Citation for published version:
Sumesh, TA, Namboodiri, V & Gupta, P 2019, Supervised Hashing for Retrieval of Multimodal Biometric Data. in C Arora & K Mitra (eds), *Computer Vision Applications - 3rd Workshop, WCVA 2018, held in Conjunction with ICVGIP 2018, Revised Selected Papers.* Communications in Computer and Information Science, vol. 1019 CCIS, Springer, Singapore, pp. 89-101, 3rd Workshop on Computer Vision Applications, WCVA 2018, held in conjunction with the 11th Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP 2018, Hyderabad, India, 18/12/18. https://doi.org/10.1007/978-981-15-1387-9_8

DOI:
10.1007/978-981-15-1387-9_8

Publication date:
2019

Document Version
Early version, also known as pre-print

Link to publication

Supervised Hashing for Retrieval of Multimodal Biometric data

No Author Given

No Institute Given

Abstract. Biometrics systems commonly utilize multi-biometric approaches where a person is verified or identified based on multiple biometric traits. However, requiring systems that are deployed usually require verification or identification from a large number of enrolled candidates. These are possible only if there are efficient methods that retrieve relevant candidates in a multi-biometric system. To solve this problem, we analyze the use of hashing techniques that are available for obtaining retrieval. We specifically based on our analysis recommend the use of supervised hashing techniques over deep learned features as a possible common technique to solve this problem. Our investigation includes a comparison of some of the supervised and unsupervised methods viz. Principal Component Analysis (PCA), Locality Sensitive Hashing (LSH), Locality-sensitive binary codes from shift-invariant kernels (SKLSH), Iterative quantization: A procrustean approach to learning binary codes (ITQ), Binary Reconstructive Embedding (BRE) and Minimum loss hashing (MLH) that represent the prevalent classes of such systems and we present our analysis for the following biometric data: face, iris and fingerprint for a number of standard datasets. The main technical contributions through this work are as follows: a) Proposing Siamese network based deep learned feature extraction method b) Analysis of common feature extraction techniques for multiple biometrics as to a reduced feature space representation c) Advocating the use of supervised hashing for obtaining a compact feature representation across different biometrics traits. d) Analysis of the performance of deep representations against shallow representations in a practical reduced feature representation framework. Through experimentation with multiple biometrics traits, feature representations, and hashing techniques, we can conclude that current deep learned features when retrieved using supervised hashing can be a standard pipeline adopted for most unimodal and multimodal biometric identification tasks.

Keywords: Biometric systems, Supervised hashing

1 Introduction

There has been tremendous growth in personal digital data stored across the Internet. With the proliferation of social media applications, this trend has increased. These data majorly comprising of images of persons has become a means to identify people. But the size of this data is enormous to the tune of billion in the case of Facebook as it has got more than 2 billion users. Similarly, Twitter, Instagram, and other social media applications have millions of images of the users. Several countries across the world also maintain a unique identification system of their citizens. These systems store various biometric features like face, iris and fingerprint images of the persons in addition to other credentials. When we think about using these images for identification purposes, indexing techniques using approaches like multidimensional trees comes into the picture. But indexing has always been a challenging task in the case of biometric databases due to various challenges like high dimensional feature representations, a varying number of dimensions for same trait and scalability. Further, with an extensive collection of data available over the internet, there is a need for faster indexing and search so that finding nearest neighbors can be done quickly.

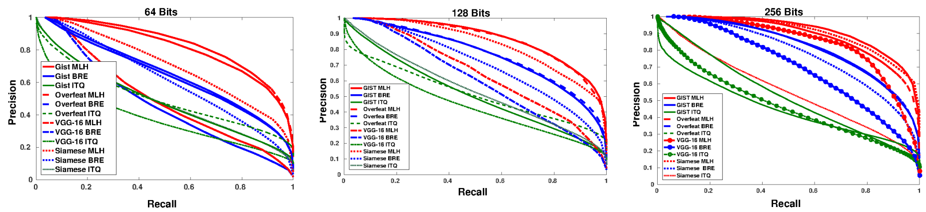

Fig. 1. Precision Recall curves for LFW Face database with Siamese, Gist, Overfeat and VGG-16

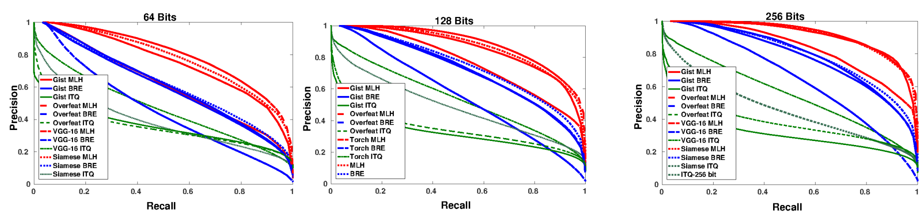

Fig. 2. Precision Recall curves for CASIA Fingerprint database with Siamese, Gist, Overfeat and VGG-16

Various biometrics traits usually need high dimensional feature representation, and they suffer from the curse of dimensionality. For instance, a face has a large number of feature points making it a feature rich biometric trait. For example, a face image of size 100×100 can have feature points up to 10,000. Due to the easy availability of non-intrusive surveillance systems, the face could be easily used to recognize people. However, it requires handling large databases of faces for identification.

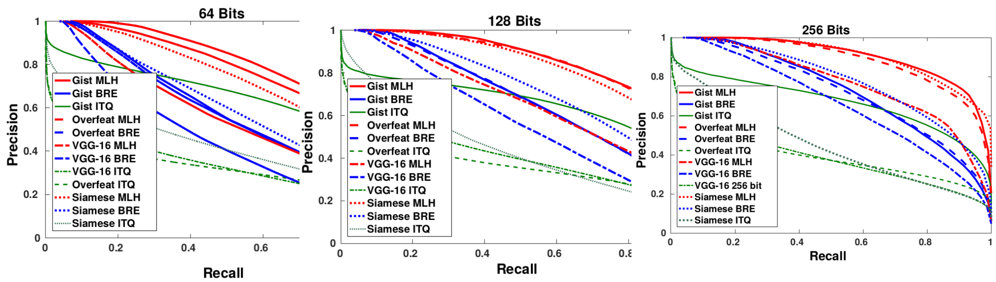

Fig. 3. Precision Recall curves for CASIA Iris database with Siamese, Gist, Overfeat and VGG-16

Feature representation of biometric data can affect the performance of the indexing mechanism. Previously some methods in the literature have used heuristics based feature representation. The indexing mechanism in such pipelines mostly use tree data structure like Kd-Tree. But these data structures are not very useful in handling the curse of dimensionality and storage requirements as these methods were dealing with feature representation in the real space itself. Thus these methods could not work well with high dimensional data in the order 1K features or more. Fortunately, there have been some attempts recently to use binary hashing techniques in the visual object recognition and scene recognition, as an effort to enhance the speed of the image retrieval and reduce the storage requirement. To the best of our knowledge, there has been no systematic analysis of this approach in the domain of biometric identification. In this paper, We, therefore, explore the possibility of including supervised binary hashing in the existing pipeline of the biometric identification system.

There have been a few instances where the use of hashing techniques for biometric data proposed in the past. Sergey Tulyakov et al. [12] proposed a hashing method for fingerprint data. In this method, minutiae points are represented as complex numbers and hash functions are constructed based on some complex function which is independent of the order of minutiae points. Yagiz Sutcu et al. [13] proposed a hash function based on one way transform function, designed as a sum of properly weighted and shifted Gaussian functions for biometrics. David CL Ngo et al. [14] proposed a method for dimensionality reduction using random thresholding projection to improve the accuracy of the face recognition. Christian Rathgeb and Andreas Uhl [15] proposed a hashing based on thresholding for Iris based recognition system. But most of these proposals are specific to some specific biometric data, and their main focus was on improving security in the verification pipeline and not the retrieval speed improvement or storage space optimization.

In this paper, we propose a feature extraction mechanism based on Siamese Network [17]. In our implementation, we use only three convolution layers and one fully connected layer. We observe that the deep learned features got from this model provides comparable performance with other pretrained models we

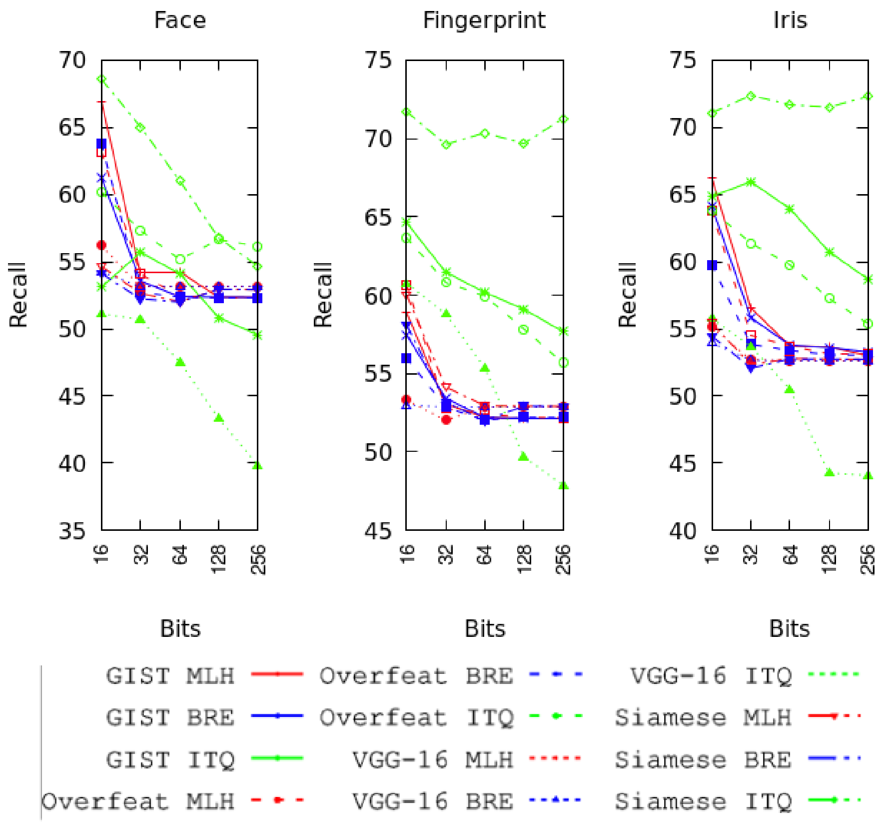

Fig. 4. Comparing hashing methods - MLH, BRE, and ITQ with respect to Bit vs Recall Performance

experimented. We also advocate the use of supervised hashing method in the existing pipeline of biometric identification system to reduce the dimensionality of biometric databases. We can make such a suggestion based on a thorough evaluation of various feature representations and hashing techniques for multiple biometric traits. These methods, in general use random projection to map feature vectors in real space to binary space such that similar items in real space concerning Euclidean distance mapped to objects of low hamming distance in binary space. In our evaluation, we consider both supervised and unsupervised techniques. We observe that supervised methods are better than unsupervised ones. We compare the performance of methods under these classes which generate a binary and non-binary representation of the data.

Table 1. Area under the curve values for LFW Face database for Siamese(4096D) vs Gist(512D) vs Overfeat(4096D) vs VGG-16(8192)

FACE - SIAMESE					
Bits	16	32	64	128	256
MLH	0.29	0.50	0.63	0.75	0.82
BRE	0.23	0.37	0.51	0.6	0.69
ITQ	0.39	0.44	0.49	0.54	0.56
SKLSH	0.24	0.45	0.61	0.73	0.83
LSH	0.17	0.25	0.31	0.40	0.51
PCA	0.21	0.22	0.19	0.18	0.20
FACE - GIST					
MLH	0.30	0.68	0.77	0.83	0.88
BRE	0.22	0.43	0.61	0.73	0.78
ITQ	0.34	0.40	0.49	0.51	0.53
SKLSH	0.10	0.23	0.27	0.61	0.59
LSH	0.24	0.32	0.36	0.40	0.49
PCA	0.21	0.19	0.14	0.13	0.14
FACE - OVERFEAT					
MLH	0.35	0.68	0.80	0.83	0.85
BRE	0.26	0.42	0.62	0.74	0.79
ITQ	0.42	0.43	0.50	0.52	0.45
SKLSH	0.29	0.33	0.49	0.56	0.74
LSH	0.34	0.37	0.43	0.43	0.44
PCA	0.22	0.21	0.19	0.19	0.15
FACE - VGG-16					
MLH	0.23	0.38	0.44	0.59	0.78
BRE	0.19	0.29	0.41	0.54	0.63
ITQ	0.28	0.17	0.37	0.44	0.47
SKLSH	0.15	0.16	0.19	0.32	0.39
LSH	0.15	0.16	0.26	0.33	0.40
PCA	0.24	0.21	0.20	0.20	0.18

2 HASHING METHODS

In this section, we provide a brief overview of various hashing methods in the literature. While we do not propose a new technique, the focus of this study is to evaluate whether it is possible to obtain a compact representation for multiple biometric data. We provide an overview of these techniques to gain sufficient insight into the various methods that form the crux of this paper. Our main observation after empirical analysis leads to suggesting the use of supervised binary hashing for representation of biometric data.

2.1 Locality Sensitive Hashing (LSH)

LSH ([3], [4]) is an unsupervised data independent hashing method, which reduces dimensionality of input data by mapping similar items to same buckets

Table 2. Area under the curve values for CASIA Fingerprint database for Siamese(4096D) vs Gist(512D) vs Overfeat(4096D) vs VGG-16(8192)

FINGERPRINT - SIAMESE					
Bits	16	32	64	128	256
MLH	0.28	0.56	0.74	0.83	0.88
BRE	0.23	0.42	0.58	0.70	0.76
ITQ	0.36	0.43	0.39	0.49	0.46
SKLSH	0.29	0.31	0.55	0.72	0.80
LSH	0.29	0.39	0.43	0.47	0.49
PCA	0.22	0.19	0.19	0.19	0.20
FINGERPRINT - GIST					
MLH	0.28	0.56	0.71	0.81	0.87
BRE	0.26	0.45	0.57	0.68	0.75
ITQ	0.29	0.33	0.36	0.32	0.31
SKLSH	0.24	0.27	0.34	0.53	0.65
LSH	0.18	0.26	0.32	0.32	0.41
PCA	0.25	0.21	0.21	0.18	0.16
FINGERPRINT - OVERFEAT					
MLH	0.33	0.64	0.76	0.85	0.89
BRE	0.25	0.42	0.57	0.68	0.74
ITQ	0.31	0.33	0.35	0.35	0.38
SKLSH	0.25	0.28	0.31	0.51	0.62
LSH	0.20	0.29	0.32	0.34	0.40
PCA	0.23	0.25	0.22	0.20	0.17
FINGERPRINT - -16					
MLH	0.26	0.44	0.58	0.72	0.79
BRE	0.20	0.29	0.42	0.52	0.60
ITQ	0.36	0.39	0.47	0.54	0.55
SKLSH	0.11	0.22	0.27	0.36	0.55
LSH	0.20	0.24	0.35	0.42	0.49
PCA	0.26	0.23	0.20	0.22	0.17

with high probability. The LSH function maps a point p to bucket $g_j(p)$. To process a query q, it searches all indices (hash tables) $g_1(q), ..., g_l(q)$. For the approximate $k - NN$, it outputs the k points p_i closest to q. The optimal value of k is chosen such that a point p closed by the distance to q falls into the same bucket as q and a point $p^{'}$ far away by the distance from q into the different bucket.

2.2 Locality-Sensitive Binary Codes from Shift-Invariant Kernels (SKLSH)

SKLSH [8] proposed by Raginsky and Lazebnik is also an unsupervised data independent hashing method. It uses random projection to obtain a binary encoding of data such that similar data points map to binary strings with low hamming distance. It assumes that data initially embedded in $\mathbb{R}^D$, and a shift-invariant kernel $K(,)$ is defined on that space. Then a randomized continuous

Table 3. Area under the curve values for CASIA Iris database for Siamese(4096D) vs Gist(512D) vs Overfeat(4096D) vs VGG-16(8192)

IRIS - SIAMESE					
Bits	16	32	64	128	256
MLH	0.33	0.56	0.66	0.79	0.86
BRE	0.32	0.44	0.58	0.66	0.74
ITQ	0.40	0.38	0.42	0.44	0.45
SKLSH	0.30	0.39	0.56	0.70	0.81
LSH	0.28	0.37	0.39	0.44	0.47
PCA	0.22	0.22	0.18	0.19	0.19
IRIS - GIST					
MLH	0.21	0.54	0.73	0.80	0.85
BRE	0.18	0.36	0.51	0.61	0.67
ITQ	0.59	0.60	0.65	0.65	0.67
SKLSH	0.39	0.40	0.59	0.80	0.89
LSH	0.48	0.49	0.56	0.65	0.68
PCA	0.40	0.33	0.28	0.26	0.22
IRIS - OVERFEAT					
MLH	0.24	0.52	0.70	0.80	0.83
BRE	0.18	0.39	0.52	0.61	0.67
ITQ	0.30	0.37	0.33	0.37	0.40
SKLSH	0.18	0.25	0.39	0.54	0.65
LSH	0.20	0.28	0.38	0.33	0.37
PCA	0.25	0.21	0.22	0.18	0.15
IRIS - VGG-16					
MLH	0.20	0.37	0.51	0.61	0.72
BRE	0.19	0.27	0.40	0.52	0.61
ITQ	0.31	0.32	0.33	0.40	0.38
SKLSH	0.16	0.21	0.33	0.35	0.51
LSH	0.17	0.19	0.28	0.26	0.36
PCA	0.25	0.22	0.19	0.19	0.20

mapping is done which guarantees to preserve kernel values with high probability. Then this mapping is binarized with the kernel values preserved. It provides a simple and data independent mapping with theoretical convergence guarantees.

2.3 Iterative Quantization (ITQ)

Iterative Quantization is a simple and efficient dimensionality reduction scheme, proposed by Yunchao Gong and Svetlana Lazebnik [5] to reduce the quantization error by mapping the high dimensional data to the vertices of a binary hypercube (zero-centered). It assumes that there is a set of n data points (zero-centered) $x_1, ..., x_n, x_i \in R^d$ that form the rows of the data matrix $X \in R^{n\times d}$. ITQ learns a binary code matrix $B \in \{-1, 1\}^{n\times c}$, where c is the code length. For each bit $k = 1, ..., c$, the binary encoding function is defined by $h_k(x) = sgn(xw_k)$, where w_k is a column vector of hyperplane coefficients and $sgn(v) = 1$ if $v \geq 0$ and 0 otherwise.

Table 4. MAP values for comparing effect of code length on various hashing methods

FACE												
	MLH				BRE				ITQ			
Feature	SIA	Gist	OF	Vgg	SIA	Gist	OF	Vgg	SIA	Gist	OF	Vgg
256	0.89	0.9	0.87	0.82	0.75	0.8	0.8	0.67	0.5	0.53	0.48	0.61
128	0.81	0.85	0.85	0.62	0.67	0.74	0.75	0.58	0.48	0.51	0.52	0.56
64	0.69	0.79	0.81	0.47	0.6	0.62	0.63	0.43	0.42	0.48	0.42	0.49
32	0.53	0.73	0.73	0.41	0.38	0.45	0.43	0.32	0.36	0.37	0.35	0.19
16	0.36	0.43	0.47	0.26	0.35	0.27	0.34	0.21	0.29	0.3	0.37	0.32
FINGERPRINT												
256	0.9	0.9	0.91	0.81	0.77	0.78	0.76	0.62	0.33	0.37	0.42	0.56
128	0.84	0.84	0.87	0.74	0.72	0.7	0.69	0.53	0.36	0.35	0.38	0.54
64	0.75	0.74	0.78	0.59	0.6	0.6	0.58	0.43	0.29	0.33	0.36	0.46
32	0.6	0.58	0.67	0.46	0.44	0.47	0.45	0.3	0.33	0.31	0.3	0.39
16	0.35	0.33	0.4	0.28	0.3	0.31	0.28	0.21	0.26	0.24	0.27	0.34
IRIS												
256	0.89	0.89	0.86	0.75	0.75	0.7	0.7	0.64	0.31	0.54	0.43	0.54
128	0.81	0.83	0.83	0.63	0.69	0.64	0.64	.54	0.31	0.51	0.39	.54
64	0.7	0.77	0.73	0.53	0.57	0.54	0.54	0.42	0.31	0.48	0.33	0.44
32	0.56	0.6	0.55	0.38	0.45	0.39	0.42	0.28	0.26	0.44	0.34	0.39
16	0.36	0.31	0.32	0.22	0.34	0.25	0.22	0.2	0.3	0.46	0.26	0.32

2.4 Binary Reconstructive Embedding (BRE)

Binary Reconstructive Embedding (BRE) is a supervised hashing method proposed by Brian Kulis and Trevor Darell [6]. The method uses the learning of hash functions that minimize reconstruction error between the original distances and the Hamming distances of the corresponding binary embeddings. A scalable coordinate-descent algorithm is used for the proposed hashing objective to learn hash functions in a variety of settings efficiently.

It assumes that the data is normalized to have a unit l_2 norm to help proper comparison of distances in the input space to hamming space. A b-dimensional binary embedding obtained by projecting the data using a set of b binary valued hash functions $h_1, ..., h_b$. Then low dimensional reconstruction is given by $\tilde{x_i} = [h_1(x_i); h_2(x_i); ...; h_b(x_i)]$. Then the squared error between the original distance and the distance reconstructed distance is minimized to get a good reconstruction.

2.5 Minimum Loss Hashing for Compact binary codes (MLH)

Mohammad Norouzi and David M. Blei proposed Minimum loss hashing [7] which is a supervised binary hashing technique that uses random projections to map high-dimensional input into binary codes. It assigns a 1, if the bit corresponding to the input is on one side of the hyperplane and 0, if it is on the other side.

Then a hinge-like loss function in SVM, which based on some threshold ρ bits in the Hamming space assign a cost to a pair of binary codes and a similarity label. If similar codes have a Hamming distance less than or equal to ρ bits, then it assigns a small cost otherwise it assigns a large cost. Finally, it learns a parameter matrix w which maps high dimensional inputs to binary codes by minimizing the empirical loss over training points.

3 EXPERIMENTATION

We performed our experiments on popular databases of the face, iris, and fingerprint which ensure variation in the database and across the databases relevant to the feature points. Some of the images in the face database are profile picture in the case of face images. Some of the iris images taken with spectacles on and some fingerprint images are rotated. Each database has undergone three traversals of training and query traversal. The face database we use is Labeled Faces in the Wild (LFW) which consists of 13,234 images. The iris database we use is CASIA-Iris-Thousand version 4.0 of 1000 subjects. The fingerprint database we use is CASIA Fingerprint image database Version 5.0 of 500 subjects. Both the databases consist of 20,000 images.

We implemented a Siamese neural network (SIA) which takes two images in parallel. We feed a pair of similar or a pair of dissimilar images during any iteration. A single fully connected layer outputs 4096 dimension feature vector which then passes through a Sigmoid() function. This output is compared to see if the images are similar or not. We also extract feature descriptors from each of the databases using Gist [11], Overfeat (OF) [1] CNN and torch CNN using VGG-16 [16] training model. We run three iterations of each of the methods on all of the databases.

We use the original implementation provided by Olivia and Torrabla [11] for extracting Gist features of 512 dimensions. We use the implementation provided by CILVR lab at New York University [1] to extract Overfeat features. We use torch CNN with VGG-16 training model which runs only on CPU. We take the output of the fc7 layer of both the CNNs, which gives 4096 and 8192 dimension feature vectors respectively. We divide each data set into 1000 training samples and 3000 testing samples. On each training set, we compute the Euclidean distance for each data point to find their 100 ground-truth neighbours. Then we compute precision and recall statistics during testing using the ground-truth neighbours and non-neighbours.

3.1 Analysis of Feature Vector Representations

We analyze the retrieval performance concerning feature representation of different modalities, using Precision-Recall for the MLH and BRE which are the supervised hashing techniques. We also compare the Precision-Recall performance of unsupervised method ITQ to establish the superior performance of supervised methods over the unsupervised methods. We evaluate the feature representations obtained from Siamese network, Gist, Overfeat, and VGG-16 for bit sizes ranging from 16 to 256 bits. The Precision-Recall graphs for this comparison are provided in Figures 1 to 3. We have omitted the curves of LSH, SKLSH, and PCA to avoid cluttering. Also we have not included the Precision-Recall curves for 16 bit and 32 bit representation to save the space. We observed that the performance is not good for this code lengths in general. We observe that the feature vector representation using Overfeat and Gist are performing better and their performances are comparable. Siamese representation also gives comparable performance for 128 bit and 256 bit code lengths. The VGG-16 representation provides a slightly inferior performance, which is more visible (from table 3(AUC) and table 4(MAP)) for the lower bit sizes up to 64 bits of all the modalities. We observe that consistent retrieval performance achieved across all the modalities, with Siamese, Gist or Overfeat representation.

3.2 Analysis of Hashing

We can infer from figure 4, and tables 1 to 3 most of the unsupervised methods are inferior to supervised methods, such as MLH and BRE. We also compare the unsupervised method ITQ as it was performing better in 16 bit and 32 bit case of Iris. If we ignore the lower bit cases (16 and 32) of Iris, in all other cases MLH was performing better. Among the unsupervised methods, SKLSH was performing better for the 128 bit and 256 bit case and ITQ otherwise. It is recommended to use the supervised method MLH with 128 bit or 256 bit to achieve a better retrieval performance across all modalities. Some of the unsupervised methods show better performance in the Bit versus recall curves. But their precision performance is poor compared to supervised methods as obvious from the tables 1 to 3, and 4. This means that the relevant items retrieved may be containing more false items.

3.3 Biometric wise analysis of methods

All biometric databases retrieved with better accuracy by supervised methods, especially the MLH, compared to unsupervised methods except for 16 and 32 bit cases of Iris database. It is evident from tables 1 to 3 that for bit sizes from 64 bit onward the supervised methods, especially MLH, works well regardless of the biometric modalities.

3.4 Computation factors

The training time for supervised techniques was taking around 12 to 14 hours on core i7 desktop with 16GB RAM. The training time was almost uniform regardless of the modalities or feature representation in the supervised setting. But on an i7 machine with 8GB RAM it took nearly two days to finish the training. The unsupervised techniques were taking a maximum of a couple of minutes for the entire process for all modalities. The feature vector generation with Gist took the least time, an hour, with VGG-16 it took nearly 6 hours, and with Overfeat takes more than 24 hours. Siamese network took around 2 hours for iris and face and 6 hours for the fingerprint case when run on GTX 1080 Ti.

4 Discussion

Experimentation results show that MLH supervised method works consistently well with all biometric databases for binary code size from 64 bits to 256 bits. So for any of the face, fingerprint and iris databases, MLH can be the best choice. In the case where (16 and 32 bit cases of Iris) unsupervised methods are superior, but the overall accuracy is less in those cases. So it is recommended to use MLH on the combination of the above three biometric databases sets, with a binary code size of 128 bits or 256 bits.

If we ignore computational limitations regarding feature vector generation time or training time, then MLH provides the best accuracy over Siamese, Gist and Overfeat feature representations for a bit size of 256 bits. Both methods were performing comparably well. We also observed that maximum accuracy obtained for face and iris databases with Gist and Siamese. In the case of the fingerprint database, Overfeat provided maximum accuracy.

As we have seen previously, supervised methods are performing consistently well across all data sets for 128 and 256 bit sizes regardless of the computational need and biometric modality. Then if we have to find a trade-off between accuracy and storage, then 128 bit could be the best choice. We recommend this because, with 128 bit, accuracy is closer to that of 256 bits, while it needs less storage size compared to 256 bit case.

From tables 1 to 3 and figures 1 to 3, we found that MLH over Overfeat feature representation of the fingerprint database performed most accurate for a bit size of 256.

So if we ignore the computational time for preparing the feature representation, then it would be better to choose Overfeat feature representation. We suggest this because Overfeat representation contains more details of the biometric images compared to Gist and its performance is consistent with all data sets, for bit sizes of 128 or 256 bits.

5 CONCLUSION

Our experimentation and analysis show that MLH supervised hashing method performs consistently better than unsupervised methods for all bit lengths except 16 bit case of Iris database across all feature representations. So it would be ideal to use a setting where supervised hashing employed for multimodal biometric data retrieval with feature representations being either Gist or Overfeat.

References

1. Sermanet, Pierre and Eigen, David and Zhang, Xiang and Mathieu, Michaël and Fergus, Rob and LeCun, Yann: Overfeat: Integrated recognition, local-

ization and detection using convolutional networks. booktitle="arXiv preprint arXiv:1312.6229(2013)
2. Davis, J. V., Kulis, B., Jain, P., Sra, S., Dhillon, I. S.:Information-theoretic metric learning. In: Proceedings of the 24th international conference on Machine learning,pp.209-216 ACM,(2007),
3. Indyk, Piotr, and Rajeev Motwani.:Approximate nearest neighbors: towards removing the curse of dimensionality,Proceedings of the thirtieth annual ACM symposium on Theory of computing,ACM(1998)
4. Charikar, Moses S.:Similarity estimation techniques from rounding algorithms.,Proceedings of the thirty-fourth annual ACM symposium on Theory of computing.,ACM(2002)
5. Gong, Yunchao, and Svetlana Lazebnik.:Iterative quantization: A procrustean approach to learning binary codes,IEEE Conference on Computer Vision and Pattern Recognition (CVPR),IEEE(2011)
6. Kulis, Brian, and Trevor Darrell.:Learning to hash with binary reconstructive embeddings.,Advances in neural information processing systems,2009
7. Norouzi, Mohammad, and David M. Blei.:Minimal loss hashing for compact binary codes,Proceedings of the 28th international conference on machine learning (ICML-2011)
8. Raginsky, Maxim, and Svetlana Lazebnik.:Locality-sensitive binary codes from shift-invariant kernels,Advances in neural information processing systems,2009
9. Wang, Jun, Sanjiv Kumar, and Shih-Fu Chang.:Semi-supervised hashing for scalable image retrieval,IEEE Conference on Computer Vision and Pattern Recognition (CVPR),IEEE(2010)
10. Rahimi, Ali, and Benjamin Recht.:Random features for large-scale kernel machines,Advances in neural information processing systems,2007
11. Oliva, Aude, and Antonio Torralba.:Modeling the shape of the scene: A holistic representation of the spatial envelope,International journal of computer vision 42.3,pp-145-175,2001
12. Tulyakov, Sergey, Faisal Farooq, and Venu Govindaraju.: Symmetric hash functions for fingerprint minutiae,International Conference on Pattern Recognition and Image Analysis,Springer Berlin Heidelberg(2005)
13. Sutcu, Yagiz, Husrev Taha Sencar, and Nasir Memon.:A secure biometric authentication scheme based on robust hashing,Proceedings of the 7th workshop on Multimedia and security,ACM(2005)
14. Ngo, David CL, Andrew BJ Teoh, and Alwyn Goh.:Biometric hash: high-confidence face recognition,IEEE transactions on circuits and systems for video technology 16.6,pp-771-775,IEEE(2006)
15. Rathgeb, Christian, and Andreas Uhl,:Iris-biometric hash generation for biometric database indexing,20th International Conference on Pattern Recognition (ICPR),IEEE(2010)
16. Simonyan, Karen, and Andrew Zisserman.:Very deep convolutional networks for large-scale image recognition,arXiv preprint arXiv:1409.1556, arXiv(2014)
17. Koch, Gregory, Richard Zemel, and Ruslan Salakhutdinov.:Siamese neural networks for one-shot image recognition., ICML Deep Learning Workshop. Vol. 2. 2015.