ONLINE PLAGIARISM DETECTION FOR IMAGES

S SOWMYA MITHRA[1] K P SUPREETHI[2]

[1] JNTUH College of Engineering, Hyderabad, India.

[2] JNTUH College of Engineering, Hyderabad, India.

ABSTRACT - plagiarism is a big problem in academics, researches and it can be a big problem in every department of education sector. Students plagiarize in different areas like homework, assignments, projects, essays etc. Collecting information from multiple sources is considered as a process of learning but this learning experience is diminished when students plagiarize by copying assignments and getting credits for work they have not done. In this project we have developed a system that helps in detecting plagiarism of images in which whenever a student submits an assignment, it detects whether it is plagiarized or not by comparing with other student assignments. For this we are using Reverse Image Processing to get proposed output.

KEY WORDS – plagiarism, reverse image processing, a-hash, pre-processing, search-by-image, automated system.

1. INTRODUCTION

Plagiarism is defined as copying of someone's work and presenting it as one's own work. This system is used to analyse the plagiarized data of images [1] [2] [5]. Plagiarism affects the education quality of the students and thereby reduces the economic status of the country. Plagiarism is done by copying an image from a book or the internet without citing the original source.

The problem of plagiarism has become an important issue in the field of education and technology due to the wide use and availability of electronic devices and internet makes it easy for students, authors and even academic people to access and use any piece of information and embed it into his/her own work without proper citation. The problem is raising in an exponential manner. In this plagiarism detection software, user can upload his/her assignment along with name and roll number in the website provided for assignment submission. This web application will process the content inside the assignment and separates images from them and helps in detecting plagiarism [6]. The assignment undergoes three 3 stages namely – pre-processing, computation, classification into whether the assignment is plagiaried or not. For easy understanding, final result is represented in two ways – graphical, numerical.

OBJECTIVES

To develop an interface for easy submission of student assignments and detect plagiarism. Plagiarism detection report will be generated.

SCOPE

In recent times, the use of internet has widely increased which is leading to easy opportunity of plagiarism, the proposed system will help in detecting the same. So the plagiarism detection will be very helpful in the future. Our system can also be used as 'search by image' [9] [10].

FEATURES

Student assignments are not modified once submitted. The system will also work for images which are different in size, brightness, background, grey-scaling, but have similar content [3] [4].

2. PROBLEM STATEMENT

In recent times, more often, assignments of students are submitted in electronic forms, but it leads towards the easy opportunity of plagiarism. With the spread of information over the globe, it is very easy to copy the data from different sources and paste it in a single work without giving any acknowledgement to the sources. These actions lead towards lack of learning in students. So there is a need for detecting plagiarism to increase and improve the quality of learning of a student. Manual detection of plagiarism is difficult and time consuming. This arises the need to design an automated system to detect plagiarism and also to improve the quality of learning of the student.

3. PROPOSED SYSTEM

In proposed system, we are going to develop a system to detect the plagiarism of images in the academic assignment which will help to stop copying the assignment of other student and will improve the quality of education and also will help to improve personal skills of student. In this system, plagiarism detector measures the similar images that matches and detects plagiarism. For detecting the plagiarism, the proposed system is using reverse image search [9] [10] [8].

FLOWCHART

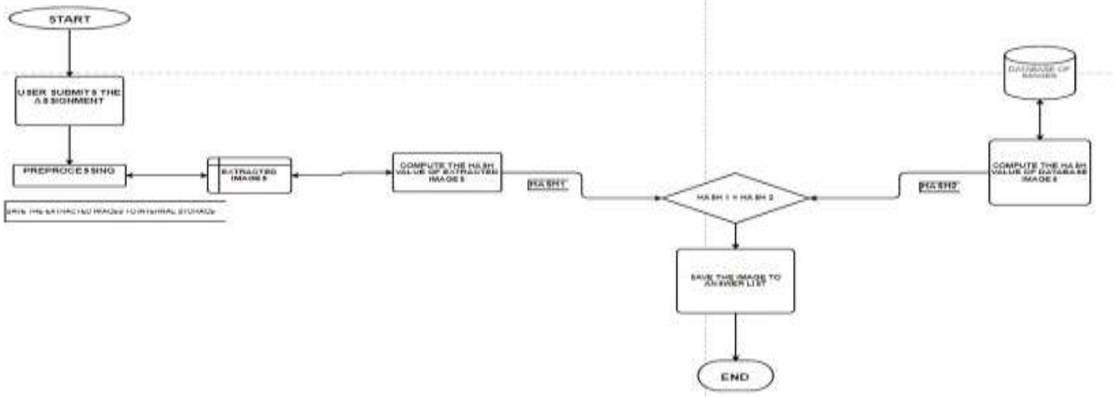
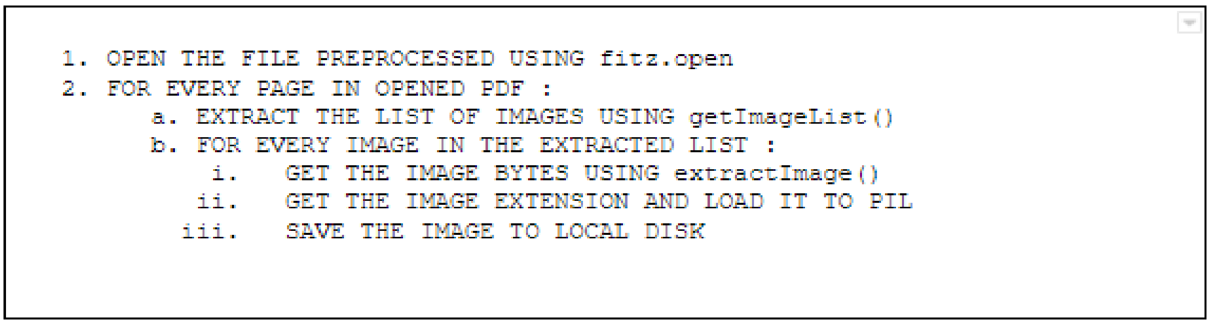

START
USER SUBMITS THE ASSIGNMENT
PREPROCESSING
SAVE THE EXTRACTED IMAGES TO INTERNAL STORAGE
EXTRACTED IMAGES
COMPUTE THE HASH VALUE OF EXTRACTED IMAGES
HASH1
HASH 1 = HASH 2
HASH2
COMPUTE THE HASH VALUE OF DATABASE IMAGES
DATABASE OF IMAGES
SAVE THE IMAGE TO ANSWER LIST
END

Figure-1: Flowchart for online plagiarism detection for images

WORKING FLOW –

- **Collection of assignments –** All the assignments or documents will be collected in electronic format along with the name and roll number of student. So that plagiarism can be detected efficiently. A user interface is created that takes file (pdf , docs.) as input and has client side and server side validation. The submitted assignments are saved onto the local storage for further processing and are deleted once processed [6].
- **Pre-processing –** pre-processing is a major step in which all submitted assignments are converted to appropriate format. All assignments are converted to same format. In this step, images are separated from text and are saved on to database for further processing. All other data types like alphabet, numbers, punctuations, etc. are eliminated and only .jpeg, .png, .jpg files are collected.

Pseudocode for pre-processing –



```
1. OPEN THE FILE PREPROCESSED USING fitz.open
2. FOR EVERY PAGE IN OPENED PDF :
      a. EXTRACT THE LIST OF IMAGES USING getImageList()
      b. FOR EVERY IMAGE IN THE EXTRACTED LIST :
           i.   GET THE IMAGE BYTES USING extractImage()
          ii.   GET THE IMAGE EXTENSION AND LOAD IT TO PIL
         iii.   SAVE THE IMAGE TO LOCAL DISK
```

Figure-2: Pseudocode for Pre-processing

- **Hash value computation –** compute hash values of images already stored on the database save the values for future use. If any new image is added do the same for the new image. The newly added image is compared with every image in the database and classified according to the percentage match of hash values.

A-HASH VALUE COMPUTATION -

This algorithm is quite fast but not sensitive to such transformations like scaling of initial image, compressing and stretching, brightness and contrast. It is based on the average value, and, as a result, is sensitive to the operations that change this average value (for instance, change of levels or colour balance) [3] [4] [7].

To build A-Hash one should perform the following steps –

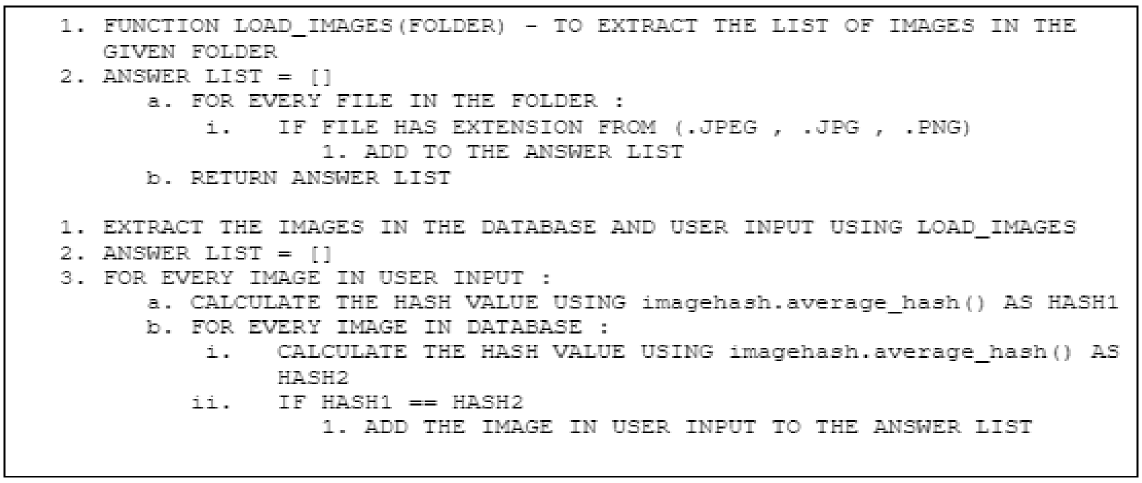
Step 1: Decreasing the image size
The initial image is compressed to some reasonable size (usually it is 8x8 or 16x16 points which will show a 64 or 256 bytes respectively).

Step2: Image grey-scaling
This move helps to decrease the hash size in three times as it describes the number of components from 3 values of RGB to one level of grey.

Step3: Computing the average value
Then the average on all the image points is calculated.

Step4: Simplifying the image
Every pixel gives a value of 0 if it is less than the average value and it gives a value of 1 when its value is greater than average. Thus, the image is converted into the set of the bits. It's read line by line, and the set of values becomes the hash.

Pseudocode for hash value computation –



```
1. FUNCTION LOAD_IMAGES(FOLDER) - TO EXTRACT THE LIST OF IMAGES IN THE
   GIVEN FOLDER
2. ANSWER LIST = []
     a. FOR EVERY FILE IN THE FOLDER :
          i.   IF FILE HAS EXTENSION FROM (.JPEG , .JPG , .PNG)
                 1. ADD TO THE ANSWER LIST
     b. RETURN ANSWER LIST

1. EXTRACT THE IMAGES IN THE DATABASE AND USER INPUT USING LOAD_IMAGES
2. ANSWER LIST = []
3. FOR EVERY IMAGE IN USER INPUT :
     a. CALCULATE THE HASH VALUE USING imagehash.average_hash() AS HASH1
     b. FOR EVERY IMAGE IN DATABASE :
          i.   CALCULATE THE HASH VALUE USING imagehash.average_hash() AS
               HASH2
         ii.   IF HASH1 == HASH2
                 1. ADD THE IMAGE IN USER INPUT TO THE ANSWER LIST
```

Figure-3: Pseudocode for Hash value computation

- **Comparison and classification -** Once the hash values are calculated, the hash value of that image is compared with hash value of all images in the database. The images whose hash value matches are classified as plagiarized and remaining as non-plagiarized.

- **Result -** The obtained data is saved in an excel sheet and the plagiarized images are saved in a word document. The same data is represented in a bar graph.

Excel Sheet – The data about the hash differences between the images that are submitted and the images that are already stored in the database are saved into an excel sheet for further reference. Each and every image has its hash difference value with all the images in the database in respective excel sheets.

Bar graph – The below graph illustrates the relationship between the hash value difference and the database images. Image of the database on x-axis and hash value difference on the y-axis, corresponding to image0.

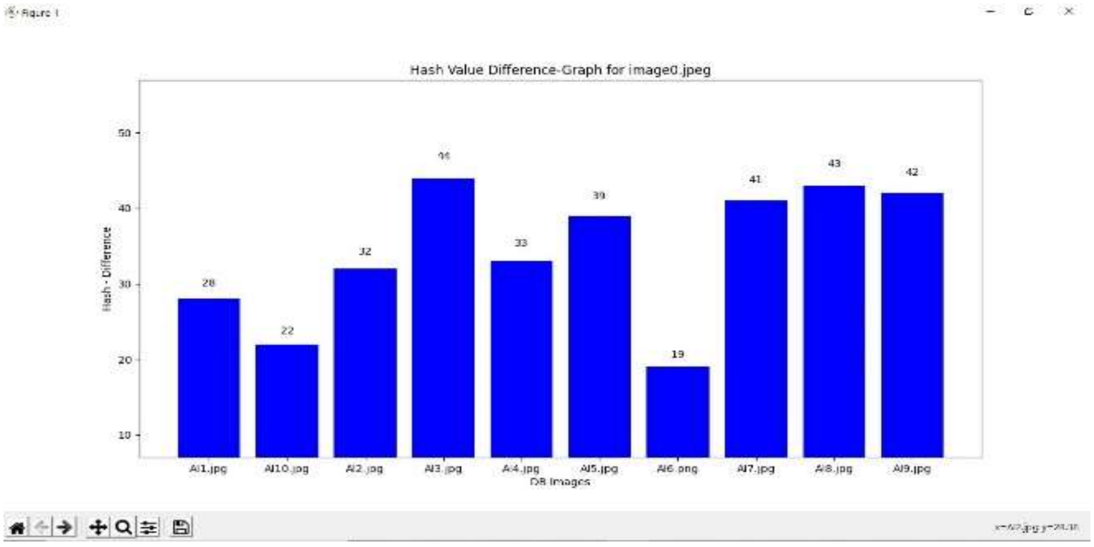
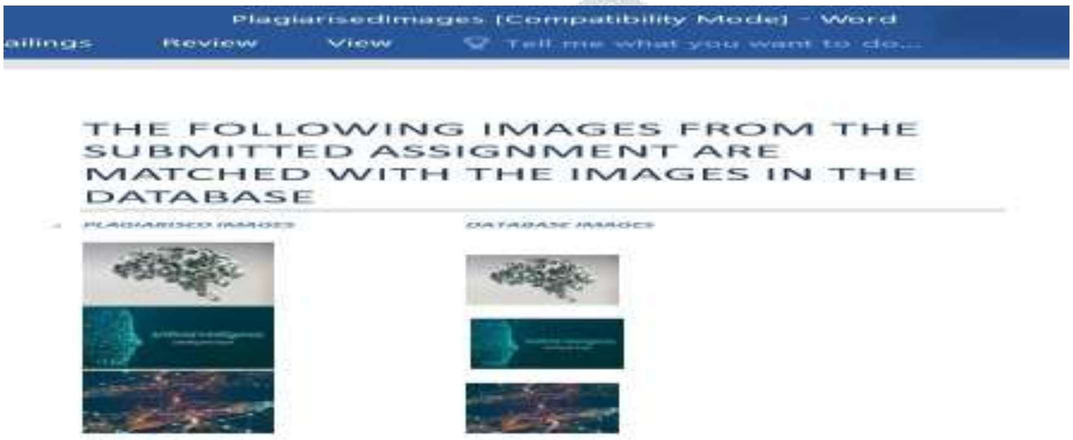


Figure4: Bar graph showing the hash value difference for image0

Resultant document –

These are the plagiarized images obtained when performed plagiarism detection on the submitted assignment document. i.e., the images that are copied and are similar to the images (for which the hash value difference is 'zero") that are already existing in a database. This is the final output of the system.


Plagiarisedimages [Compatibility Mode] - Word
ailings
Review
View
Tell me what you want to do...

THE FOLLOWING IMAGES FROM THE SUBMITTED ASSIGNMENT ARE MATCHED WITH THE IMAGES IN THE DATABASE

PLAGIARISED IMAGES DATABASE IMAGES

Figure5: Final Result Document showing the list of plagiarized images along with actual image

4. CONCLUSION

Plagiarism detection is essential for protecting the written work. It is concluded that all institutes, students and teachers must be aware of plagiarism and anti-plagiarism tools [6]. In this paper we have developed a simple system that helps in plagiarism detection of student assignments particularly for images [1] [2] [5]. It has good detection rate and can be checked for large number of assignments quickly and efficiently.

REFERENCES

[1]B. Hadi and M. J. Kargar, "Plagiarism detection of flowchart images in the texts," *2017 3th International Conference on Web Research (ICWR)*, 2017, pp. 128-132, doi: 10.1109/ICWR.2017.7959317.

[2]P. Hurtik and P. Hodakova, "FTIP: A tool for an image plagiarism detection," 2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR), 2015, pp. 42-47, doi: 10.1109/SOCPAR.2015.7492780.

[3]Ibrahin, Amirul & Khalifa, Othman & Ahmed, Diaa Eldein. (2020). Plagiarism Detection of Images. 183-188. 10.1109/SCOReD50371.2020.9250940.

[4]S, Akshay & B N, Chaitanya & Kumar, Rishabh. (2019). Image Plagiarism Detection using Compressed Images. 8. 1423-1426.

[5]Hurtik, Petr & Števuliáková, Petra. (2015). FTIP: A tool for an image plagiarism detection. 42-47. 10.1109/SOCPAR.2015.7492780.

[6]Plagscan - http://www.plagscan.com/,Plagscan

[7]https://ourcodeworld.com/articles/read/1006/how-to-determine-whether-2-images-are-equal-or-not-with-the-perceptual-hash-in-python

[8] https://7webpages.com/blog/image-duplicates-detection-python/

[9]https://en.wikipedia.org/wiki/Reverse_image_search#:~:text=Reverse%20image%20search%20is%20a,what%20formulates%20a%20search%20query

[10] https://en.wikipedia.org/wiki/Content-based_image_retrieval