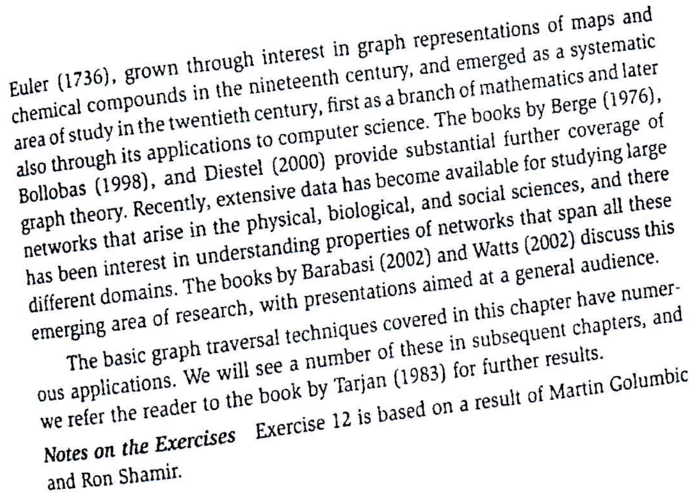


Euler (1736), grown through interest in graph representations of maps and chemical compounds in the nineteenth century, and emerged as a systematic area of study in the twentieth century, first as a branch of mathematics and later also through its applications to computer science. The books by Berge (1976), Bollobas (1998), and Diestel (2000) provide substantial further coverage of graph theory. Recently, extensive data has become available for studying large networks that arise in the physical, biological, and social sciences, and there has been interest in understanding properties of networks that span all these different domains. The books by Barabasi (2002) and Watts (2002) discuss this emerging area of research, with presentations aimed at a general audience.

The basic graph traversal techniques covered in this chapter have numerous applications. We will see a number of these in subsequent chapters, and we refer the reader to the book by Tarjan (1983) for further results.

Notes on the Exercises Exercise 12 is based on a result of Martin Golumbic and Ron Shamir.

OUTPUT:

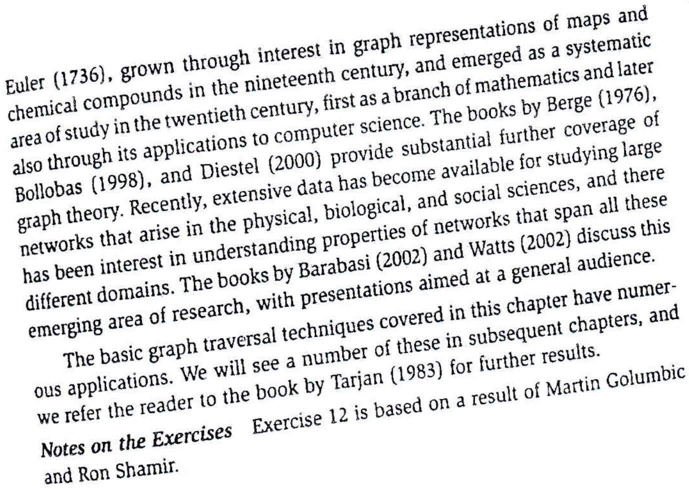


Euler (1736), grown through interest in graph representations of maps and chemical compounds in the nineteenth century, and emerged as a systematic area of study in the twentieth century, first as a branch of mathematics and later also through its applications to computer science. The books by Berge (1976), Bollobas (1998), and Diestel (2000) provide substantial further coverage of graph theory. Recently, extensive data has become available for studying large networks that arise in the physical, biological, and social sciences, and there has been interest in understanding properties of networks that span all these different domains. The books by Barabasi (2002) and Watts (2002) discuss this emerging area of research, with presentations aimed at a general audience.

The basic graph traversal techniques covered in this chapter have numerous applications. We will see a number of these in subsequent chapters, and we refer the reader to the book by Tarjan (1983) for further results.

Notes on the Exercises Exercise 12 is based on a result of Martin Golumbic and Ron Shamir.

Deskew The Image:- text skew correction algorithms is too much essential in the field of automatic document analysis. Many times Captured text images are skewd with some angel. It's a major problem for document analysis. Here is an **INPUT** example we've used which is binarised already.



Euler (1736), grown through interest in graph representations of maps and
chemical compounds in the nineteenth century, and emerged as a systematic
area of study in the twentieth century, first as a branch of mathematics and later
also through its applications to computer science. The books by Berge (1976),
Bollobas (1998), and Diestel (2000) provide substantial further coverage of
graph theory. Recently, extensive data has become available for studying large
networks that arise in the physical, biological, and social sciences, and there
has been interest in understanding properties of networks that span all these
different domains. The books by Barabasi (2002) and Watts (2002) discuss this
emerging area of research, with presentations aimed at a general audience.

The basic graph traversal techniques covered in this chapter have numer-
ous applications. We will see a number of these in subsequent chapters, and
we refer the reader to the book by Tarjan (1983) for further results.

Notes on the Exercises Exercise 12 is based on a result of Martin Golumbic
and Ron Shamir.

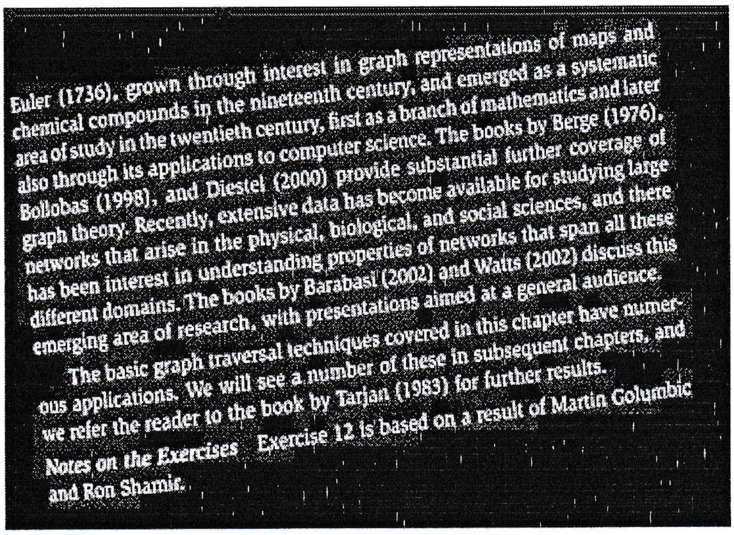
We have to follow some steps to deskew the image in a correct position.

(a) Detecting the block of text in the image.

(b)Computing the angle of the rotated text.

(c)Rotating the image to correct for the skew.

We use OpenCV to deskew the image. To apply the algorithm the text image must contain dark background and light or white text ; however, to apply our text skew correction process, we first need to invert the image (i.e., the text is now light on a dark background –– we need the inverse). So we use **bitwise_not** function of **OpenCv.** And the Image will be like that.



Euler (1736), grown through interest in graph representations of maps and chemical compounds in the nineteenth century, and emerged as a systematic area of study in the twentieth century, first as a branch of mathematics and later also through its applications to computer science. The books by Berge (1976), Bollobas (1998), and Diestel (2000) provide substantial further coverage of graph theory. Recently, extensive data has become available for studying large networks that arise in the physical, biological, and social sciences, and there has been interest in understanding properties of networks that span all these different domains. The books by Barabasi (2002) and Watts (2002) discuss this emerging area of research, with presentations aimed at a general audience.

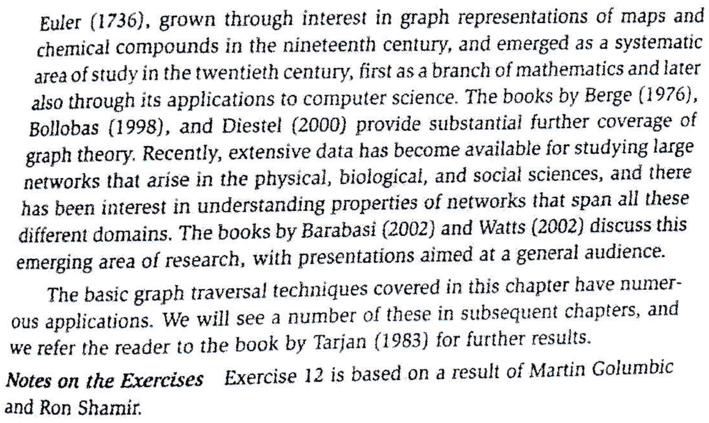
The basic graph traversal techniques covered in this chapter have numerous applications. We will see a number of these in subsequent chapters, and we refer the reader to the book by Tarjan (1983) for further results.

Notes on the Exercises Exercise 12 is based on a result of Martin Golumbic and Ron Shamir.

Then finds all (x, y)-coordinates in the image that are part of the foreground.We pass these coordinates into **cv2.minAreaRect** which then computes the minimum rotated rectangle that contains the entire text region.The **cv2.minAreaRect** function returns angle values in the range **[-90, 0).** As the rectangle is rotated clockwise the angle value increases towards zero. When zero is reached, the angle is set back to **-90** degrees again and the process continues.

If the angle is less than -**45** degrees, in which case we need to add **90** degrees to the angle and take the inverse.Now that we have determined the text skew angle, we need to apply an affine transformation to correct for the skew.

Determine the center (**x, y**)-coordinate of the image. We pass the coordinates and rotation angle into the **cv2.getRotationMatrix2D**. This rotation matrix **M** is then used to perform the actual transformation. draws the **angle** on our image so we can verify that the output image matches the rotation angle.And final output will in Dark text and White Background.
OUTPUT is:-



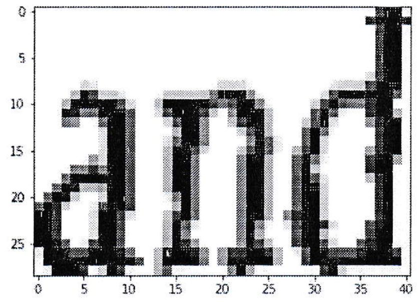
Euler (1736), grown through interest in graph representations of maps and chemical compounds in the nineteenth century, and emerged as a systematic area of study in the twentieth century, first as a branch of mathematics and later also through its applications to computer science. The books by Berge (1976), Bollobas (1998), and Diestel (2000) provide substantial further coverage of graph theory. Recently, extensive data has become available for studying large networks that arise in the physical, biological, and social sciences, and there has been interest in understanding properties of networks that span all these different domains. The books by Barabasi (2002) and Watts (2002) discuss this emerging area of research, with presentations aimed at a general audience.

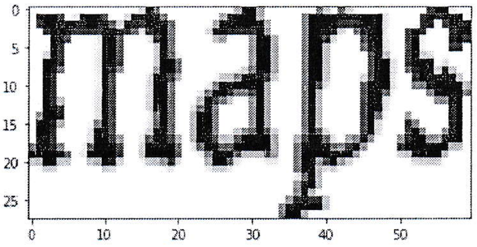
The basic graph traversal techniques covered in this chapter have numerous applications. We will see a number of these in subsequent chapters, and we refer the reader to the book by Tarjan (1983) for further results.

Notes on the Exercises Exercise 12 is based on a result of Martin Golumbic and Ron Shamir.

Component(word) Finding:- In this stage we treat every word of the text Image as a graph Connected Component. We've search the binarised image pixel wise and we found any black pixel then we searched up to 6 pixels down and right side of that pixel. If any black pixel have found then include that pixel with that component. It means we searched a 6X6 box where the upper left corner of the box is the black pixel of the existing component.

Initially the first black pixel will the starting node of the component and after required steps neighbour pixels(which are within the 6X6 box) are included in the component. By Searching this entire text image we will get all the Words. Here is the example-

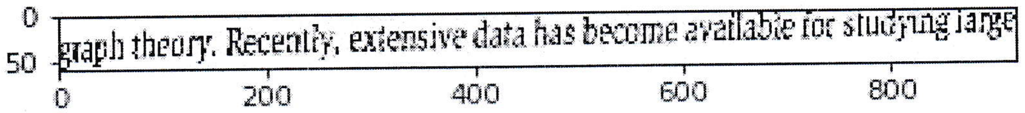




Line Creation & Ranking:- In this step we conquer the words which are in the same line with their correct position and decide the ordering of the lines.

To create the line by conquering the words we followed some logical conditions. At first we measure the possible highest height of a character of the text Image , which is denoted as CH. The process of linking two neighboring words is addressed as follows: Let (x_1,y_1) ,(x_2,y_2) denote the bounding box coordinates of an assigned word and (x_3,y_3), (x_4,y_4) denote the bounding box coordinates of a candidate neighbor word. From all words in the right side of the assigned word which satisfy the condition :

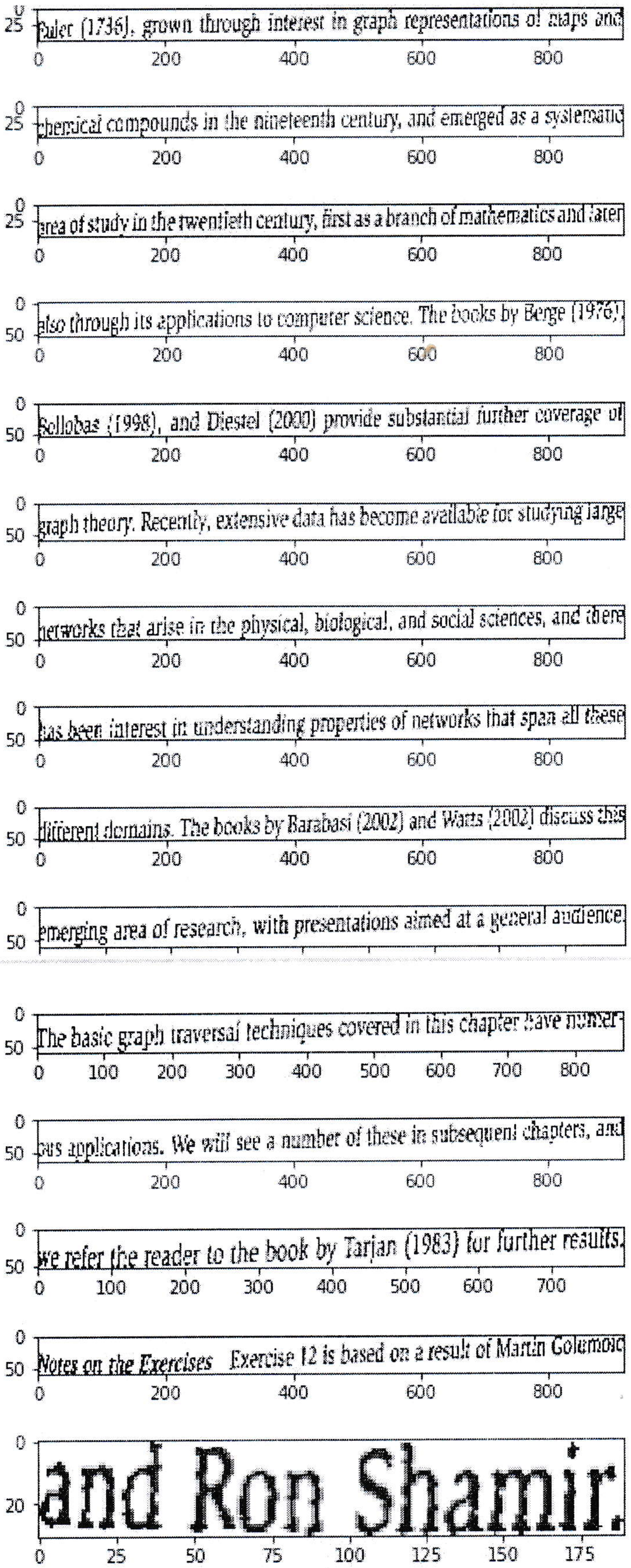
$[y_1,y_2]$ **Intersection** $[y_3,y_4]$ = **NOT NULL** (represents the horizontal overlapping), we select the one with the smaller distance $D=x_3-x_2$ only if $0<D<6*CH$ Since many words may satisfy the condition of horizontal overlapping, selecting the one with the smaller distance, we select the immediate neighbor word of the same text line. Next, this word is assigned as processed and we search in the right side for a neighbor word till the last word of the text line is assigned as processed. Here is an example of a created line:



graph theory. Recently, extensive data has become available for studying large

0
50
0
200
400
600
800

After that to ordering the lines we compare the Y values of the coordinates of any pixel of a component of a line with the other line's component's pixel. The higher Y value implies, the line is ordered as behind to the lower one. By this logic we compare all the line with each other and decide the ordering. Here is the example of our result.

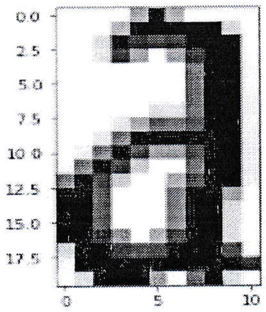


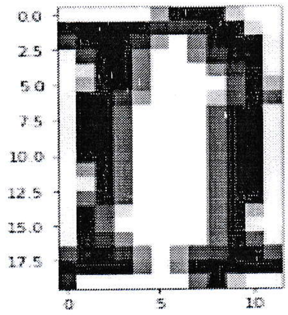
Euler (1736), grown through interest in graph representations of maps and chemical compounds in the nineteenth century, and emerged as a systematic area of study in the twentieth century, first as a branch of mathematics and later also through its applications to computer science. The books by Berge (1976), Bollobas (1998), and Diestel (2000) provide substantial further coverage of graph theory. Recently, extensive data has become available for studying large networks that arise in the physical, biological, and social sciences, and there has been interest in understanding properties of networks that span all these different domains. The books by Barabasi (2002) and Watts (2002) discuss this emerging area of research, with presentations aimed at a general audience.

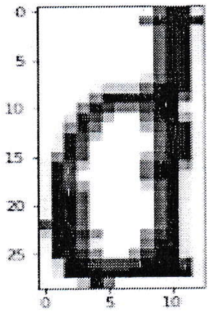
The basic graph traversal techniques covered in this chapter have numerous applications. We will see a number of these in subsequent chapters, and we refer the reader to the book by Tarjan (1983) for further results.

Notes on the Exercises Exercise 12 is based on a result of Martin Golumbic and Ron Shamir.

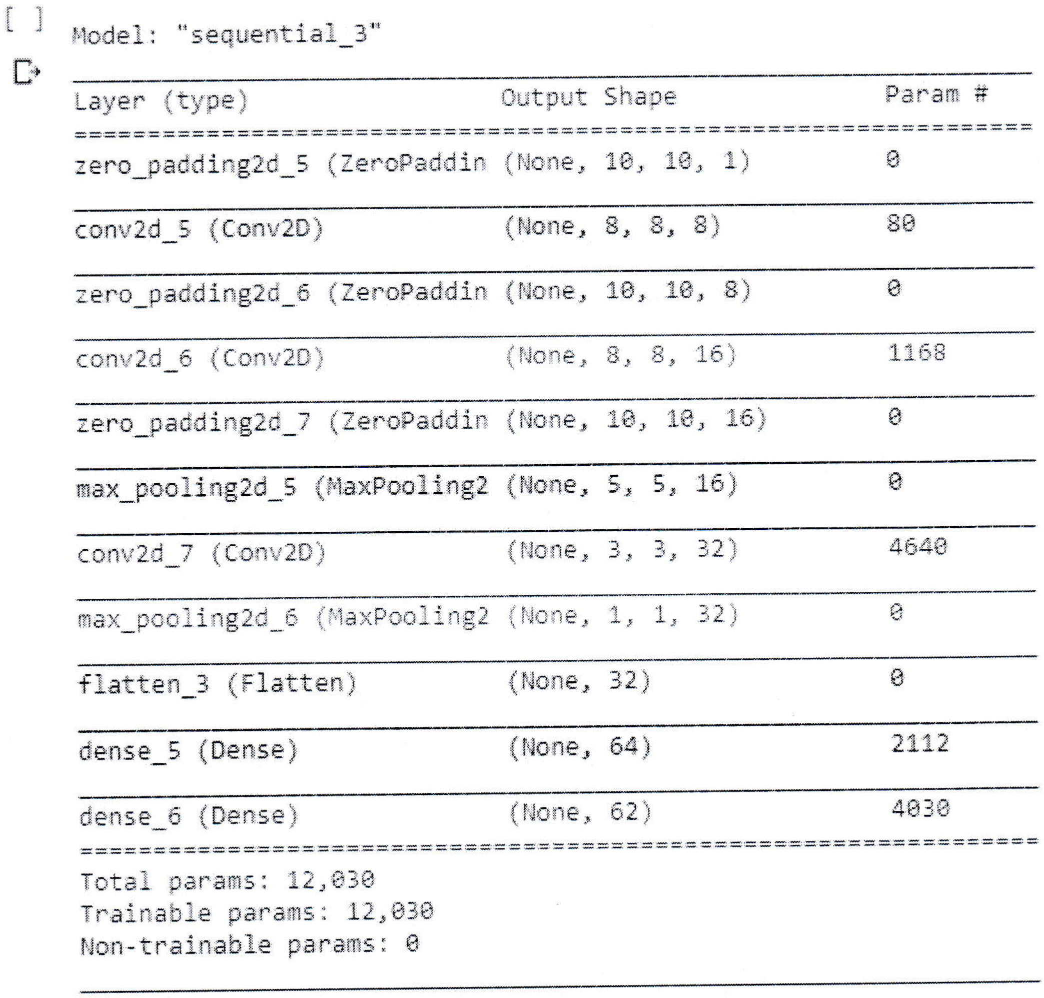
Character Extraction From Component(Word):- In this Step we extracted all the characters from their components.We follow a quite similar approach here like word finding. By taking a component we searched for a black pixel and then searched for the next black pixel which is adjacent with that pixel. Then we put them together as a graph component and searched for the remaining black pixels which are adjacent. By following this logic we extract characters from a word. And we use this process for all the words.Here is an example of extraction:





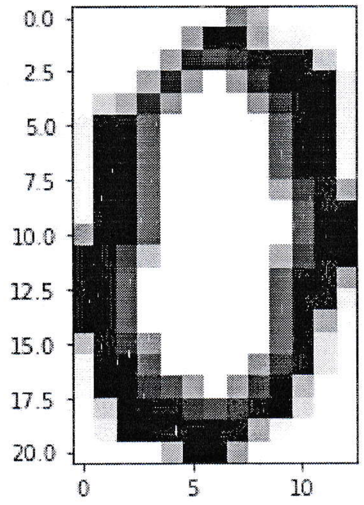


Recognize Extracted Character Using CNN:- We collect data set from **kaggle.com** for the training our CNN.Here is an overview of our CNN.

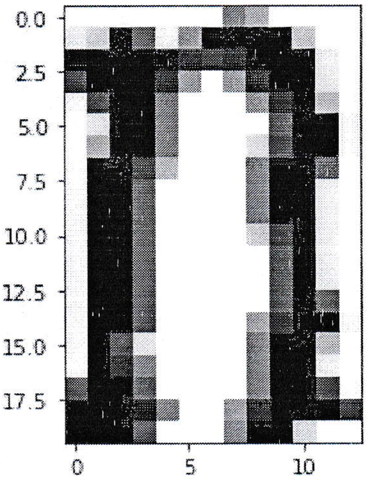


```
[ ]  Model: "sequential_3"
     _________________________________________________________________
     Layer (type)                 Output Shape              Param #
     =================================================================
     zero_padding2d_5 (ZeroPaddin (None, 10, 10, 1)         0
     _________________________________________________________________
     conv2d_5 (Conv2D)            (None, 8, 8, 8)           80
     _________________________________________________________________
     zero_padding2d_6 (ZeroPaddin (None, 10, 10, 8)         0
     _________________________________________________________________
     conv2d_6 (Conv2D)            (None, 8, 8, 16)          1168
     _________________________________________________________________
     zero_padding2d_7 (ZeroPaddin (None, 10, 10, 16)        0
     _________________________________________________________________
     max_pooling2d_5 (MaxPooling2 (None, 5, 5, 16)          0
     _________________________________________________________________
     conv2d_7 (Conv2D)            (None, 3, 3, 32)          4640
     _________________________________________________________________
     max_pooling2d_6 (MaxPooling2 (None, 1, 1, 32)          0
     _________________________________________________________________
     flatten_3 (Flatten)          (None, 32)                0
     _________________________________________________________________
     dense_5 (Dense)              (None, 64)                2112
     _________________________________________________________________
     dense_6 (Dense)              (None, 62)                4030
     =================================================================
     Total params: 12,030
     Trainable params: 12,030
     Non-trainable params: 0
     _________________________________________________________________
```

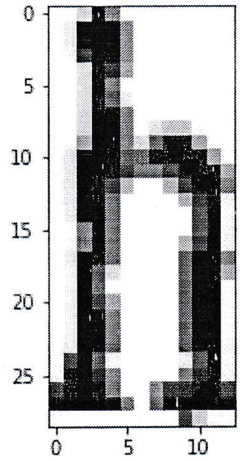
The characters of the text image are recognized.Here is some samples of characters which are recognized by our CNN.The extracted input character is shown and the corresponding output character which recognized is shown in text format at the lower left corner side of the input.


0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
5
10

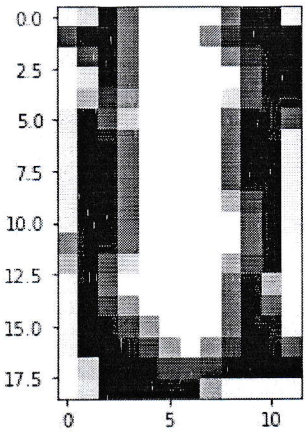
O


0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0
5
10

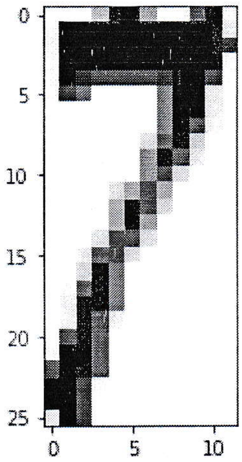
n


0
5
10
15
20
25
0
5
10

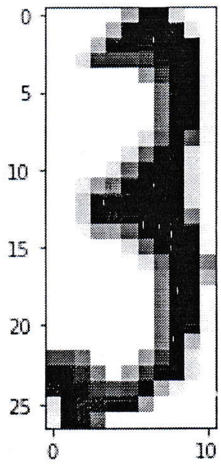
h


0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0
5
10

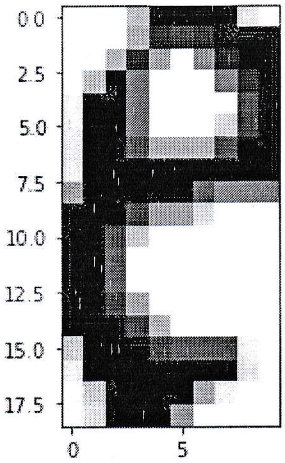
U


0
5
10
15
20
25
0
5
10

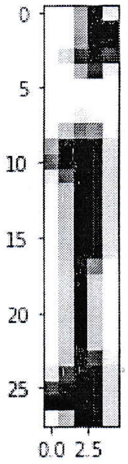
7


0
5
10
15
20
25
0
10

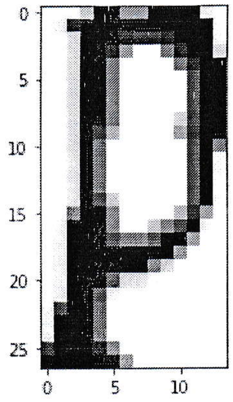
3


0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0
5

e


0
5
10
15
20
25
0.0
2.5

i


0
5
10
15
20
25
0
5
10

p

Final Output:- Having recognized characters, it just some file writer code to get out .txt file.

Problems we faced

Due to lack of training data , this step even after having 97% training set accuracy did not perform that well as we expected it be.Also we lacked datasets for comma , fullstop and other charecters.

Referrals

1. OTSU thresholding algorithm, for binarization.

2. PAPER:- Goal-Oriented Rectification of Camera-Based Document Images, by Nikolaos Stamatopoulos, Basilis Gatos, Ioannis Pratikakis, Member, IEEE, and Stavros J. Perantonis , LINK- https://www.researchgate.net/publication/46578204_Goal-Oriented_Rectification_of_Camera-Based_Document_Images

3. https://www.pyimagesearch.com/2017/02/20/text-skew-correction-opencv-python/

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE (AUTONOMOUS)
RAMAKRISHNA MISSION ASHRAMA
NARENDRAPUR, KOLKATA-700 103, W. B., INDIA
(A Branch Centre of Ramakrishna Mission, Belur Math, Howrah-711202)

Project Completion Certificate

Date: 03/06/2020

This is to certify that **Sri Souvik Bera** (ROLL NO: CSUG/037/17) a student of Department of Computer Science, Ramakrishna Mission Residential College(Autonomous), has undergone a Project work from January 15, 2020 to May 30, 2020 titled **"Hospital Service Queue Management System with Wireless approach"**.

Department of Computer Science
Ramakrisnha Mission Residential College
(Autonomous)
Narendrapur, Kolkata-700 103

Dr. Siddhartha Banerjee
(Head of the Department)
Department of Computer Science

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE(AUTONOMOUS),NARENDRAPUR

HOSPITAL SERVICE QUEUE MANAGEMENT SYSTEM WITH WIRELESS APPROACH

NAME - SOUVIK BERA

COLLEGE ROLL NO. - CSUG/037/17

REG. NO. - A03-1122-0037-17

Department of Computer Science

2020

HOSPITAL SERVICE QUEUE MANAGEMENT SYSTEM WITH WIRELESS APPROACH

Abstract. This paper presents a proposed alternative system for queuing management that could reduce inconvenience to the public. The motivation of this system is depicted from an observation on the people queuing for services in the hospitals and the government offices without committing to the estimated time for their demand. Waiting for the service is counterproductive which consumes an unacceptable amount of productive time for the patients. We develop the system to manage the queue without physically lining up and allow people to monitor their queue status by their wireless handheld devices. The project accomplishes its objective as a tool to manage the hospital queue online where customers, patients and stakeholder can access theirs queues remotely over the Internet through a web application. The results benefit to both stakeholder to manage their time for other desire activities and hospitals in utilizing its spacious area for other business proposes.

Keywords: Hospital queuing management system, web application.

1. Introduction

The innovation of technologies could bring support to the quality of life for human in various aspects and objectives. However, in order to apply and implement technology system to be used requires the costly investment for itself. This constraint leads to the inescapable archaic management methods, and the systems still coexist alongside the advances in procedures. One of the unavoidable significances is the hospital service for the people, especially among the undeveloped country and developing country. The public hospitals likely support the poor and middle classes which have to patronize the public services in the state hospitals.

A growing population base will continue having a pressure to the existing hospital facilities. With the cycle of limited facilities, it leads to the coupled staffing shortages which will guarantee that long queues to remain synonymous anytime visiting a hospital and other public service facilities. The people must take a queue as long as they need the services. Whether the problem is caused by staff shortages, equipment shortages, or the hospital capacity is not sufficient for the population area they serve. Long queues are an unwanted and unnecessary burden to the public as well as the hospital staffs. Long queues are then associated with a negative image of the hospital experience, but most people can't avoid to be under this present system.

For this project, we propose the system with the main objective as to create a visual queue for hospital online where people can access and reserve their queue wirelessly over the Internet. The system allows people to monitor their queuing status from the web service application. This beneficial system is designed to offer the options for people who are waiting for the service; they can go anywhere while they are in the queue rather than standing and presenting themselves in front of the service area.

2. Literature Review

The traditional queuing management methods mostly used in the hospital are queue card and smart queue as described it featured by figure 1.When using queue card system, the people in the queue are assigned by numbers according to the arrival order. This method allows the patients to be able to manage their time based on an estimation of the time available until their number is called. Venturing outside of the immediate area is a constant gamble. The queue number may guarantee service according to the number priorities; however, a delay in returning may still result in the loss of a queue position.

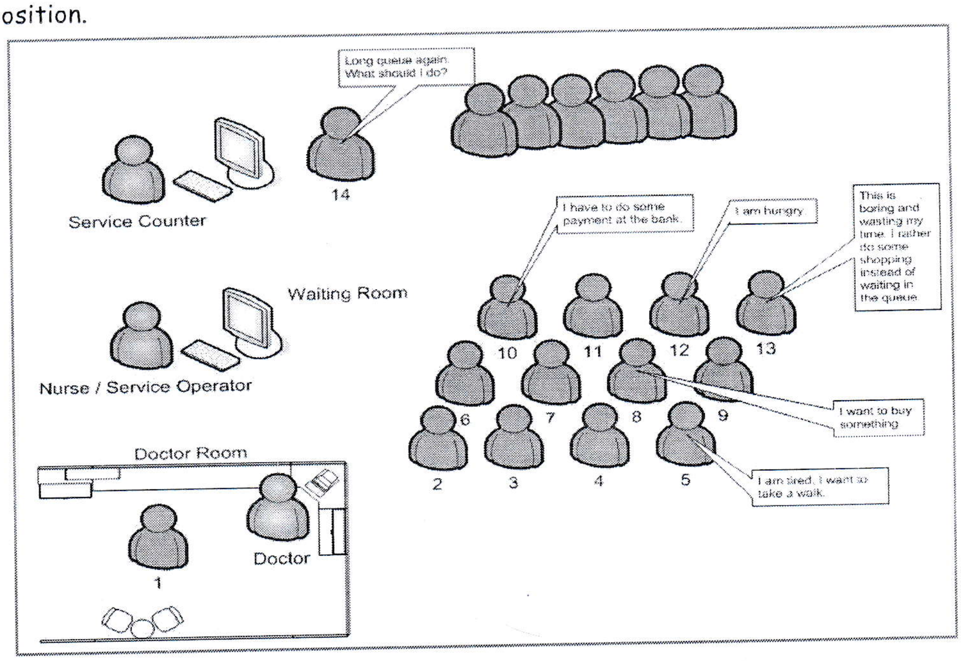

Long queue again.
What should I do?
14
Service Counter
I have to do some payment at the bank.
I am hungry.
This is boring and wasting my time. I rather do some shopping instead of waiting in the queue.
Waiting Room
10
11
12
13
Nurse / Service Operator
6
7
8
9
I want to buy something
Doctor Room
2
3
4
5
I am tired, I want to take a walk.
Doctor
1

Fig. 1. Typical state hospital queue management system

Most of the private hospitals provide a smart queue system as well as helpdesks and counter services for their customers. The smart queue system provides automatic queue numbers along with automatic voice calling and LED display panels on a progressive basis. However, this system still requires patients to congregate in the immediate area to monitor the progress of queue numbers being serviced. This service only eliminates the need to stand in an organized line, but does not address a more productive method for time utilization.

Based on a survey, people waiting in a queue get a service from public hospitals in rural area in Thailand reveal that they are compelled to endure the endless waits. They lined up at the service counter. Any abandonment results in their requirement to return to the back of the line and an even longer wait. With such a long queue and waiting period, it represents a considerable amount of time wasted for the people involved. Any desire to venture outside the immediate area is outweighed by the uncertainty of not having information regarding the progress of the queue. They simply cannot miss their position due to a lack of information. This problem motivated us to develop a method to manage the reserved queue to alleviate on minimizing the number of people in the physical queue.

3. Service Queue Management System with Wireless Approach

3.1 System Boundary and Architecture

The new approach of the hospital queue management system will provide stakeholder with tools to manage their queue status wirelessly[1]. The system would allow them to know what is going on with the queue wherever they go. As can be seen in the figure 2 a new comer arrives at the service counter before booking into the hospital queue. With their wireless devices, the queue status can be accessed through the Internet, and it provides information to everyone in the queue.

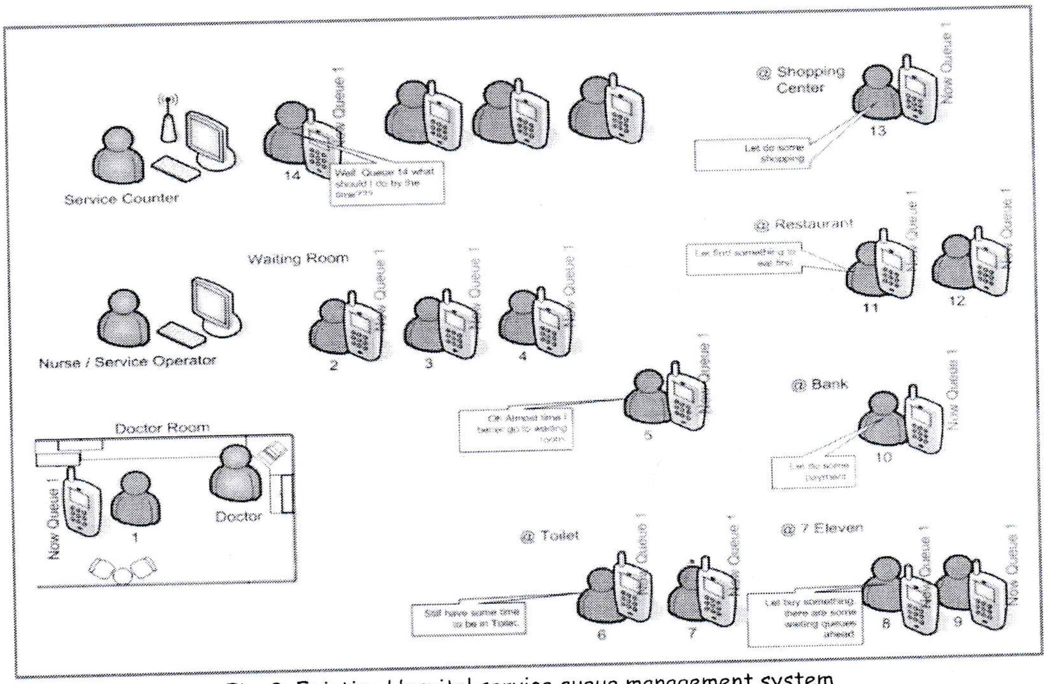

Service Counter
14
Now Queue 1
Well. Queue 14 what should I do by the time???
Waiting Room
Nurse / Service Operator
2
3
4
Doctor Room
Doctor
1
Now Queue 1
@ Shopping Center
13
Let do some shopping
@ Restaurant
Let find something to eat first
11
12
Ok. Almost time I better go to waiting room.
5
@ Bank
Let do some payment
10
@ Toilet
Still have some time to be in Toilet.
6
7
@ 7 Eleven
Let buy something, there are some waiting queues ahead
8
9

Fig. 2. Existing Hospital service queue management system

The proposed system, the boundary and its functionality are described in a form of UML concepts[1,2] shown in figure 3. The system's functionality is demonstrated and explained as the role of four actors and seven use cases as following:

Actors role:

- **System Admin:** represents an administrator who grants access to all system features; the role is to register a new hospital and queue administration to the system.
- **Queue Admin:** represents a hospital queue administrator, the role is to create queues and operators to the system.
- **Queue Operator:** represents a person who takes care of each queue. The role is to register queue client to the system and to manage all activities in the queue.
- **Queue Client:** represents the person who requires hospital service and is seated in the queue. The role is to view the queue status in order to know when to be in the service.

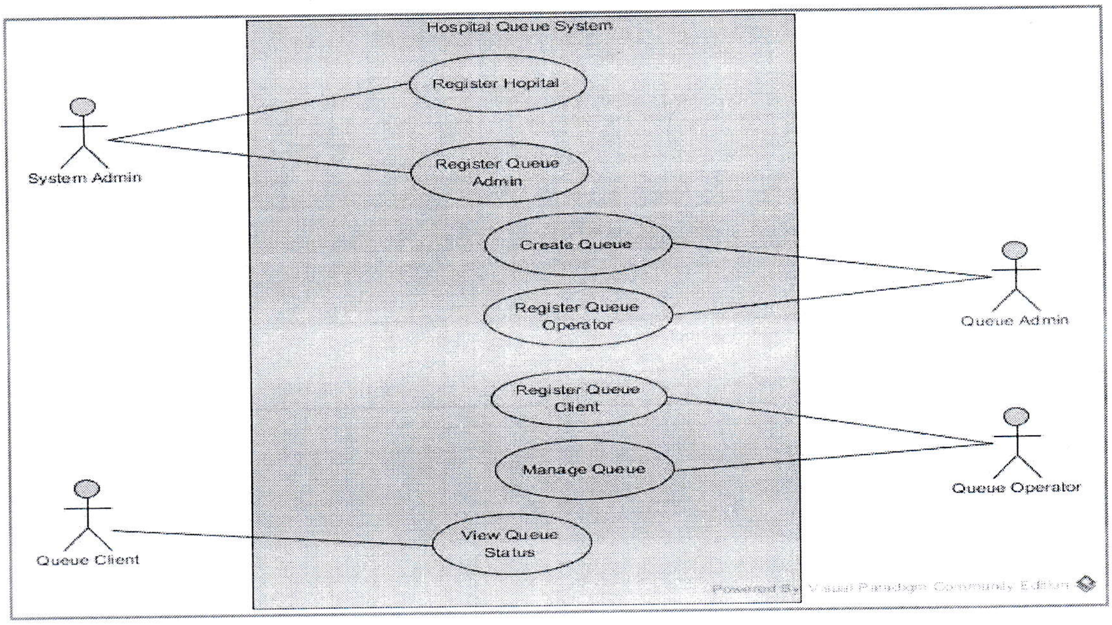


Fig. 3. Hospital Queue System Use Case Diagram

Use case role:

- **Register Hospital:** Describes a behavior for the system administration to register hospital details into the hospital queue system.
- **Register Queue Admin:** describes a behavior that a queue administration is created by the system admin.
- **Create Queue:** Describes a queue that is created by queue administration.
- **Register Queue Operator:** Describes a behavior that a queue operator is created by the queue administration.
- **Manage Queue:** Describes how the queue operator manages all activities that happen in a queue, including the insertion of a client to the queue, put queued client into a service and end the client from the .queue after the service is complete.
- **View Queue Status:** Describes a behavior where a queue client can check or view their queue status during the queue process.
- **View Queue Status:** Describes the behavior where a queue client can check or view their queue status during the queue process.

3.2 Database and Development Tools

We consider using Java programming run on Java EE environment[3], Glassfish as its web server and running web service on WSDL (Web Services Description Language) model running with XML to view and exchange data. A system based on Java technology could be strong in term of security and great in term of performance and implement with RESTful[4] architecture. The set of tools is an advantage of ease to access and deployment.

Database is created by using PHPMyAdmin[5] which is MySQL[6] management program. It comes together with XAMP[7]. The entities and relationships system is deployed with relational database[8] principle which consists of five tables. Qbusiness table stores the entity of the hospital that is registered to the hospital queue system. Quser table stores the entity of user who works with the system. Qqueue table stores the entity of a queue which is created by queue administration. Qtran table stores the entity of a queue transaction which is generated through queuing process. Customer table stores the entity of a queue client or a patient who requires hospital services and seating in the queue.

3.3 Queuing Management Mechanism

In this project "First Come, First Serve" concept and queuing theory with Little's law[9,10]is deployed as the system discipline to manage service queue. Given ʌ is the average number of items arriving per unit time; W is average waiting time per for an item, and L is an average number of items in the queuing system, so L = ʌ/W.

The Arrival Rate (ʌ) is formulated by a division of Total arrival (N) by Total Time(T) as ʌ = N/T. This means that at the time interval T the system has been observed, the number of arrival N entering to the system queue.

Finding individual waiting time : In order to find the time remaining or waiting time for an individual in the queue, we need to know the average waiting time W of the system at the period time T by being calculated from equation (1).

$$W = \frac{1}{N}\sum_{i=1}^{N} Wi \qquad \text{..............................} (1)$$

To calculate the waiting time for the Nth queue number to be in service, the average waiting time needs to be calculated onward to get the most likely average time. For instance, the queue may have an average L customers waiting in the queue with arrival rate ʌ, so calculating the average waiting time is W = L/ ʌ. Therefore, the expected waiting time for the Nth queue to reach the service is

$$WN = \sum_{i=1}^{N} Wi \qquad \text{..........................} (2)$$

According to the equation(2), an individual waiting time could be rewritten as the average waiting time multiply by the number of individual a queue number approximately. As of continuous system, the estimate time waiting could be denoted as the equation (3).

$$W(n,m) = mW_n \qquad \text{............................} (3)$$

Where m is the queue number of an individual queue and n is the number of queues included in average, an estimated waiting time of an individual, $W_{(n,m)}$ could be suggested to the customer of the queue as the multiplication of queue number m with the average waiting time W_n.

4. System Prototype Implementation

The proposed hospital queue system is required to run over the Internet or intranet; therefore, the stakeholder, system administrative users and patients can use their smart phones and Internet access devices to view their queue status. The system prototype is demonstrated by testing with a set of tools and equipment as described below:

- Locally testing with XAMP

The web server needs to be set up and tested on Windows environment and running on XAMP v3.2.1 [7], which is a bundle package of Apache, MySQL and PHP[11,12]. However, the system cannot fully operate locally since the customer/client/patient must be able to view a queue status over their wireless device. Therefore, the system has to be online to serve this requirement.

- **Online hosqueue.com**

To take the system online, a domain name needs to be registered. Also, it has been named as hosqueue.com. The system domain is also hosted with one of a domain hosting providers which gives the system space and requires server environment for the system to run.

The system has been done on top of the previous code taken from the open source software called Complain Management System written by Tousif Khan [13]. The system is built on top of pre-coding and structure with a new database design and new business processing. The system is built on PHP [11], [14] and Java Script [4], coding example is shown in the following page. The code represents part of PHP requesting queuery to the database before converting into JSON data format which performs RESTful web service. As it can be seen in the code, one important requesting element is AvgTime (Average Time). The system allows the queue admin to modify the number of samples of individual waiting time as a set of average time waiting. AvgTime is then to be used to calculate time remaining for the next remaining queue

```
$sql = "SELECT qtran.QueueID,queueno,CustID,arrive,tstatus,qqueue.AvgTim e FROM
".$dbname.".qtran INNER JOIN ".$dbname.".qqueueON qtran.QueueID =
qqueue.QueueID WHERE qtran.QueueID ='".$qid."'";
$result=mysql_query($sql); $rows =array(); while($r = mysql_fetch_assoc($result)) {
$rows['Queue'][] = $r; }
 printjson_encode($rows);
```

[Example of PHP script on data conversion by using json_encode function yields requesting queue data output into JSON data format.]

The main operation is on queue management system on PHP demonstration. When the customer/client/patient queue viewer is mainly on Android[3], [15], [16] application, coding example is shown below. This part of the code allows the application to retrieve JSON data format from the PHP web service. AvgTime abruptly calculates time remaining equivalent to the sequential order of the patient queue number. This part of the system allows the user to access data over their wireless device.

```
public void ListDrawer() {
try{JSONObjectjsonResponse =
newJSONObject(jsonResult);
JSONArrayjsonMainNode = jsonResponse.optJSONArray("Queue");
rowQueue.clear();
for (inti = 0; i<jsonMainNode.length(); i++)
        {
                JSONObjectjsonChildNode = jsonMainNode.getJSONObject(i);
                columnQueue.set(0,jsonChildNode.optString("QueueID"));
                columnQueue.set(1,jsonChildNode.optString("queueno"));
                columnQueue.set(2,jsonChildNode.optString("CustID"));
                columnQueue.set(3,jsonChildNode.optString("arrive"));
                columnQueue.set(4,jsonChildNode.optString("tstatus"));
                columnQueue.set(5,jsonChildNode.optString("AvgTime"));
                columnQueue.set(6,String.valueOf(Integer.valueOf(jsonChildNode.optString("AvgT
ime"))                                  *60*(i+1)));
                columnQueue.set(7,String.valueOf(Integer.valueOf(jsonChildNode.optString("AvgT
ime"))                                  *60*(i+1)));
```

```
                rowQueue.add(new ArrayList<String>(columnQueue));
            }
    } catch (JSONException e) { ... }}
```

[Example of Android programming function calledListDrawer, which retrieves JSON data format and displays on Android client application.]

- **Installation Client Application with Android**

  An Installation client program for Android application hosqueue.com is stored in an APK file after its compilation. The customer can download the file and install it to an android device. The program requires running on Android 4.0.3 (Ice Cream Sandwich) and above.

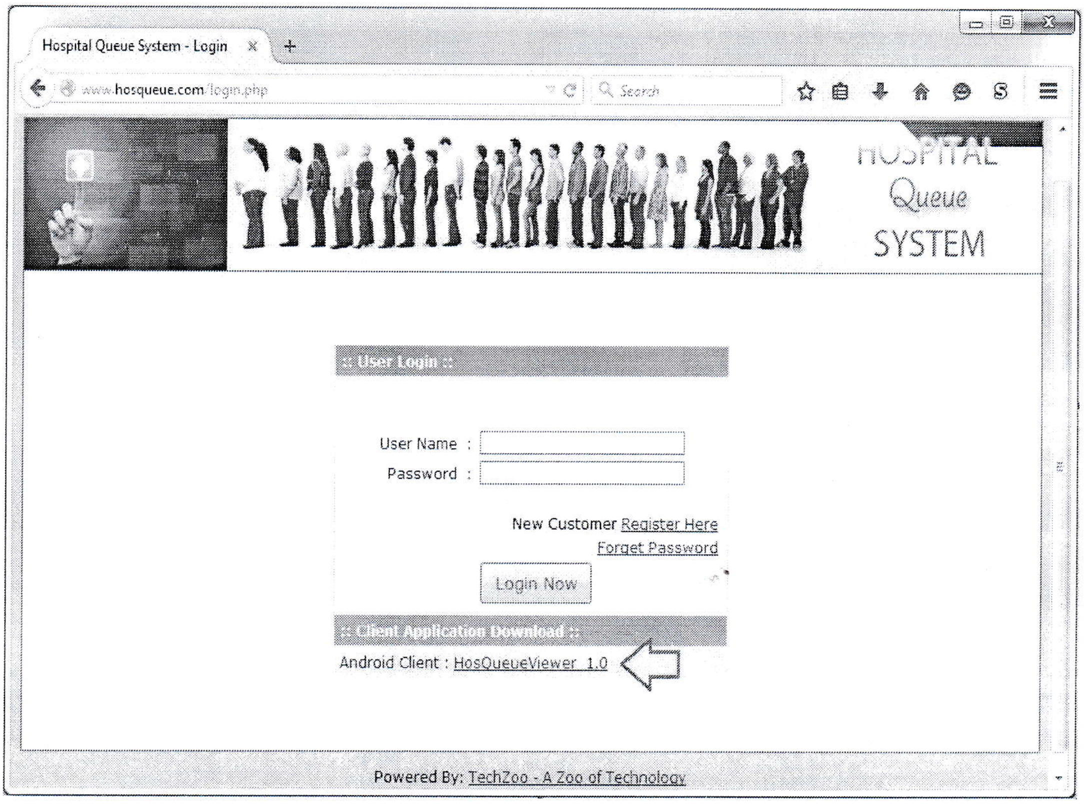
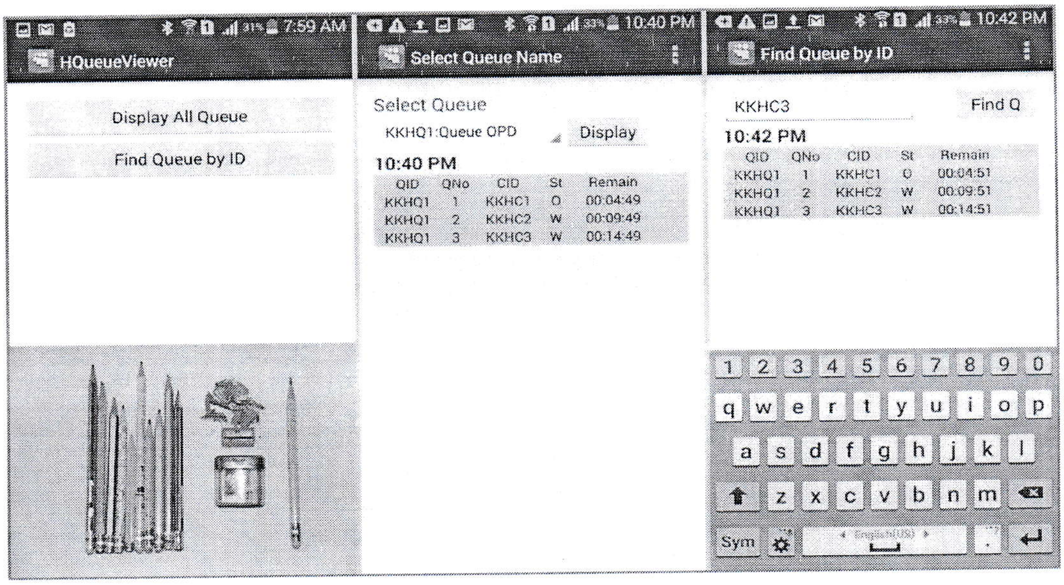

Hospital Queue System - Login
www.hosqueue.com/login.php
Search
HOSPITAL
Queue
SYSTEM
:: User Login ::
User Name :
Password :
New Customer Register Here
Forget Password
Login Now
:: Client Application Download ::
Android Client : HosQueueViewer 1.0
Powered By: TechZoo - A Zoo of Technology

Fig. 4. Available download of hospital queue viewer 1.0

The system function will display all the queue and find queue by ID as shown in figure 5.

- **Display All Queue**: The Display All Queue button leads to a view by queue selected screen where the customer can view the queue by choosing a particular queue that they want to view. Therefore, the customer is required to know which queue to look for.
- **Find Queue by ID**: The Find Queue by ID helps the customer in searching the queue in case that the customer does not know which queue it is. However, the

customer is still required to know their customer ID to be used as a finding key to the queue.



10:40 PM

QID	QNo	CID	St	Remain
KKHQ1	1	KKHC1	O	00:04:49
KKHQ1	2	KKHC2	W	00:09:49
KKHQ1	3	KKHC3	W	00:14:49

10:42 PM

QID	QNo	CID	St	Remain
KKHQ1	1	KKHC1	O	00:04:51
KKHQ1	2	KKHC2	W	00:09:51
KKHQ1	3	KKHC3	W	00:14:51

Fig.5. Queue display by searching customer ID

The queue displays the queue number, customer ID., queue status, and estimated time of service. This can help the customer go anywhere nearby or do other activities while still knowing the queue status.

5. Conclusion

Hospital Service Queue System is a project to eliminate the traditional physical queue and replace it with a convenient management. This project is designed to help the public who suffers from long queues in hospitals, especially the public hospitals. The main system functionalities which are constructed and implemented online are ready for hospital queue services; hence, the customer/patient/client can view a queue status over their wireless.

The contribution of this system does not only serve the people requesting the service in hospital but also utilize their time to do other activities. Also, the advantage of using open source it could benefit to community as a whole. Not only one hospital can benefit with the current system design and setting, but multiple hospitals can be served at the same time. An individual hospital can manage its own queues with a given power user as a queue administration. With this design, a cost sharing arrangement is possible amongst hospitals without having any budget to spend for the extra development.

❖ References

1. Kendall, K. E., Kendall J. E.: Systems Analysis and Design. 8th, Ed., Pearson Education, Harlow (2011)
2. Bruegge, B., Dutoit, A.H.: Object-Oriented Software Engineering Using UML, Patterns, and Java. 3rd, Ed., International Edition, Pearson Education, Upper Saddle River, NJ,(2010)
3. Java Software - Oracle, https://www.oracle.com/java, Accessed 15 April 2015
4. The World Wide Web Consortium (W3C),http://www.w3.org , Accessed 12 May 2015
5. phpMyAdmin, http://www.phpmyadmin.net, Accessed 23 February 2015

6. MySQL Community, http://www.mysql.org, Accessed 23 February 2015

7. XAMPP Installers and Downloads for Apache Friends, https://www.apachefriends.org, Accessed 23 February 2015

8. Hoffer, J. A., Mary, S., Heikki, T.: Modern Database Management, 10th,Ed., Prentice-Hall, Upper Saddle River, NJ (2011)

9. Chhajed, D., Lowe, T.J.:Building Intuition: Insights From Basic Operations Management Models and Principles.,pp. 81–84. Springer, Heidelberg (2008)

10. Cooper, R.B.: Introduction to Queueing Theory, 2nd,Ed., pp.178-185, Elsevier North Holland, Inc. New York (1981)

11. PHP - Hypertext Preprocessor, http://php.net, Accessed 12 February 2015

12. The Apache Software Foundation, https://www.apachefriends.org, Accessed 23 February 2015

13. A Zoo of Technology, http://www.techzoo.org, Accessed 15 November 2014

14. Welling, L., Thomson, L.: PHP and MySQL® Web Development, 4th,Ed.,Addison- Wesley Professional, Boston (2008)

15. Meier, R.: Professional Android 4 Application Development, updated for Android 4, John Wiley & Sons, Inc. Indiana (2012)

16. Android, https://www.android.com, Accessed 17 January 2015

-------------------------------------Thank You------------------------------------

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE (AUTONOMOUS)
RAMAKRISHNA MISSION ASHRAMA
NARENDRAPUR, KOLKATA-700 103, W. B., INDIA
(A Branch Centre of Ramakrishna Mission, Belur Math, Howrah-711202)

Project Completion Certificate

Date: 03/06/2020

This is to certify that **Sri Suryadeep Das** (ROLL NO: CSUG/159/17) a student of Department of Computer Science, Ramakrishna Mission Residential College(Autonomous), has undergone a Project work from January 15, 2020 to May 30, 2020 titled "**Automatic Essay Grading Using Machine Learning**".

Department of Computer Science
Ramakrishna Mission Residential College
(Autonomous)
Narendrapur, Kolkata-700 103

Dr. Siddhartha Banerjee
(Head of the Department)
Department of Computer Science

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE(AUTONOMOUS),NARENDRAPUR

AUTOMATIC ESSAY GRADING USING MACHINE LEARNING

NAME : SURYADEEP DAS

ROLL NO : CSUG/159/17

Department of Computer Science

2020

AUTOMATIC ESSAY GRADING USING MACHINE LEARNING

PROJECT REPORT

-Suryadeep Das

(CSUG/159/17)

INTRODUCTION:

Automated essay scoring (AES) is the use of specialized computer programs to assign grades to essays written in an educational setting. It is a form of educational assessment and an application of natural language processing. Its objective is to classify a large set of textual entities into a small number of discrete categories, corresponding to the possible grades, for example, the numbers 1 to 10. Therefore, it can be considered a problem of statistical classification.

Several factors have contributed to a growing interest in AES. Among them are cost, accountability, standards, and technology. Rising education costs have led to pressure to hold the educational system accountable for results by imposing standards. The advance of information technology promises to measure educational achievement at reduced cost.

OVERVIEW:

This project aims to build a machine learning system for automatic scoring of essays written by students.The basic idea is to search for features which can model the attributes like language fluency, vocabulary, structure, organization, content etc. As we pointed out in our initial draft, such a system can have a high utility in many places. For instance, currently, evaluation of essay writing section in exams like GRE, GMAT, and TOEFL is done manually. And, so automating such a system may prove to be highly useful. We have built a linear regression model with polynomial basis function to predict the score of a given essay. The subsequent sections explain the input data, features extraction, detailed approach, results, and future scope of the work.

DATA EXPLORATION:

We have taken the input data from Kaggle.com. We are given ~13000 essays written by school students of Grade 7, 8 and 10. These essays are divided into 8 sets - each set of essays from a different context - to ensure variability of the domain. Each set of essays was generated from a single prompt. Along with the ASCII text of each essay, we also have scores given to each essay by two human evaluators and a combined resolved score.

We split this data into two sets: training set and testing set, as follows:

Essay set	Total number of Essays	Number of Essays used for training	Number of Essays used for testing
1	1783	1200	583
2	1800	1200	600
3	1726	1200	526
4	1772	1200	572
5	1805	1200	605
6	1800	1200	600
7	1569	1200	369
8	723	500	223
TOTAL	**12978**	**8900**	**4078**

Once we have the training and testing data, we can extract features from each of the document and train our model. These are explained in subsequent sections.

FEATURES EXTRACTION:

Feature extraction is the most important part of any machine learning task and so is the case with us. To build effective essay scoring algorithm, our aim is to try to model attributes like language fluency, grammatical and syntactic correctness, vocabulary and types of words used, essay length, domain information etc.

At present, our model is using the following set of features:

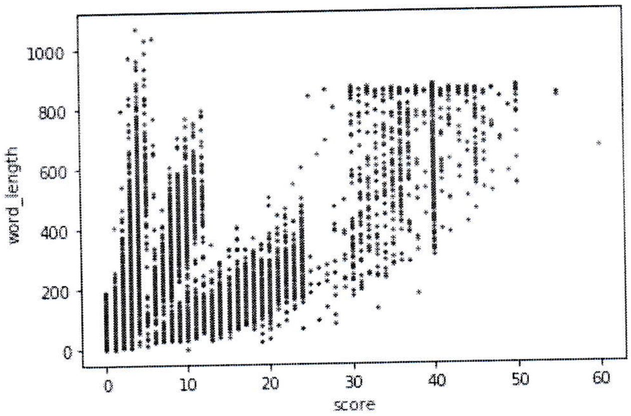
1. Total number of words used in the essay.
2. Total number of sentences.
3. Sentiment analysis.
4. Average word length.
5. Grammatical errors.

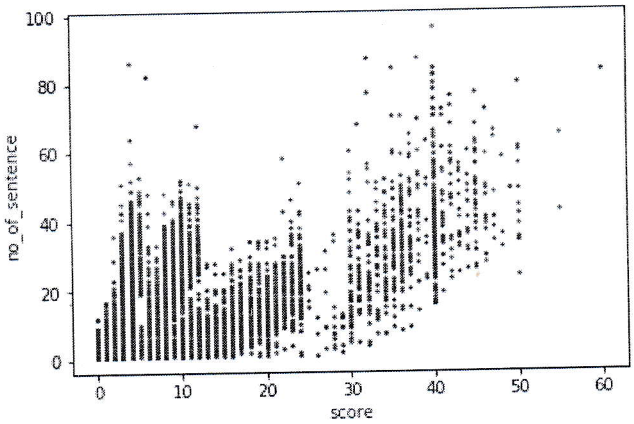
We have used the textblob library in python to extract the above features from given texts.

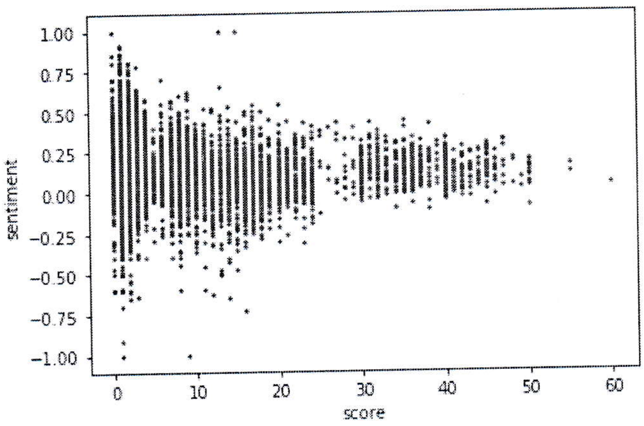
APPROACH, EVALUATION and RESULTS:

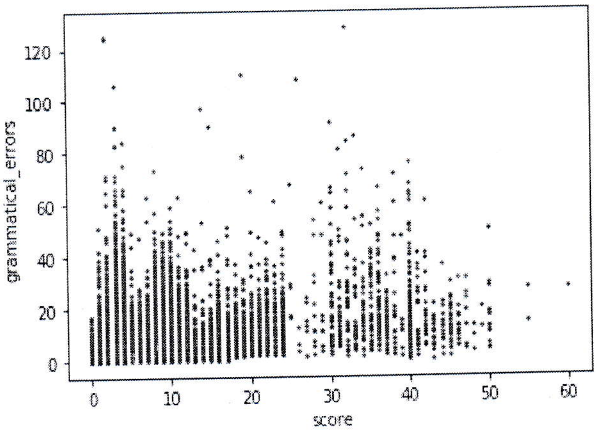
First these essays were individually scored by human beings, after which feature extraction was carried out. Once we were done with feature extraction, we applied different regression model to find the optimal condition.

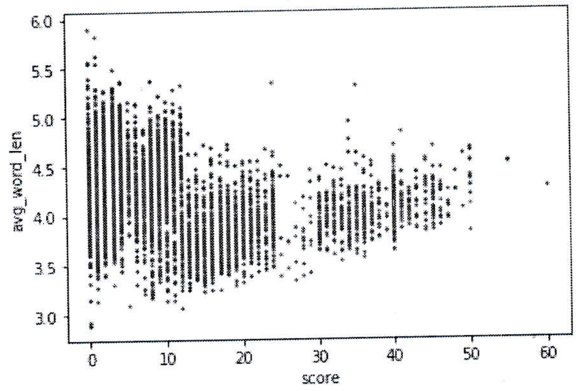
We plot graphs of the scores given manually to the respective features.











The next step was to normalize the dataset to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. For machine learning, every dataset does not require normalization. It is required only when features have different ranges.

Once the data was normalised we used it to train different regression models with an average of 75% of the data and the results were compared.

```
[['SVR', 0.5088554660112243],
 ['SGDRegressor', 0.4194873891624755],
 ['BayesianRidge', 0.47782025610749496],
 ['LassoLars', -0.00083156872616863],
 ['DecisionTreeRegressor', 0.19079996102015295],
 ['RandomForestRegressor', 0.6071590075261986],
 ['PassiveAggressiveRegressor', -22.852702249161165],
 ['TheilSenRegressor', 0.3734631557334913],
 ['LinearRegression', 0.47782613303407506],
 ['Ridge', 0.47775937057294576],
 ['ElasticNet', -0.00083156872616863]]
```

As, RandomForestRegressor gives R square score of about 0.6, which is considered good fit, our essay_prediction model will be in RandomForestRegressor.

CONCLUSIONS:

Automatic essay grading is a very useful machine learning application. It has been studied quite a number of times, using various techniques like latent semantic analysis etc. The current approach tries to model the language features like fluency, grammatical correctness, domain information content of the essays, and tries to fit the best polynomial in the feature space using RandomForestregression with polynomial basis functions

The future scope of the given problem can extend in various dimensions. One such area is to search and model good semantic and syntactic features. For this, various semantic parsers etc. can be used. Other area of focus can be to come up with a better tool like neural networks etc.

REFERENCES:

1. [illegible]
2. [illegible]
3. [illegible]
4. [illegible]

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE (AUTONOMOUS)
RAMAKRISHNA MISSION ASHRAMA
NARENDRAPUR, KOLKATA-700 103, W. B., INDIA
(A Branch Centre of Ramakrishna Mission, Belur Math, Howrah-711202)

Project Completion Certificate

Date: 03/06/2020

This is to certify that **Sri Swapnajoy Saha** (ROLL NO: CSUG/189/17) a student of Department of Computer Science, Ramakrishna Mission Residential College(Autonomous), has undergone a Project work from January 15, 2020 to May 30, 2020 titled **"VSUMM: An approach based on color features for Automatic Summarization".**

Department of Computer Science
Ramakrisnha Mission Residential College
(Autonomous)
Narendrapur, Kolkata-700 103

Dr. Siddhartha Banerjee
(Head of the Department)
Department of Computer Science

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE (AUTONOMOUS)
RAMAKRISHNA MISSION ASHRAMA
NARENDRAPUR, KOLKATA-700 103, W. B., INDIA
(A Branch Centre of Ramakrishna Mission, Belur Math, Howrah-711202)

Project Completion Certificate

Date: 03/06/2020

This is to certify that **Sri Arnab Kundu** (ROLL NO: CSUG/173/17) a student of Department of Computer Science, Ramakrishna Mission Residential College(Autonomous), has undergone a Project work from January 15, 2020 to May 30, 2020 titled **"VSUMM: An approach based on color features for Automatic Summarization"**.

Department of Computer Science
Ramakrishna Mission Residential College
(Autonomous)
Narendrapur, Kolkata-700 103

Dr. Siddhartha Banerjee
(Head of the Department)
Department of Computer Science

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE(AUTONOMOUS),NARENDRAPUR

VSUMM: AN APPROACH BASED ON COLOR FEATURES FOR AUTOMATIC SUMMARIZATION

NAME: SWAPNAJOY SAHA
COLLEGE ROLL NO.: CSUG/189/17

NAME: ARNAB KUNDU
COLLEGE ROLL NO.: CSUG/173/17

Department of Computer Science

2020

VSUMM: AN APPROACH BASED ON COLOR FEATURES FOR AUTOMATIC SUMMARIZATION

ABSTRACT

The fast evolution of digital video has brought many new multimedia applications and, as a consequence, research into new technologies that aim at improving the effectiveness and efficiency of video acquisition, archiving, cataloging and indexing, as well as increasing the usability of stored videos. Among all possible research areas, video summarization is one of the most important topics, which may enable a quick browsing of a large collection of video data and to achieve efficient content access and representation. Essentially, this research area consists of automatically generating a short summary of a video, which can either be a static summary or a dynamic summary. In this paper, we present VSUMM, a methodology for the development of static video summaries. The method is based on color feature extraction from video frames and unsupervised classification. We also develop a new subjective method to evaluate video static summaries. The video summaries are manually created by users and compared with different approaches found in the literature. Experimental results show – with a confidence level of 98% – that the proposed solution provided static video summaries with superior quality relative to the approaches to which it was compared.

Keywords- **Video summarization**; **Static video summary**; **Keyframes**; **Clustering**; **Color histogram**.

INTRODUCTION

The recent advances in compression techniques, the decreasing cost of storage and the availability of high-speed connections have facilitated the creation, storage and distribution of videos. This leads to an increase in the amount of video data deployed and used in applications such as search engines and digital libraries, for example. This situation puts not only multimedia data types into evidence, but also leads to the requirement of efficient management of video data. Such requirements paved the way for new research areas, such as video summarization.

According to *[1]*, there are two fundamental types of video summaries: **static video summary** – also called **representative frames, still-image abstracts** or **static storyboard** and **dynamic video**

skimming – also called **video skim**, **moving-image abstract** or **moving storyboard**. Static video summaries are composed of a set of **keyframes**(A keyframe is a frame that represents the content of a logical unit, as a shot or scene. This content must be the most representative as possible) extracted from the original video, while dynamic video summaries are composed of a set of **shots**(A shot represents a spatio-temporally coherent frame sequence, which captures a continuous action from a single camera.) and are produced taking into account the similarity or domain-specific relationships among all video shots.

One advantage of a video skim over a keyframe set is the ability to include audio and motion elements that potentially enhance both the expressiveness and the amount of information in the summary. In addition, according to *[2]*, it is often more entertaining and interesting to watch a skim than a slide show of keyframes. On the other hand, keyframe sets are not restricted by any timing or synchronization issues and, therefore, they offer much more flexibility in their organization for browsing and navigation purposes, in comparison with the strict sequential display of video skims, as demonstrated in *[3]*, *[4]*, *[5]*, *[6]*, *[7]*. In this paper, we focus on the production of static video summaries.

Recently, video summarization has attracted considerable interest from researchers and as a result, various algorithms and techniques have been proposed in the literature, most of them based on clustering techniques (*[8]*, *[9]*, *[10]*, *[11]*). Comprehensive surveys of past video summarization results can be found in *[1]*, *[12]*, *[13]*.

In the case of clustering-based techniques, the basic idea is to produce the summary by clustering together similar frames/shots and then showing a limited number of frames per cluster (usually, one frame per cluster). For such approaches it is important to select the features upon which the frames can be considered similar (e.g., color distribution, luminance, motion vector) and also to establish different criteria that can be employed to measure the similarity.

Although there are some techniques that produce summaries of acceptable quality, they usually use intricate clustering algorithms that make the summarization process computationally expensive *[11]*. For example, in *[9]* the time needed for computing the summaries takes around 10 times the video length. This means that a potential user would wait around 20 minutes to have a concise representation of a video that he/she could have watched in just two minutes.

In this paper, it is proposed a simple and effective approach for automatic video summarization, called **Video SUMMarization** (VSUMM). The method is based on the extraction of color features from video frames and unsupervised classification. In addition, a new subjective methodology to evaluate video summaries is developed, called **Comparison of User Summaries (CUS)**. In this methodology, the video summaries are manually created by users and are compared with approaches found in the literature. The evaluation of VSUMM is done on 50 videos from the **Open Video Project (OV)** *[14]* and experimental results show that the VSUMM approach produces video summaries with superior quality relative to the approaches to which it was compared.

The main contributions of this paper are:

(1) a mechanism designed to produce static video summaries, which presents the advantages of the main concepts of related work in the video summarization;

(2) a new evaluation method of video summaries, which reduces the subjectivity in the evaluation task, quantifies the summary quality and allows comparisons among different techniques quickly.

PAST WORKS

Different approaches have been studied in order to elaborate our solution. Some of the main ideas that are related to the proposed solution are discussed next.

Zhuang et al. *[17]* proposed a method for keyframe extraction based on unsupervised clustering. In that work, the video is segmented into shots and then a color histogram (in the HSV color space) is computed from every frame. The clustering algorithm uses a threshold _ which controls the clustering density. Before a new frame is classified as pertaining to a certain cluster, the similarity between this node and the centroid of the cluster is computed first. If this value is less than _, it means that this node is not close enough to be added into the cluster. The keyframes selection is employed only to the clusters which are big enough to be considered as keyclusters. In such case, a representative frame is extracted from this cluster as the keyframe. A keycluster is considered large enough if it is larger than the average cluster size. For each keycluster, the frame which is closest to the keycluster centroid is selected as the keyframe. According to *[17]*, the proposed technique is efficient and effective, however, no comparative evaluation is performed for validating such assertions.

Hanjalic and Zhang *[18]* presented a method for automatically producing a summary of an arbitrary video sequence. The method is based on cluster-validity analysis and is designed to work without any human supervision. The entire video material is first grouped into clusters. Each frame is represented by color histograms in the YUV color space. A partitional clustering is applied n times to all frames of a video sequence. The pre-specified number of clusters starts at one and is increased by one each time the clustering is applied. Next, the system automatically finds the optimal combination(s) of clusters by applying the cluster-validity analysis. After the optimal number of clusters is found, each cluster is represented by one characteristic frame, which then becomes a new keyframe for that video sequence. *[18]* concentrated on the evaluation of the proposed procedure for cluster-validity analysis, instead of on evaluating the produced summaries.

Gong and Liu *[19]* proposed a technique for video summarization based on the Singular Value Decomposition (SVD). At first, a set of frames in the input video is selected (one from every ten frames) and then, color histograms in the RGB color space are used to represent video frames. To incorporate spatial information, each frame is divided into 3 × 3 blocks, and a 3D-histogram is created for each of the blocks. These nine histograms are then concatenated together to form a feature vector. Using this feature vector extracted from the frames, a feature-frame matrix A (usually sparse) is created for the video sequence. Therefore, the SVD is performed on A to obtain the matrix V , in which each column vector represents one frame in the refined feature space. Next, the cluster closest to the origin of the refined feature space is found, the content value of this cluster is computed and this value is used as the threshold for clustering the remaining frames. From each cluster, the system selects the frame that is closest to the cluster center as keyframe. This method is not compared with other techniques.

Mundur et al. *[9]* developed a method based on Delaunay Triangulation (DT), which is applied for clustering the video frames. The first step of their method is to obtain the video frames from the original video, pre-sampling the video frames. Each frame is represented by a color histogram in the HSV color space. This histogram is represented as a row vector and the vectors for each frame are concatenated into a matrix. To reduce the dimensions of this matrix, the Principal Components Analysis (PCA) is applied. After that, the Delaunay diagram is built. The clusters are obtained by separating edges in the Delaunay diagram. Finally, for each cluster, the frame that is nearest to its center is selected as the keyframe. To evaluate the summaries, *[9]* defined three objective metrics:

significance factor, overlap factor and compression factor. In spite of the fact that the proposed method has been designed to be fully automatic (i.e., with no user-specified parameters and well suited for batch processing), it requires between 9 to 10 times the video length to produce the summary. Furthermore, the method does not preserve the video temporal order.

Furini et al. *[10]* introduced VISTO (VIdeo STOryboards), a summarization technique designed to produce on-the-fly video storyboards. VISTO is composed of three phases. First, the video is analyzed in order to extract the HSV color description. For each input frame, a 256- dimension vector is extracted. Those vectors are then stored in a matrix and then, in the second phase, the clustering algorithm is applied to extracted data. The authors exploited the triangular inequality in order to filter out useless distance computations. To obtain the number of clusters, the pairwise distance of consecutive frames is computed. If the distance is greater than the threshold , the number of clusters is incremented. The third and last phase aims at removing meaningless video frames from the produced summary. VISTO is evaluated through a comparison study with other approaches: the DT technique *[9]* and the Open Video storyboards *[14]*. *[10]* asked a group of 20 people to evaluate the produced summaries, using the following procedure: the video is presented to the user, and just after that, the corresponding summary is also shown. The users are asked whether the summary is a good representation of the original video. The quality of the video summary is scored on a scale going from 1 (bad) to 5 (excellent), and the mean opinion score is considered as an indication of the summary quality.

Guironnet et al. *[20]* proposed a method for video summarization based on camera motion. It consists in selecting frames according to the succession and the magnitude of camera motions. The method is based on rules to avoid temporal redundancy among the selected frames. The authors developed a subjective method to evaluate the proposed summary. In their experiments, 12 subjects are asked to watch a video and to create a summary manually. From the summaries of different subjects, an "optimal" one is built automatically. This "optimal" summary is then compared with the summaries obtained by different methods. The construction of an "optimal" summary is a difficult stage, which requires various parameters to be fixed.

According to the analysis of the approaches found in literature, it can be noticed that the keyframe selection techniques used several visual features and statistics. These features can affect both the computational complexity and the summary quality. Normally, the extraction of the video features may produce a high dimensional matrix. For this reason, dimensionality reduction techniques are used in order to reduce the size of those matrices, as it can be seen in *[9]*, *[19]*, for example. Needless to say, this additional step requires even more processing time. Another serious problem that can be observed is the lack of trustworthy comparisons among existing techniques. In other words, a consistent evaluation framework is seriously missing in video summarization research.

The VSUMM approach, proposed in present work, draws on the advantages of the existing techniques and concepts presented in related work. A fully reproducible evaluation framework is proposed and applied for comparisons among VSUMM and three other proposals, indicating that VSUMM is able to provide better summaries, according to the defined metrics.

METHODOLOGY - VSUMM APPROACH

Figure 1 illustrates the steps of our method to produce static video summaries. Initially, the original video is split into frames (step 1). In next step (step 2), color features are extracted to form a color histogram in HSV color space. VSUMM does not consider all the video frames, but takes a sample. In addition, the meaningless frames found in the sample are removed. After that (step 3), the frames are grouped by *k*-means clustering algorithm. Then (step 4), one frame per cluster is selected (this selected frame is the keyframe). To refine the static video summary composed by the keyframes (step 5), the keyframes that are too similar are eliminated. Finally, the remaining keyframes are arranged in the original temporal order to facilitate the visual comprehension of result. Each step is detailed in next subsections.

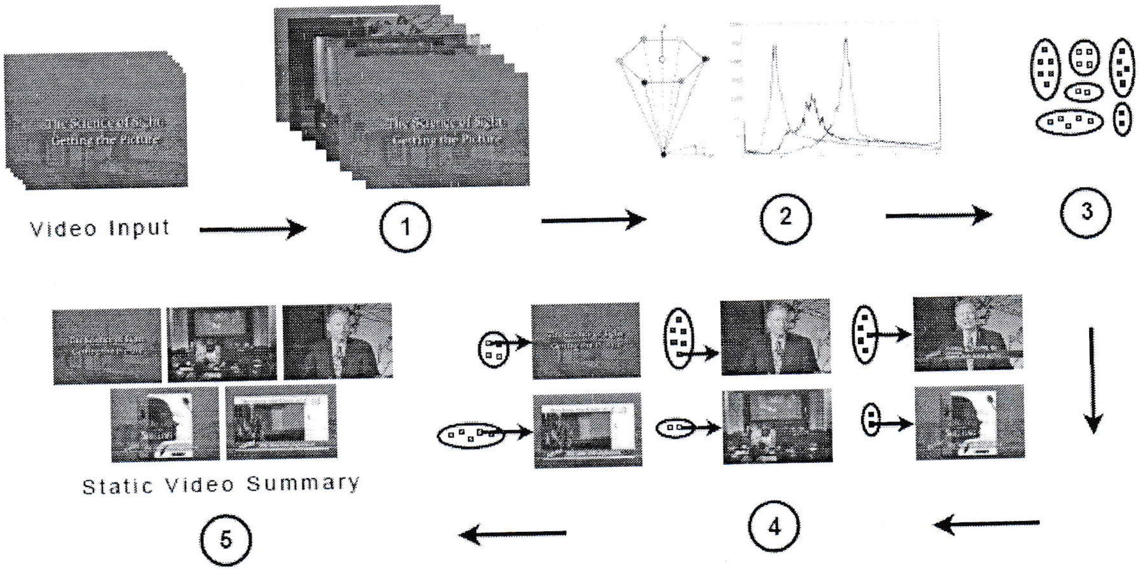


Figure 1: VSUMM approach

A. Temporal Video Segmentation

Temporal video segmentation is the first step towards automatic video summarization. Its goal is to divide the video stream into a set of meaningful and manageable basic elements (e.g., shots, frames) *[21]*. In literature, the **shot boundary detection** *[22]* is widely used as first step to produce summaries (e.g., *[8]*, *[17]*, *[18]*, *[23]*, *[24]*, *[25]*, *[26]*, *[27]*).

Another type of video segmentation is the **extraction of video frames**, where there is no temporal analysis of the video. Each frame is treated separately, the video sequence is split into images. Several authors have used this approach (e.g., [4], [9], [10], [19], [28]), and it is also used in this work. Moreover, VSUMM does not consider all the video frames, but takes only a subset taken at a pre-determined sampling rate. This is the so-called **pre-sampling** approach.

By using a sampling rate, the number of video frames to be analyzed are reduced. The sampling rate assumes a fundamental importance, since the smaller the sampling rate, the shorter the video summarization time. Nevertheless, very low sampling rates can lead to poor quality summaries. Videos that have long shots tend to present an advantage with the pre-sampling approach, on the other hand, in those videos that present shorter shots, important parts of its content may not be represented.

The relationship between the **loss of information** and the **shot size** is directly associated with the sample rates selected during the summarization process.

In VSUMM, the sampling rate is obtained by dividing the number of video frames by the video frame rate. For instance, for a two-minute-long video with a frame rate of 30 frames per second (i.e., 3600 frames), the total number of frames to be extracted is given by 120 (3600/30) frames.

B. Color Feature Extraction

Color is perhaps the most expressive of all the visual features *[29]*. In VSUMM, color histogram *[30]* is applied to describe the visual content of video frames. This technique is computationally trivial to compute and is also robust to small changes of the camera position. Furthermore, color histograms tend to be unique for distinct objects. For these reasons, this technique is widely used in automatic video summarization (*[9]*, *[10]*, *[17]*, *[18]*, *[19]*).

Some key issues of histogram-based techniques are the selection of an appropriate color space and the quantization of that color space. In VSUMM, the color histogram algorithm is applied to the HSV color space, which is a popular choice for manipulating color. The HSV color space was developed to provide an intuitive representation of color and to be near to the way in which humans perceive and manipulate color. The VSUMM color histogram is computed only from the Hue component, which represents the dominant spectral component color in its pure form *[31]*. Moreover, the quantization of the color histogram is set to 16 color bins, aiming at reducing significantly the amount of data without loosing important information. The color bins value was established through experimental tests (see *[16]*).

C. Elimination of Meaningless Frames

A **meaningless frame** is a monochromatic frame due to fade-in/fade-out effects. To remove possible meaningless frames, VSUMM computes the standard deviation of the frame feature vector. As the standard deviation of monochromatic frames is equal to zero or a sufficiently small value close to zero (There are frames that are not completely homogeneous in color, but can be regarded as meaningless frames), VSUMM just removes these frames.

This step is also employed by *[10]*. Unlike VSUMM, which removes meaningless frames as a pre-processing step, *[10]* apply it as a post-processing step, after an initial summary is produced. Nevertheless, there is no point in using meaningless frames in the clustering step and, hence, the removal of such frames is performed before clustering in VSUMM.

D. Clustering

The *k*-means clustering algorithm *[32]* is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem *[33]*. In this work, the *k*- means algorithm is applied to cluster similar frames, although slightly modified in how it initially distributes the video frames among the *k* clusters. This modification is applied to improve *k*-means performance while producing more effective results.

The frames are initially grouped in sequential order, instead of randomly as in the original *k*-means algorithm. As an example, suppose $k = 5$ and a set of 50 frames sampled from a video. In the original *k*-means, the frames would be initially allocated randomly among the 5 clusters in order to start their iterative refinement. In case of VSUMM, the initial allocation is going to be done by associating the first 10 frames to the first cluster, the next 10 frames to the second one, and so on. This procedure is adopted based on the fact that consecutive frames typically show some similarity among them already, making it faster for *k*-means to converge.

One drawback of the *k*-means clustering algorithm is that it demands the number of clusters *k* to be fixed a priori. Nevertheless, *k* is related to the summary size, which is going to depend both on video length and on its dynamics. This means that different videos require different values for *k*. To overcome this difficulty imposed by *k*-means, a fast procedure to make a reasonable estimate of the number of clusters is implemented. VSUMM computes the pairwise distance of consecutive frames in the extracted sample, according to Euclidean distance. Then, the value selected for *k* is based on a threshold *T*, which measures the sufficient content change in the video sequence. Every time the distance between two consecutive frames is greater than *T*, then *k* is incremented. The threshold value applied in this work, established through experimental tests, is equal to 0.5.

Figure 2 shows an example of how these distances are distributed along time. It is observed that there are points in time in which the distance between consecutive frames varies considerably (corresponding to peaks), whereas there are longer periods in which the variation is very small (corresponding to denser regions). Usually, peaks correspond to a sudden change in the video, while in dense regions frames are more similar to one another. Hence, frames between two peaks can be considered as a set of similar frames and therefore, the number of peaks provides a reasonable estimation to the number of clusters *k*.

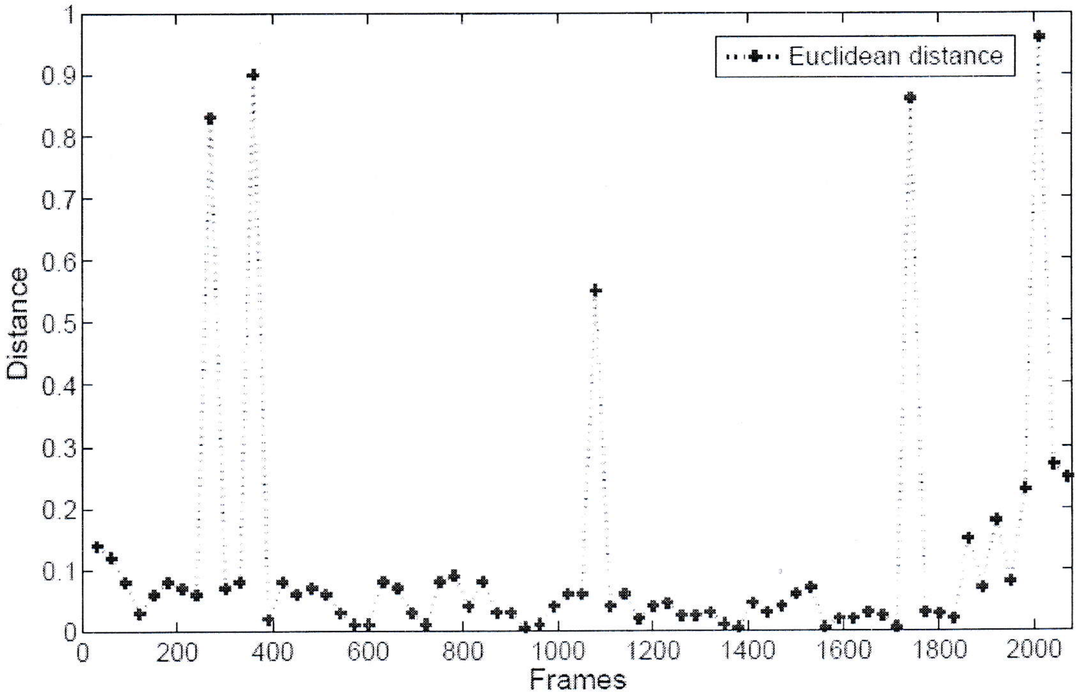


Figure 2: Pairwise distances of sampled frames of the video The Great Web of Water, Segment 02 (available at [14]).

It is worth noticing that our method for the estimation of the number of clusters is based on a simple shot boundary detection method *[34]*, whereas k is incremented for each sufficient content change in the video sequence.

E. Keyframe Extraction

Once the clusters are formed by k-means, they can be further analyzed for keycluster selection. The strategy applied for keyclusters selection is similar to the one proposed in *[17]*, which is also applied in *[27]*. In VSUMM, a cluster is considered a **keycluster** if its size is larger than half the average cluster size (this value has shown to be more suitable as cut-off point than the average cluster size, as defined in *[17]*). For each keycluster, the frame which is closest to the keycluster centroid – measured by Euclidean distance – is selected as a keyframe.

F. Elimination of Similar Keyframes

The goal of this step is to avoid that keyframes too similar appear in the produced summaries. For this purpose, the keyframes are compared among themselves through color histogram. The similarity is based on a threshold T, the same used to estimate the number of clusters. If the measured similarity is lower than T, then the keyframe is removed from the summary.

In Figure 3, it is possible to see an example of similar keyframes ($T< 0.5$) and non-similar keyframes ($T >= 0.5$). It is interesting to notice that the frames do not need to be identical to be considered too similar.

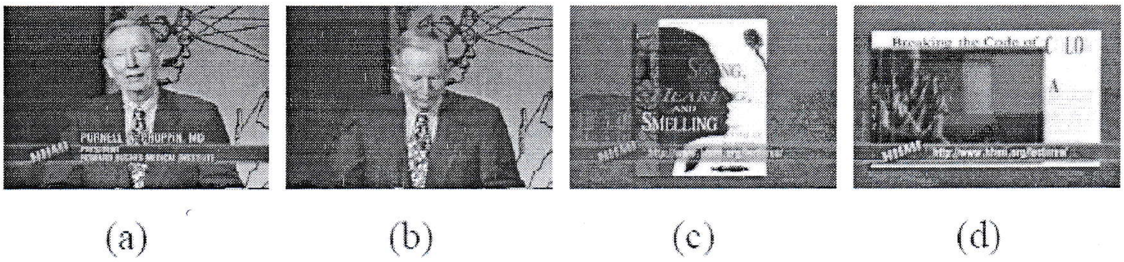


Figure 3: Similar keyframes (a–b) and non-similar keyframes (c–d) of the video Senses And Sensitivity, Introduction to Lecture 2 (available at [14]).

Finally, the remaining keyframes are arranged in temporal order to make the produced summary easier to understand.

CONCLUSIONS

Automatic video summarization has been receiving growing attention from the scientific community. This attention can be explained by several factors, for instance, (1) the advances in the computing and network infrastructure, (2) the growth of the number of videos published on the Internet, (3) scientific challenges, (4) practical applications as search engines and digital libraries, (5) inappropriate use of traditional video summarization techniques to describe, represent and perform search in large video collections. As examples the video search engines as Alta Vista, Google, and Yahoo usually represent entire videos by a single keyframe.

In this paper, we presented VSUMM, a mechanism designed to produce static video summaries. It presents the advantages of the concepts of related work in the video summarization; on a single method, VSUMM includes the main contributions of previously proposed techniques. We also developed a new subjective method to evaluate the proposed summary, which (1) reduces the subjectivity of evaluation task, (2) quantifies the summary quality and (3) allows comparisons among different techniques quickly.

One of the future lines of investigation will be to test VSUMM on different genres of videos, such as cartoons, sports, tv-shows; and test VSUMM on long videos. In addition, other features will be investigated, for example, motion, shape, texture. The fusion of different features is also an interesting future direction. Furthermore, techniques to estimate the number of clusters will be exploited, for instance, Akaike's Information Criterion (AIC) *[37]* or Minimum Description Length (MDL) *[38]*. Moreover, other clustering algorithms will be investigated, for example, DBSCAN *[39]*, a density-based clustering method. Finally, VSUMM can be extended to produce video skims. It can be created from keyframes by joining fixed-size segments, subshots, or the whole shots that enclose them, as employed in *[18]*.

REFERENCES

[1] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 3, no. 1, 2007.
[2] Y. Li, T. Zhang, and D. Tretter, "An overview of video abstraction techniques," HP Laboratory, HP-2001-191, Tech. Rep., July 2001.
[3] M. M. Yeung and B.-L. Leo, "Video visualization for compact representation and fast browsing of pictorial content," IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 5, pp. 771–785, 1997.
[4] S. Uchihashi, J. Foote, A. Girgensohn, and J. S. Boreczky, "Video manga: generating semantically meaningful video summaries," in Proceedings of the ACM International Conference on Multimedia (Part 1), New York, NY, USA, 1999, pp. 383–392. 4http://video.altavista.com 5http://video.google.com 6http://video.search.yahoo.com
[5] A. Girgensohn, "A fast layout algorithm for visual video summaries," in Proceedings of the International Conference on Multimedia and Expo (ICME). Washington, DC, USA: IEEE Computer Society, 2003, pp.70-80.
[6] J. ´Cali´c, D. P. Gibson, and N. W. Campbell, "Efficient layout of comic-like video summaries," IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 7, pp. 931–936, 2007. [7] T. Wang, T. Mei, X.-S. Hua, X.-L. Liu, and H.-Q. Zhou, "Video collage: A novel presentation of video sequence," in Proceedings of the IEEE International Conference on Multimedia and Expo, 2007, pp. 1479–1482.
[8] Y. Hadi, F. Essannouni, and R. O. H. Thami, "Video summarization by k-medoid clustering," in Proceedings of the ACM Symposium on Applied Computing (SAC), New York, NY, USA, 2006, pp. 1400–1401.
[9] P. Mundur, Y. Rao, and Y. Yesha, "Keyframe-based video summarization using Delaunay clustering," International Journal on Digital Libraries, vol. 6, no. 2, pp. 219–232, 2006.
[10] M. Furini, F. Geraci, M. Montangero, and M. Pellegrini, "VISTO: visual storyboard for web video browsing," in Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR), 2007, pp. 635–642.

[11] ——, "On using clustering algorithms to produce video abstracts for the web scenario," in Proceedings of the IEEE Consumer Communication and Networking (CCNC). IEEE Communication Society, January 2008, pp. 1112–1116.

[12] Y. Li, S.-H. Lee, C.-H. Yeh, and C.-C. Kuo, "Techniques for movie content analysis and skimming: tutorial and overview on video abstraction techniques," Signal Processing Magazine, IEEE, vol. 23, no. 2, pp. 79–89, March 2006.

[13] A. G. Money and H. Agius, "Video summarisation: A conceptual framework and survey of the state of the art," Journal of Visual Communication and Image Representation (JVCIR), vol. 19, no. 2, pp. 121–143, February 2008.

[14] The Open Video Project. http://www.open-video.org.

[17] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," in Proceedings of the IEEE International Conference on Image Processing (ICIP), vol. 1, 1998, pp. 866–870.

[18] A. Hanjalic and H. Zhang, "An integrated scheme for automated video abstraction based on unsupervised clustervalidity analysis," IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 8, pp. 1280–1289, 1999.

[19] Y. Gong and X. Liu, "Video summarization using singular value decomposition," in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). Los Alamitos, CA, USA: IEEE Computer Society, 2000, pp. 2174–2180.

[20] M. Guironnet, D. Pellerin, N. Guyader, and P. Ladret, "Video summarization based on camera motion and a subjective evaluation method," EURASIP Journal on Image and Video Processing, pp. Article ID 60 245, 12 pages, April 2007.

[21] I. Koprinska and S. Carrato, "Temporal video segmentation: a survey," Signal Processing: Image Communication, vol. 16, no. 5, pp. 477–500, 2001.

[22] C. Cotsaces, N. Nikolaidis, and L. Pitas, "Video shot detection and condensed representation: A review," IEEE Signal Processing Magazine, vol. 23, no. 2, pp. 28–37, 2006.

[23] S. V. Porter, M. Mirmehdi, and B. T. Thomas, "A shortest path representation for video summarisation," in Proceedings of the IEEE International Conference on Image Analysis and Processing, 2003, pp.460–465.

[24] J. Rong, W. Jin, and L. Wu, "Key frame extraction using intershot information," in Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), 2004, pp. 571– 574.

[25] Z. Li, G. M. Schuster, and A. K. Katsaggelos, "Minmax optimal video summarization," IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, no. 10, pp. 1245– 1256, 2005. [26] I.-C. Chang and K.-Y. Chen, "Content-selection based video summarization," Digest of Technical Papers International Conference on Consumer Electronics (ICCE), pp. 1–2, 2007.

[27] G. C´amara-Ch´avez, F. Precioso, M. Cord, S. Phillip-Foliguet, and A. de A. Ara´ujo, "An interactive video content-based retrieval system," in 15th International Conference on Systems, Signals and Image Processing (IWSSIP), Bratislava, Slovakia, June 2008, pp. 133–136.

[28] I. Yahiaoui, B. M´erialdo, and B. Huet, "Automatic video summarization," in Multimedia Content-Based Indexing and Retrieval (MCBIR), 2001.

[29] A. Tr´emeau, S. Tominaga, and K. N. Plataniotis, "Color in image and video processing: most recent trends and future research directions," EURASIP Journal on Image and Video Processing, vol. 2008, no. 3, pp. 1–26, 2008.

[30] M. J. Swain and D. H. Ballard, "Color indexing," International Journal of Computer Vision, vol. 7, no. 1, pp. 11–32, November 1991.

[31] B. S. Manjunath, J. R. Ohm, V. V. Vinod, , and A. Yamada, "Color and texture descriptors," IEEE Transactions Circuits and Systems for Video Technology, vol. 11, no. 6, pp. 703– 715, June 2001.

[32] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in Proceedings of The Berkeley Symposium on Mathematical Statistics and Probability, L. M. L. Cam and J. Neyman, Eds., vol. 1. University of California Press, 1967, pp. 281–297.

[33] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification. Springer-Verlag New York, Inc., 2001, ch. Unsupervised Learning and Clustering, p. 654.

[34] S. J. F. Guimaraes, M. Couprie, A. de Albuquerque Ara´ujo, and N. J. Leite, "Video segmentation based on 2D image analysis," Pattern Recognition Letters, vol. 24, no. 7, pp. 947– 957, 2003.

[37] H. Akaike, "A new look at statistical model identification," IEEE Transactions on Automatic Control, vol. 19, pp. 716– 723, 1974.

[38] J. Rissanen, "Modelling by shortest data description," Automatica, vol. 14, pp. 465–471, 1978.

[39] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A densitybased algorithm for discovering clusters in large spatial databases with noise," in Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD), 1996, pp. 226–231.