

# Implementation of pHash algorithm for face recognition in secured remote online examination system

Zeba Khanam<sup>1</sup> and Mohammed Najeeb Ahsan<sup>2</sup>

<sup>1</sup>College of Computing and Informatics, Saudi Electronic University, KSA

<sup>2</sup>School of Management Studies

Sri Satya Sai University of Technology & Medical Sciences

Sehore (M.P), India.

---

## ABSTRACT

*In the rapidly growing learning environment remote online examination is the most common and efficient way of a candidate's assessment. But the online assessment systems have stakes associated with them and so their security is a major concern. There are different types of security threats associated with them like content theft, examinee cheating or copying, someone else taking the exam on behalf of the examinee. Therefore it's important to curb these threats to make these systems more reliable and credible. This paper presents a web based online examination system which uses a novel pHash algorithm for enhanced security to perform automated verification of the identity of the examinee by capturing a snapshot of him before the exam starts. The system will make automated image comparisons between the registered image of student and the sudden multi snapshots taken during the duration of the test that would reduce the probability of identity theft.*

**Keywords:** pHash, Perceptual hash functions, Face recognition techniques, Cryptographic hashing, identity infringement, Secure online examination, DCT, Perceptual hash algorithms.

---

## I. INTRODUCTION

The secured remote online examination system hits the area of student verification by using the technique of image comparison. There's a high demand for the introduction of intelligence and automation in the area of Online-exam related processes such as automated image processing and comparison. The product developed enables the students to take exams remotely and online, with real time according to their local time with the specialty of verification of presence by taking random time snapshots for students during test that may be predetermined individually by the teacher before the test starts. Therefore the important question that this paper addresses is "How to ensure that there is no identity impersonation happening at the remote exam site and the student registering for the examination is the same who is taking the exam?"

Therefore to establish and verify the identity of the examinee after random intervals during exams the system implements Face comparison algorithms to prevent cheating. This class can compare two images to find if they are similar or not.

## 2. RELATED WORK

Security related to online examination system has been researched and discussed by different researchers on different parameters such as online monitoring and proctoring, database security, authentication of examinees, usage of biometric schemes for security and many more[1][2]. Few of the biometric authentication schemes have been proposed and implemented such as finger prints, hand scanning, facial recognition etc for automatic identification of the candidate utilizing the stored or captured data. [3][4].The biometric systems come with their own pros and cons and none of them seems to have been perfect. All the scenarios lead to the fact that continuous user authentication and verification is of utmost significance. One of the works discussed, using keystroke dynamics for validating the user using biometric user authentication [5].However there were some loopholes associated with it as well as the keystroke pattern may vary depending upon the circumstances and mood of the individual. Although there are numerous techniques of biometric

identification such as iris scans and finger printing, but face recognition has received a lot of attraction from researchers spanning from different field of study such as neural networks, pattern recognition , computer graphics etc.[10][11] is proven more effective. The face recognition technique DCT has been used in different ways to compare the images. The study [11] is developed on the concept of extracting local features manually from a given frontal face. The local features represent left eye, right eye, nose and mouth. Discrete Cosine Transform (DCT) is applied to each of these local features individually as well as to the global features. The results are calculated for degree of recognition using local features and then the global features. The results depict that normalizing the image gives a better performance. Keeping all the background information into consideration this paper presents a face recognition system assessing continuously throughout the entire duration of the exam using pHash algorithms. Perceptual hash is a fast growing field of study that differentiates two dissimilar images despite the attacks like compression, cropping or skewing of images.

### **3. IMPLEMENTATION**

pHash is an open source implementation of different perceptual hash algorithms. The website is designed using HTML, CSS, JSP, PHP, etc., which does not depend on platform and can be transported to any server. Therefore the algorithm is implemented in PHP. This system employs the DCT (Discrete cosine transform) based image hash function and assesses the results. Though there are other hash functions as well that have been implemented by other scholars such as the block mean value based hash function [6] and have an edge over DCT with respect to the efficiency but DCT was well suited as per the requirements of our system . The reason for choosing this over the other hash functions is the simplicity of calculation and the system requirement that only needs to establish the identity of the same person while taking pictures at random intervals. The algorithm is robust against minor attacks like blurring, skewing, compression etc.

#### **3.1 Design and Implementation Constraints**

- The system is running successfully. Implemented on Apache Xampp server the WEB PHP and MySQL under any platform, but this system has been implemented in Windows.
- The resolution will be a major design consideration to work under different platforms.
- The photos (Face snapshot) must be formatted JPG or PNG with suitable resolution not more than 800x600 and must not exceed 100 KB.
- The registration photos reside on server storage.
- The exam verification photos at random intervals taken during exams are also uploaded and stored on the server storage but in a different folder.

#### **3.2. Image Comparison Algorithm**

In this section, the image comparison algorithm is depicted and the implementation is shown in the figure 1. It resizes the images to 8x8 square, converts the result into colored scale images, takes the color mean, and computes the hammering distance of the images' bits.

A class named compare Images is created that can compare two images to find if they are similar. It resizes the images to 8x8, converts the result into colored scale images, takes the color mean, and computes the hammering distance of the images' bits, returns an array with one and zeroes. If a color is bigger than the mean value of colors, it is one. The class returns the hammering distance of two images' bit value.

```

public function compare($a,$b)
{
    /*main function. returns the hammering distance of two images' bit value*/
    $i1 = $this->createImage($a);
    $i2 = $this->createImage($b);

    if(!$i1 || !$i2){return false;}

    $i1 = $this->resizeImage($i1,$a);
    $i2 = $this->resizeImage($i2,$b);

    imagefilter($i1, IMG_FILTER_GRAYSCALE);
    imagefilter($i2, IMG_FILTER_GRAYSCALE);

    $colorMean1 = $this->colorMeanValue($i1);
    $colorMean2 = $this->colorMeanValue($i2);

    $bits1 = $this->bits($colorMean1);
    $bits2 = $this->bits($colorMean2);

    $hammeringDistance = 0;

    for($a = 0;$a<256;$a++)
    {
        if($bits1[$a] != $bits2[$a])
        {
            $hammeringDistance++;
        }
    }

    return $hammeringDistance;
}

```

**Fig 1. Function `compare()` from the class `CompareImages` for computing the hammering distance between two images.**

Perceptual hash algorithms describe a class of comparable hash functions [6]. Features in the image are used to generate a distinct (but not unique) fingerprint, and these fingerprints are comparable. They usually use distance or similarity functions to compute the difference between the perceptual hash values of two images. There are four perceptual hash functions that are usually used to perform image verification.

Perceptual hashes are a different concept compared to cryptographic hash functions like MD5 and SHA1. With cryptographic hashes, the hash values are random. The data used to generate the hash acts like a random seed, so the same data will generate the same result, but different data will create different results. Comparing two SHA1 hash values really only tells you two things. If the hashes are different, then the data is different. In addition, if the hashes are the same, then the data is likely the same. (Since there is a possibility of a hash collision, having the same hash values does not guarantee the same data.) Therefore Cryptographic hash functions are good for non multimedia contents like exe files but when verifying a multimedia object there are different representations possible of a single multimedia object such as an image that can be achieved using various editing techniques such as resizing or cropping. In contrast, perceptual hashes can be compared -- giving you a sense of similarity between the two data sets.

Every perceptual hash algorithm more or less has the same basic properties: images can be scaled larger or smaller, have different aspect ratios, and even minor coloring differences (contrast, brightness, etc.) and they will still match similar images. In this implementation, a few refactorings too have been performed depending upon the scenario to improve the code [7] [8] and efficiency. The code quality has been enhanced but the performance evaluation and improvements are beyond the scope of this paper. We intend to re implement the algorithm using TDD and incorporate changes in the code so as to improve its performance and validate it on the basis of various parameters like the speed of execution etc [9][12] [14] but would be assessed in the next paper in line as a future work on the same algorithm.

With pictures, high frequencies give the detail, while low frequencies depict the structure. A large, detailed picture has many high frequencies. A very small picture lacks details, so it is all low frequencies. An alternative approach to face recognition proposed in the past [13] aims on locating interesting features in a face by Gabor filters. Then the filtered image is multiplied with a 2D Gaussian to focus on the center of the face and avoid extracting features at face cage is then searched for peaks, which are called feature points used for recognition. To show how the Average Hash algorithm works,

Algorithm is used by pHash. The pHash approach extends the average approach to the extreme, using a discrete cosine transform (DCT) to reduce the frequencies. The mechanism is depicted as follows:

1. **Size and Color reduction.** The pHash works with small images and grayscale. Preferably, the image is supposed to be 32x32 that is considered as optimal and should be larger than 8x8; The advantage of having this size of image is to ease the complexity involved with DCT. Similarly the grayscale is used to further simplify the
2. **Direct Cosine Transform Calculation and reduction.** The JPEG image type primarily uses an 8x8 DCT. But in this algorithm a 32x32 DCT is employed. The DCT partitions the image as a collection of scalars and frequencies. The the top-left is maintained at 8x8 and DCT at 32x32. The lowest frequencies are represented by the corners.
3. **Mean DCT computation.** The mean value of DCT is to be computed next. (Low-frequency values of DCT 8x8 by omitting the first term since there can be a significant difference in the DC coefficient and so the average won't be a fitted one).
4. **Reduction of the DCT once again.** The 64 hash bits are to be adjusted to 0 or 1 depending on their mean value. If 64 DCT value > mean set it to 0 else set to 1. The result only depicts the approximate relative scale of the frequencies to the average and doesn't project the actual low frequencies; therefore the result won't be altered if there is no variation in the overall structure of the image; this computation can easily endure the color histogram variations.
5. **Computing the Hash.** Set the 64 bits into a 64-bit integer. The order does not matter, just as long as you are consistent. To see what this fingerprint looks like, simply set the values (this uses +255 and -255 based on whether the bits are 1 or 0) and convert from the 32x32 DCT (with zeros for the high frequencies) back into the 32x32 image. As with the Average Hash, pHash values can be compared using the same Hamming distance algorithm. (Just compare each bit position and count the number of differences.)

The algorithm is well implemented and has been tested with 20 persons and 10 samples/subject with different variations in facial expressions, the age group lying between 18 to 40 years, illumination levels and positions. The application of the above algorithm resulted in 98% recognition rate with the recorded features.

#### 4. CONCLUSION AND FUTURE WORK

This paper presents a face recognition system implemented using pHash algorithm incorporated in online examination software that includes a question bank and an examination engine that creates the exam, selects the exam questions randomly, shuffles exam questions, and runs exam logic with specific duration that for an enhanced security keeps taking random snapshots of the examinee during the duration of the exams. The system verifies the examinee picture by comparing his image at exam time with his image at registration time. This comparison has been done using a comparison module that employs the pHash algorithm and it produced satisfactory results while being tested by several candidates.

The target has been achieved to a satisfactory extent and we still aim to enhance the accuracy and performance of the algorithm by applying various techniques and novel development strategies and evaluate the results with the current implementation.

#### REFERENCES

- [1] Wahid, Abdul & Sengoku, Yasushi & Mambo, Masahiro. (2015). Towards constructing a secure online examination system. 10.1145/2701126.2701203.
- [2] Yang, M., Kriegman, D. and Ahuja, N. (2002) Detecting Faces in Images: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24, 34-58.
- [3] Fayyumi, A. and Zarrad, A. (2014) Novel Solution Based on Face Recognition to Address Identity Theft and Cheating in Online Examination Systems. Advances in Internet of Things, 4, 5-12
- [4] Yang, S. and Verbauwhede, I. (2003). A Secure Fingerprint Matching Technique. Proceedings of the 2003 ACM SIGMM workshop on Biometrics Methods and Applications, California, 89-94. <http://dx.doi.org/10.1145/982507.982524>
- [5] Flior, E. and Kowalski, K. (2010) Continuous Biometric User Authentication in Online Examinations. Seventh International Conference on Information Technology IEEE Computer Society, Las Vegas, 12-14 April 2010, 488-492.
- [6] Christoph Zauner. (2010) Implementation and Benchmarking of Perceptual Image Hash Functions. (thesis).
- [7] Rizvi, S A M. and Khanam, Z. (2011) A methodology for refactoring legacy code. Proceedings of International Conference on Electronics Computer Technology (ICECT 2011), pp. 198-200 (2011), IEEE Xplore.

- [8] Khanam, Z. ( 2018) Barriers to Refactoring: Issues and Solutions . February 18 Volume 4 Issue 2 , International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRSCE), PP: 232 – 235
- [9] Zeba Khanam and Mohammed Najeeb Ahsan. (2017) Evaluating the Effectiveness of Test Driven Development: Advantages and Pitfalls. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 18 (2017) pp. 7705-7716.
- [10] Ziad M. Hafed. (2001) Face Recognition Using DCT, International Journal of Computer Vision, 2001, pp. 167-188.
- [11] R. A Chadha, P. P. Vaidya, M. Mani Roja.(2011) Face Recognition Using Discrete Cosine Transform for Global and Local Features, Proceedings of the 2011 International Conference on Recent Advancements in Electrical, Electronics and Control Engineering (IconRAEeCE) IEEE Xplore: CFP1153R-ART; ISBN: 978-1-4577-2149-6.
- [12] Khanam, Z. and Rizvi, S.A.M. (2013) Aspectual Analysis of Legacy Systems: Code Smells and Transformations in C, I.J.Modern Education and Computer Science, 2013, 11, 57-63 Published on November 2013 in MECS.
- [13] Hjelmas E. (2001) Feature-Based Face Recognition, Department of Informatics, University of Oslo, pp. 1-5.
- [14] Zeba Khanam and SAM Rizvi (2014) Assessment of the Impact of Aspect Oriented Programming on Refactoring Procedural Software, Computers and Mathematics in Automation and Materials Science, MATHIMA-24, Cambridge, USA.