

# A Perceptual Matching based Deduplication Scheme using Gabor-ORB Filters for Medical Images

Sonal Ayyappan<sup>1</sup>

Research Scholar

Department of Computer Science and Engineering  
SRMIST, Chennai, India

Dr. C Lakshmi<sup>2</sup>

Professor and HOD

Department of Software Engineering  
SRMIST, Chennai, India

**Abstract**—In the ever widening field of telemedicine, there is a greater need for intelligent methods to selectively choose data that are relevant enough to be transmitted over a network and checked remotely. By the very nature of medical imaging, a large amount of data is generated per imaging or scanning session. For instance, a Magnetic Resonance Images (MRI) scan consist of hundreds to thousands of images related to slices of the organ being scanned. But at often times all of these slices are not of interest during the process of medical diagnosis by the medical practitioner. Not only does this result in the access of unwanted data remotely, but it can also put greater strain on the bandwidth available over the network. If the relevant images can be selected automatically without human intervention, ensuring great sensitivity, the above-mentioned issues can also be alleviated. This paper proposes a novel method of perceptual matching and selection of relevant MRI images by using a deduplicating technique of combining Gabor filter with Oriented FAST and Rotated BRIEF (ORB) feature extraction technique on a vast set of MRI scan images. The outcome of this method are relevant deduplicated MRI scan images which can save the bandwidth and will be easy for the medical practitioner to verify remotely.

**Keywords**—Perceptual matching; ORB feature extraction; Gabor filters; MRI scan; deduplication

## I. INTRODUCTION

Advanced data frameworks have been progressively conveyed in the recent medical care scenarios. Indeed, numerous medical clinics and hospitals depend on medical clinic data frameworks (HIS), radiology data frameworks (RIS), and picture documenting and correspondence frameworks (PACS), for storage of MRI Scan Medical images. These frameworks facilitate the practitioners to share images [1][2][3]. Data Deduplication is a method of eliminating repeated copies of data in order to retain the storage capacity. It results in decreased cost per gigabyte with more area to store data. In radiology centers, more than petabytes of medical images are stored every year which may contain redundant data too. This increase of repeated data cannot be handled by the existing IT technologies [23]. In an MRI Scan collection of images for a single subject regarding any organ, views are taken from three directions. Each direction concentrates on taking images of slices of the organ with slight differences like 1mm, 4mm, 6mm etc. For a non suspicious subject, there may be more irrelevant repeated data which when stored will waste a lot of storage space. By the present nature of MRI scans of a single subject, relevant details are not distributed evenly in equidistant slides. Hence there is a need for perceptual matching and deduplication for transmitting medical images among

practitioners for expert opinions[1]. By combining Gabor filter, ORB key-point detection and Brute-Force Matcher, we achieve this perceptual deduplication where relevant data is extracted without any specific regard for the distances between the slices. A finely tuned Gabor filter provides the ORB algorithm with just the right amount of details that leads to the most optimal matching differences.

This method, being based on ORB, has got excellent performance in identifying and matching near-similar images even if a one-to-one positional correspondence between the pixels are absent. Therefore, variations like missing regions, appreciable level of orientation difference, size difference, etc. can be accounted.

In this paper the method for fine tuning the parameters of Gabor Filter is implemented. It also describes the method of feature extraction using ORB and brute force matching technique which assures the performance of perceptual matching so as to identify repeated data and discard it. Section 2 give an overview of the work done. The rest of the paper is organized as follows. Section 3 explains the algorithms used for finding out the deduplicated images and the background where this method has its relevance. In Section 4, results are discussed. The paper is concluded in Section 5.

## II. LITERATURE REVIEW

Storage used for medical imaging must be expandable in the coming years. It is normal for an incoming of Terabytes of medical imaging data per year for hospital radiology departments [4]. Deduplication is a very important aspect for the reliability of storage facilities, as the expense of storage management can be minimized by removing duplicated files [5]. A huge amount of memory is wasted in storing the redundant data for a single patient in a single take [24].

However, deciding whether two original files are the same or not by viewing two processed images is not easy. Message-locked encryption was suggested to facilitate the deduplication of processed image. This approach lets users produce the same ciphertext for the same file and enables users to reap the benefits of ciphertext deduplication [6]. Techniques had been proposed for the deduplication of encrypted data in different levels. To support the deduplication feature for partially duplicated files, block-level deduplication methods were studied [7]. Cloud media centers were implemented for dedicated deduplication system where the older versions were not working [8]. Near-duplicate data scanning for encrypted

data has been studied. A lot of wastage in the storage system occurs due to these duplicates which is very costly to afford [9].

Deduplication techniques are characteristics based implemented. Characteristics like extracting features, using hash technique for indexing extracted image feature, image similarity detection based on distance using threshold, etc were used to find almost equal images. In order to analyze the exact duplicates or the near exact duplicates different features extraction algorithms like Scale Invariant Feature Transform, Speed Up Robust Feature, Principal Component Analysis, Binary Robust Independent Elementary Features were taken into account [10].

The authors in [11] used Difference of Gaussians, Principal Component Analysis with Scale Invariant Feature Transform methods for feature extraction of images and space efficient bloom filter were hashed with these features. The locality sensitive hashing used correlated attributes to find the similar images. This method gave efficient bandwidth and saved storage resources. Later in [12] the authors used perpetual hash algorithms for creating image signatures which identified similar duplicated images. It resulted in deduplicating storage and saving bandwidth while transmission. Another method used was MapReduce technique. It manages data in huge quantities in a distributed manner. The authors in [13] used this technique resulting in a faster image duplication identifier. SIFT methods were made strong by incorporating k-means algorithm and groups of multiple image clusters. This helped to detect the near duplicate images and those were hashed based on histogram distance [14]. Another research was with Local-based binary representation which made use of binary vector and histogram for finding out the duplicate images [15]. Later a real time novel method which made use of Bloom filters along with the existing techniques was implemented. It used the correlation property and resulted in reduced latency processing [16].

A faster and more efficient feature point detector than SIFT and SURF named Oriented FAST (Features from Accelerated Segment Test) and Rotated BRIEF(ORB) technique is nowadays used widely [17]. It has advantages of low computation cost and better performance [18]. Medical images can be exchanged in a very efficient way using the cloud [25]. It has the limitation of numerous applications pointing to the same data at the same time. Due to the increasing need of storage capacity, the PACS also have its own limitations [26]

### III. BACKGROUND

A complete MRI scan image set is retrieved from the MRI scanner and used by a medical practitioner for medical diagnosis. But in case an expert opinion is required from another medical practitioner, the files are first deduplicated to select only the relevant slides. This subset of the scan image set is then encrypted using a pixel-scrambling algorithm which uses chaotic maps and intensity variations[27]. On the receiver end, the files are decrypted to retrieve the deduplicated set of scan images. Due to the deduplication method applied, the bandwidth for the transferring of medical images among practitioners and the storage for storing the relevant data can be increased for some more time. In Fig. 1 the graphical representation of the above said scenario has been explained.

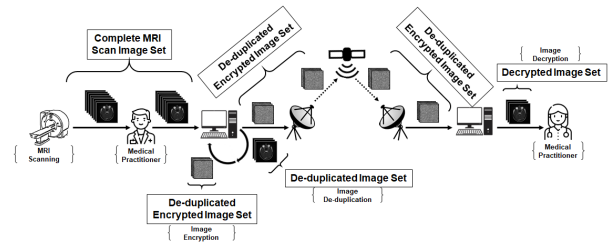


Fig. 1. Graphical Representation of the Use-Case Scenario of MRI Scan Image Deduplication.

#### A. Algorithms

Input images are chosen from complete MRI scan image set with very less slice width. Images 1 mm apart with no perceptible differences are chosen as similar images, and images 5 mm apart with some perceptible differences of valid nature are selected as dissimilar images for the execution of this method. Following are the algorithms used to find the most optimal Gabor filter parameter set. Gabor filter has parameters which needs to be fine tuned individually for this deduplication method.

---

#### Algorithm 1: Build\_filters(image1, image2)

---

**input :** ksize, sigma, theta, lambda, gamma

**output:** orb\_values, Orb\_match.csv

```

for ksize in [3,31] do
    for sigma in range(1,20), step-size: 1 do
        for theta in range(0, pi), step-size: pi/16 do
            for lambda in range(10,100), step-size: 10 do
                do
                    for gamma in range(0.1,1.2), step-size: 0.2 do
                        Initialize Gabor filter kernel with
                        getGaborKernel(ksize, sigma,
                        theta,lambda, gamma, 0) → kernel
                        Apply filter by calling
                        filter2D(image1/image2,kernel) →
                        fimage1, fimage2.
                        Call orb(fimage1, fimage2) →
                        KeypointA, KeypointB, orb_match
                        Call Sklearn struc-
                        tural_similarity(fimage1,fimage2)
                        → ssim
                        Append (Ksize, sigma, theta,
                        lambda, gamma, KeypointA,
                        KeypointB, ssim, orb_match) to
                        orb_values list.

```

```

Create pandas dataframe orb_df from orb_values
Save orb_df as csv file Orb_match.csv

```

---

Algorithm 1 explains the procedure for fine tuning the parameter set for building the Gabor filter. Using these parameters Algorithm 3 sets the Gabor filter. Algorithm 2 provides the base condition for iterating ORB feature matching technique. The perpetual matching is carried out using Algorithm 4.

**Algorithm 2: Base Parameter Iteration**

**input :** (image1, image2), (image2, image3)  
**output:** orb\_values, Orb\_match.csv

Set default value for ksize, sigma, lambda, gamma from chosen\_parameters.  
 Choose one parameter out of ksize, sigma, lambda, gamma to iterate → chosen\_parameter.  
**for** chosen\_parameter in chosen\_parameter\_list **do**  
   Build gaborfilter kernel using  
    $build\_filter(defaultparameters, chosen\_parameter) \rightarrow filter\_list$   
   Apply filter by calling  
    $process(image1/image2/image3, filter\_list) \rightarrow image1, image2, image3.$   
   Call orb(image1, image2) → SKeypointA, SKeypointB, Sorb\_match.  
   Call orb(image2, image3) → DKeypointA, DKeypointB, Dorb\_match.  
   Find match\_difference = Sorb\_match - Dorb\_match  
   Append default parameters, chosen parameter, (S/D)Keypoint(A/B), Sorb\_match, Dorb\_match to orb\_values list.  
 Create pandas dataframe orb\_df from orb\_values  
 Save orb\_df as csv file Orb\_match.csv

**Algorithm 3: build\_filter(ksize, sigma, lambda, gamma)**

**input :** ksize, sigma, lambda, gamma  
**output:** filter\_list

**for** theta in range (0, pi), step-size: pi/16 **do**  
   Initialize Gabor filter kernel with  
    $getGaborKernel(ksize, sigma, theta, lambda, gamma, 0) \rightarrow kernel$   
   Append kernel to filter\_list  
 Return filter\_list.

**Algorithm 4: ORB Extraction and Brute-Force Matching**

**input :** ORB Extraction and Brute-Force Matching  
**output:** Matching\_Percentage

Convert input images to gray-scale images → (img1, img2)  
 Create ORB object using ORB\_create(nfeatures = 1000, scoreType = FAST)  
 Detect key-points from input images, detect(img1, img2) → (kp1, kp2)  
 Compute ORB descriptors from input images and key-points → (kp\_1, desc\_1, kp\_1, desc\_2)  
 Create BFMatcher object with Hamming distance parameter.  
 Find matches between the images using descriptors, match(desc\_1, desc\_2) → matches[]  
 set lower = minimum number of key-points.  
 Matching\_Percentage = sizeof(matches)/lower \* 100  
 return Matching\_Percentage.

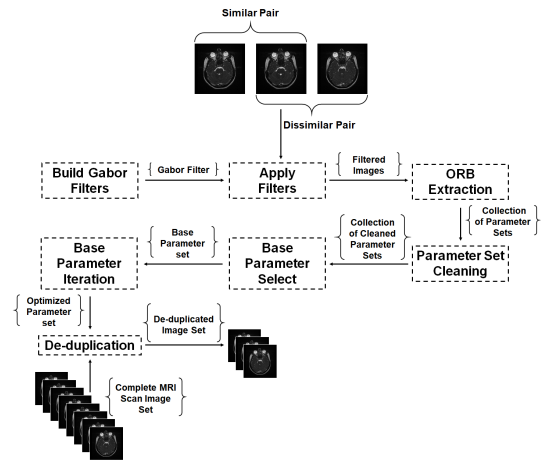


Fig. 2. Steps involved in the Proposed Deduplication Algorithm (1. Build Gabor Filters, 2. Apply Gabor Filters, 3. ORB Extraction, 4. Parameter Set Cleaning, 5. Base Parameter Select, 6. Base Parameter Iteration, 7. De-duplication)

The resultant of these algorithms are a set of nearly matched MRI images which can be regarded deduplicated and can be discarded before transmitting.

**B. Working**

Building Gabor filters and its application, ORB feature extraction, parameter set cleaning, base parameter set selection and iteration, deduplication are the key phases in the proposed deduplication scheme. This scheme is graphically represented in the Fig. 2.

The prevailing section deals with each step of working of the whole deduplication procedure.

1) *Build Gabor Filters:* Gabor filters are special classes of band-pass filters which allow certain bands of frequency and reject the rest. The Gabor filters are well known for its time and frequency characteristics. Filters with distinct scaling directions can be constructed using Gabor filters caused by different parameters[19]. In image processing, they are used for feature extraction, edge detection, texture analysis, etc. The Gabor function has the capability to capture the localized information with respect to spatial frequency, location, and selection of direction [20]. Mathematically Gabor filters [21,22]are expressed as a function:

$$g(x, y; \lambda, \theta, \phi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i(2\pi \frac{x'}{\lambda} + \phi)\right)$$

$$x' = x \cos \theta + y \sin \theta$$

$$y' = -x \sin \theta + y \cos \theta$$

Build\_filter function iterates through all Gabor filter parameters and create kernels for each set of parameters. Range of values for each parameter, namely, Ksize(x,y), Sigma(σ), Theta(θ), Lambda(λ) and gamma(γ) are predefined during this phase. These ranges are as follows: Ksize from the set [3,31], Sigma from 1 to 10 with a step size of 1, Theta from 0 to π with a step size of π/16, Lambda from 10 to 100 with a step size of 10 and Gamma from 0.1 to 1.2 with a step size of 0.2.

2) *Apply Gabor Filters*: Three images are selected to form two pairs of images, where one image is common to both. First pair is formed from MRI images that are taken 1 mm apart and represents similar images. Whereas second pair represents dissimilar images and are taken 5 mm apart. Pairs of similar and dissimilar images are read into the program and they are passed to the `apply_filter` function. Both images are filtered using the kernels and filtered Gabor images are formed.

3) *ORB Key-point Extraction*: These images are compared using ORB key-point extraction and brute-force matcher (BFMatcher). Orb function returns the number of key-points found in each image and matching percentage between them. Structural similarity index is also found between the images. These values along with Gabor filter parameters are saved into a CSV file for further processing.

4) *Parameter Set Cleaning*: The CSV file is cleaned so as to extract valid set of parameters. The cleaning process is as follows: any parameter set which gives very low number of key-points are discarded. All sets which generates very low or very high matching percentages are also discarded. Any set that produces abnormally high structural similarity index is discarded since high structural similarity index would indicate high loss of details in the images. A high minimum-threshold number of key-points in either images indicate existence of appreciable level of details.

5) *Base Parameter Select*: Parameter sets are sorted in the descending order of their matching percentages. First few parameter-sets with the highest matching percentages are retained and rest are discarded. These parameters are used to find matching percentages for dissimilar pair of images.

Difference between matching percentages of similar and dissimilar pairs are found for each parameter-set. These parameters along with key-point counts, matching percentages and matching percentage differences are stored as a CSV file. They are then sorted in the descending order of their matching percentage differences. And parameter set with the highest difference is selected as the base or default parameter set for further steps.

This parameter set represents the maximum gap between matching percentages of similar and dissimilar images. Therefore, it is the best provisional choice to decide whether two images are similar or not.

6) *Base Parameter Iteration*: Keeping the base parameter-set as the default values for each parameters, one parameter is chosen at a time to be iterated over a range of values which are characteristically valid for that particular parameter except for Theta. For every iteration, 16 Gabor filter kernels with Theta value ranging from 0 to  $\pi$  with a step size of  $\pi/16$  are formed and each set of 16 images are combined to form the new pair. ORB key-point extraction and Brute-Force matching is performed on this new pair of images. Metrics like key-point counts, structural similarity, matching percentages and matching percentage differences are found and stored for further analysis.

Values of each parameter that gave the highest match differences in their respective iterations were selected and used to form a final parameter-set. This was further used to find the matching differences between similar and dissimilar images.

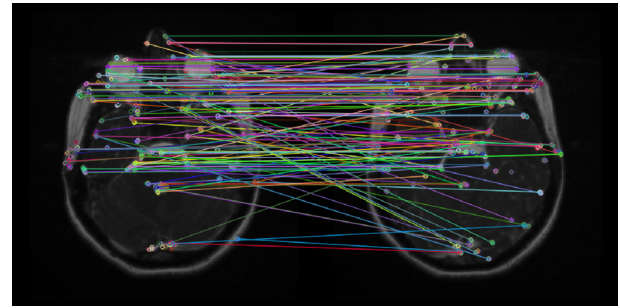


Fig. 3. Brute-Force Matching of ORB Key-Points on Filtered Images

This is done to check if parameter values that individually produced the best matching difference along with other default values could produce better results when collected together as a single parameter set. Depending on the result, it was either retained or discarded.

7) *Deduplication*: After finding the best case parameter set for Gabor filter, it can now be used for deduplication of the entire MRI Scan Image Set. The filter now being fine-tuned for MRI scans, we can set well defined thresholds when two images are similar or dissimilar. By running this in an iterative manner, the entire set is deduplicated and only relevant slides or images are retained.

The result of finding out feature matching points using ORB feature extracting and Brute Force matcher is shown in Fig. 3.

#### IV. RESULT AND DISCUSSION

All the above mentioned steps were tried using SIFT, SURF and ORB key-point extractions and their preferred matchers. It was found that ORB provided the best-case efficiency in both matching performance and execution time. Hence ORB along with Brute-Force matcher was selected as the algorithm/technique to be used for further analysis in the paper.

Matching performance evaluation is done using the number of ORB key-point extracted and 'Match Differences'.

$$M = S - D \text{ Match Difference} = S - D \text{ where, } M = \text{Match Difference} \\ S = \text{Matching percentage between similar images} \\ D = \text{Matching percentage between dissimilar images}$$

Results of the iterated base parameter set for fine tuning each parameter of the Gabor filter is as follows.

##### A. Ksize

K-size is the size of Gabor-filter kernel. Here K-size of 31 represents a kernel of size 31x31 pixels. In order to improve time-efficiency of initial iterations, just a single pair of ksize values were selected. This pair (3,31) represents two possible extremes of values that k-size could take. In later iterations along with a variable k-size and default values for other parameters, it was noted that the key-point count and matching difference stabilized after k-size value of 13. Fig. 4 shows graphically this iterations. Hence the assumption was validated that time-efficiency was improved by choosing two extremes



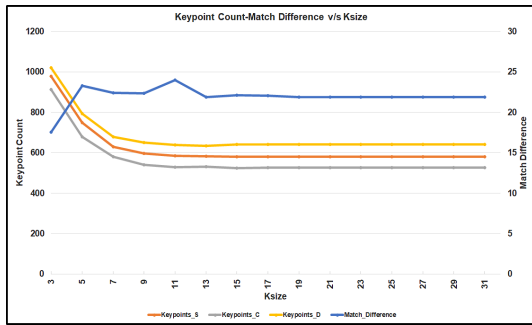


Fig. 4. Variation of Matching Difference and Key-Point Count vs the Iteration of Ksize

TABLE I. MATCH DIFFERENCES WITH VARIATION OF KSIZE

ksize	KeyPtS	KeyPtC	KeyPtD	MatchDiff
3	978	912	1019	17.54386
5	748	678	792	23.30383
7	630	581	678	22.37522
9	597	541	650	22.36599
11	585	529	638	24.00756
13	583	530	635	21.88679
15	581	525	640	22.09524
17	580	526	640	22.05323
19	580	526	640	21.86312
21	580	526	640	21.86312
23	580	526	640	21.86312
25	580	526	640	21.86312
27	580	526	640	21.86312
29	580	526	640	21.86312
31	580	526	640	21.86312

of values rather than an extensive range. It also shows that matching difference is sensitive to k-size individually for a limited range values and the default k-size value that gave the best overall performance could very well be outside the high sensitivity range.

Table I gives a summary of the similar keypoint count, dissimilar keypoint count, total matched keypoints and match differences with varying ksize in the Gabor filter. The iteration was done with an increase of 2 starting from 3 to 31. Above this, the results were repeating and constant.

**B. Sigma**

Sigma represents the bandwidth of the Gabor envelop. A higher sigma values increases the overall size of the envelop

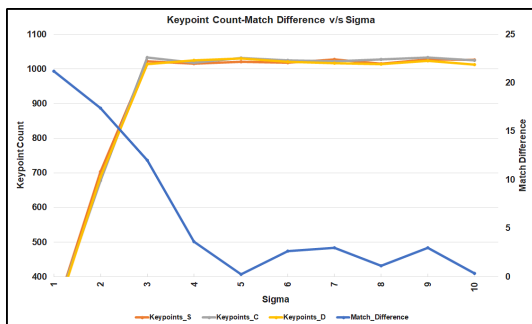


Fig. 5. Variation of Matching Difference and Key-Point Count vs the Iteration of Sigma

TABLE II. MATCH DIFFERENCES WITH VARIATION OF SIGMA

Sigma	KeyPtS	KeyPtC	KeyPtD	MatchDiff
1	580	526	640	21.86312
2	285	290	274	21.20246
3	704	678	689	17.40413
4	1022	1033	1014	12.00087
5	1015	1018	1025	3.641739
6	1021	1032	1030	0.227647
7	1018	1025	1021	2.631381
8	1028	1022	1017	3.002192
9	1016	1028	1014	1.121504
10	1028	1033	1023	2.980191

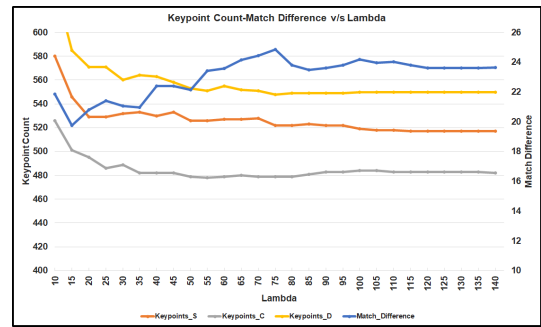


Fig. 6. Variation of Matching Difference and Key-Point Count vs the Iteration of Lambda

and allows for more strips (frequency bands) through it.

It was noted that the matching difference for medical scan images are highest for low values for sigma. This is despite a sudden increase in the key-point count as is shown from the graph. This can be clearly seen in Fig. 5. As can be noted, key-point count stabilizes after sigma of 3, but matching difference has got an overall downward trend and keep fluctuating at higher values of sigma. This is due to larger amount of details that is passed through the Gabor filter for higher sigma values. This results in adversely affecting matching performance even though there are more features to extract key-points from. Hence matching performance is highly sensitive to sigma. Sigma variation also has the greatest effect on perceptual state of the output image. Table II shows the iterated values for fixing the sigma value.

**C. Lambda**

Lambda or wavelength governs the width of the Gabor function. Higher lambda values produce thicker Gabor strips. As can be noted from Fig. 6, there is appreciable improvement in matching difference as lambda increases till a specific level. Meaning increasing amount of details in the image due to wavelength change is constructive towards better matching performance. But beyond the point of optimality, more details work against the match and hence do not produce the best possible matching difference. But overall stability of matching difference is quite high for any higher values for lambda after the point of optimality. Table III shows the iterated values for stabilizing the lambda values

**D. Gamma**

Gamma determines the aspect ratio of the Gabor filter. And it controls the height of the function. A very high

TABLE III. MATCH DIFFERENCES WITH VARIATION OF LAMBDA

lambda	KeyPtS	KeyPtC	KeyPtD	MatchDiff
10	580	526	640	21.86311787
15	546	501	585	19.76047904
20	529	495	571	20.80808081
25	529	486	571	21.39917695
30	532	489	560	21.06339468
35	533	482	564	20.95435685
40	530	482	563	22.406639
45	533	482	558	22.406639
50	526	479	553	22.12943633
55	526	478	551	23.43096234
60	527	479	555	23.5908142
65	527	480	552	24.16666667
70	528	479	551	24.42588727
75	522	479	548	24.8434238
80	522	479	549	23.79958246
85	523	481	549	23.49272349
90	522	483	549	23.60248447
95	522	483	549	23.80952381
100	519	484	550	24.17355372
105	518	484	550	23.96694215
110	518	483	550	24.01656315
115	517	483	550	23.80952381
120	517	483	550	23.60248447
125	517	483	550	23.60248447
130	517	483	550	23.60248447

TABLE IV. MATCH DIFFERENCES WITH VARIATION OF GAMMA

Gamma	KeyPtS	KeyPtC	KeyPtD	MatchDiff
0.1	22	26	17	17.9144385
0.3	302	290	296	2689655172
0.5	580	526	640	21.86311787
0.7	749	691	783	23.01013025
0.9	879	809	901	18.41779975
1.1	944	871	963	19.17336395
1.3	962	928	988	21.01293103
1.5	1017	957	1011	19.33124347
1.7	1025	994	1038	17.90744467
1.9	1031	997	1022	19.85957874
2.1	1052	1038	1051	19.46050096
2.3	1021	1026	1060	17.86375014
2.5	1043	1037	1024	18.07817774
2.7	1039	1042	1042	18.17329523
2.9	1031	1047	1046	19.39734391
3.1	1026	1037	1022	19.48650164
3.3	1032	1002	1025	18.76247505
3.5	1039	1021	1038	17.82566112
3.7	1052	1030	1052	17.96116505
3.9	1045	1029	1056	18.17298348

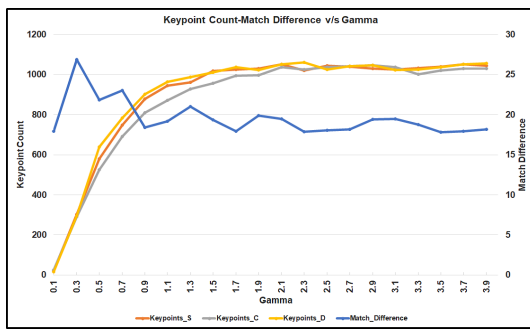


Fig. 7. Variation of Matching Difference and Key-Point Count vs the Iteration of Gamma

gamma results in lower height of the filter, while a lower value increases the height. For a series of increasing gamma values, a seesaw behavior in the matching difference was observed. For lower number of key-point counts, the matching difference appreciated with an increasing gamma, while at higher ranges with more details in the images, it showed depreciation followed by stability. Table IV shows the iteration for finding the gamma value for Gabor filter and is graphically shown in Fig. 7.

### E. Discussion

MRI Scan Images of a subject was taken at two different periods live from a radiology center. The subject under experiment was suffering from brain tumor while diagnosed. This paper deals with a method where the subject or the patient has undergone a treatment and it on regular follow-ups. Thus, this methods from all the above observations apparently justifies that MRI scan images of this particular type works very well within a narrow set of values of filter parameters. And hence it validates the need for an algorithm to fine tune filter parameters to the application, as is proposed in this paper. When each parameter value from their individual point of

optimality was clubbed together to determine the matching difference, it was noted that it faired considerably inferior to the initial base parameter set. Hence it is clear that the matching performance is sensitive to all the parameters as a whole. The base parameter-set obtained provided the best performance and could be used for deduplication of medical scan images of this particular kind reliably well. Bench mark datasets were not used as it didn't provide a whole set of MRI images for a particular subject and at two or more different periods for the same subject.

### V. CONCLUSION

In the real-world conditions where telemedicine is getting more prevalent day by day, MRI scan images needs to be transmitted over open network. This puts a considerable strain on the bandwidth and security requirement since the data is considerably large in size and highly confidential in nature. Hence by using the deduplication scheme suggested in this paper, the number of scan images can be reduced by selecting a subset of almost similar scan images. This subset would only contain scan images which contain relevant details and hence redundancy is avoided. Using chaotic maps that ensure security over data transmission, each image can be encrypted. Thus tackling two of the major challenges of telemedicine successfully.

### REFERENCES

- [1] J. Fiaidhi, C. Kuziemsy, S. Mohammed, J. Weber, T. Topaloglou. *Emerging IT trends in healthcare and well-being*, 18th ed, vol 3, pp 9-13, IT Prof, 2016
- [2] N.Saranummi *In the spotlight: health information systems* 6th ed, pp 21- 23, IEEE Rev Biomed Eng, 2013.
- [3] H.K Huang *PACS and imaging informatics: basic principles and applications* New York: Wiley-Blackwell,2010.
- [4] <https://searchhealthit.techtarget.com>
- [5] M W Storer, K Greenan, D D E Long, E L Miller, *Secure data deduplication* pp 1 - 10, 4th ACM International Workshop on Storage Security and Survivability; ACM: New York, NY, USA, 2008.
- [6] M Bellare, S Keelveedhi, T Ristenpart, *Message-locked encryption and secure deduplication*, Annual International Conference on the Theory and Applications of Cryptographic Techniques; Springer: Berlin/Heidelberg, Germany, 2013.

- [7] R Chen, Y Mu, G Yang, F Guo, *BL-MLE: Block-Level Message-Locked Encryption for Secure Large File Deduplication*, vol 10, pp 2643 - 2652, IEEE Trans. Inf. Forensics Security, 2015.
- [8] Y Zheng, X Yuan, X Wang, J Jiang, C Wang, X Gui, *Toward Encrypted Cloud Media Center with Secure Deduplication*, vol 19, pp 251 - 265, IEEE Trans. Multimedia, 2017.
- [9] H Cui, X Yuan, Y Zheng, C Wang, *Enabling secure and effective near-duplicate detection over encrypted in-network storage*, pp 1 - 9, IEEE INFOCOM 2016—The 35th Annual IEEE International Conference on Computer Communications; IEEE: New York, NY, USA, 2016.
- [10] R Kaur , I Chana , J Bhattacharya *Data Deduplication Techniques for efficient cloud storage management: a systematic review*, vol 74, pp 2035 -2085, Journal of SuperComputing, 2018.
- [11] Y Hua , W He, X Liu , D Feng *SmartEye: real-time and efficient cloud image sharing for disaster environments*, pp 1616 -1624, IEEE Conference on Computer Communications, 2015.
- [12] X Li, J Li, F Huang *A secure cloud storage system supporting privacy-preserving fuzzy deduplication* vol 20(4), pp 1437 - 1448, Journal of Soft Computing, 2016.
- [13] A S Deshmukh, P D Lambhate *A methodological survey on mapreduce for identification of duplicate images*, vol 5(1), pp 2016 - 210, International Journal of Scientific Research, 2016.
- [14] Z Li, X Feng. *Near duplicate image detecting algorithm based on bag of visual word model*, vol 8(5), pp 557-565, Journal of Multimedia, 2013.
- [15] F Nian ,T Li , X Wu , Q Gao , F Li. *Efficient near-duplicate image detection with a local-based binary representation*, vol 75(5), pp 2435 - 2452, Multimedia Tools Applications, 2016.
- [16] Y Hua , H Jiang , D Feng. *FAST: Near real-time searchable data analytics for the cloud*, pp 754 -765, IEEE Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2014.
- [17] E. Rublee, V. Rabaud, K. Konolige and G. Bradski. *ORB: an efficient alternative to SIFT or SURF*, pp 2564 -2571 International Conference on Computer Vision, 2012.
- [18] T Miroslav , H Mark. *Fast corner detection*, pp 75 -87, Image and Vision computing, 1998.
- [19] J. G. Daugman. *Two-dimensional spectral analysis of cortical receptive field profiles*, vol 20(10), pp 847 -856, Vision Research, 1980.
- [20] Y Zhang, W Li, L Zhang, X Ning, L Sun, Y Lu. *Adaptive learning Gabor filter for finger-vein recognition*, vol 7, pp 159821-159830, IEEE Access, 2019.
- [21] J. G. Daugman. *Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression*, vol 36(7), pp 1169-1179, IEEE Transaction in acoustics speech & signal processing, 1988.
- [22] M. Porat, Y. Y. Zeevi. *The generalized Gabor scheme of image representation in biological and machine vision*, vol 10(4), pp 452-468, IEEE Transaction in Pattern Analysis and Machine Intelligence, 1988.
- [23] R. Glicksman. *Archiving: Fundamentals of Storage Technology*, Society for Imaging Informatics in Medicine.
- [24] D.R.VarmaM. *Managing DICOM Images: Tips and Tricks for the radiologist*, vol 22(1), pp 4-13, Indian Journal of Radiology and Imaging, 2012.
- [25] S.G.Shiny, T. Thomas, K. Chitharanjan *Cloud Based Medical Image Exchange Security Challenges*, vol 38, pp 3453-3461, Procedia Engineering, 2012.
- [26] M.Alhajeri, S.G.S.Shah. *Limitations in and Solutions for Improving the Functionality of Picture Archiving and Communication System: an Exploratory Study of PACS Professionals' Perspective*, vol 32(1), pp 54-67, Journal of Digital Imaging, 2018.
- [27] S. Ayyappan, C. Lakshmi. *Empirical Analysis of Robust Chaotic Maps for Image Encryption*, vol 9(11), pp 393-398, International Journal of Innovative Technology and Exploring Engineering, 2020.