Iris Recognition based on Convolutional Neural Networks and Multi-Index Hashing

Maximilian Harrisson[1], **Leandro S. de Araújo**[1]

[1] Institute of Computing – Universidade Federal Fluminense – RJ, Brazil

mharrisson@id.uff.br, leandro@ic.uff.br

Abstract. *Efficient iris recognition at scale requires indexing structures that can search large galleries without a linear scan. This work evaluates 7 feature extractors, with 5 pre-trained CNNs (VGG16, ResNet50, ConvNeXt-Tiny, MobileNetV2, InceptionV3) and 2 classical descriptors (SIFT, ORB), combined with median binarization and Multi-Index Hashing on the CASIA-Iris-Lamp dataset, containing 16212 images and 411 subjects. The experiments compare extractors under a fixed 32-bit segment parameterization, analyze the trade-off between hit rate and penetration rate as a function of the number of tables, and evaluate thermometer binarization as an alternative to median. CNN extractors achieve rank-1 identification rates near 99%, while SIFT and ORB reach only 10.60% and 51.45% despite similar hit rates, showing that hit rate alone does not reflect recognition quality. VGG16 offers the best trade-off, combining a rank-1 rate of 99.45% with a penetration rate of 18.02%. Thermometer binarization improves rank-1 but produces a penetration rate of 100% across all tested configurations, eliminating the indexing advantage of Multi-Index Hashing.*

1. Introduction

Iris recognition is one of the most reliable biometric modalities, owing to the high entropy and long-term stability of iris texture patterns [Daugman 2004]. It is deployed in large-scale identification programs, such as Aadhaar in India and the Homeland Advanced Recognition Technology in the United States, which operate databases with millions of enrolled identities [Nguyen et al. 2024]. In these settings, identifying a person requires comparing a query iris against the entire gallery, and the cost of a linear scan grows prohibitively with database size. Efficient indexing structures are therefore essential to keep the identification practical at scale.

Multi-Index Hashing (MIH) [Norouzi et al. 2014] addresses this by indexing compact binary codes in multiple hash tables, achieving sublinear search complexity in Hamming space. Its effectiveness depends on the quality of the binary codes fed into the index, which in turn depends on two pipeline stages defined as feature extraction and binarization. Previous work, such as IHashNet [Singh et al. 2020], demonstrated the viability of combining MIH with neural feature extraction for iris retrieval, but using a dedicated network trained specifically for iris images. It remains unclear how well general-purpose pre-trained CNN architectures perform as feature extractors in this pipeline and how MIH parameterization should be adjusted for codes of different lengths.

To address these gaps, this work evaluates 5 pre-trained CNN architectures (VGG16, ResNet50, ConvNeXt-Tiny, MobileNetV2, and InceptionV3) and 2 classical

descriptors (SIFT and ORB) as feature extractors for MIH-based iris identification on the CASIA-Iris-Lamp dataset. The contributions involve a systematic comparison of pre-trained extractors under a fixed-segment MIH parameterization that equalizes search difficulty across different code lengths, an empirical analysis of the segment size effect on the hit rate and penetration rate trade-off, and an evaluation of thermometer binarization as an alternative to median binarization under the same indexing pattern.

2. Related Works

The use of deep learning for iris recognition has grown considerably over the past decade. [Nguyen et al. 2024] survey more than 200 publications and organize the field around two main tasks defined as segmentation and recognition. Their analysis highlights the increasing adoption of CNN, including architectures such as VGG16, for feature extraction in biometric systems, replacing classical descriptors. An example is DeepIris-Net2 [Gangwar et al. 2019], which learns iris representations directly from raw images, bypassing classical segmentation and normalization. The network produces compact binary codes called Deep-IrisCodes, showing that binarization is also viable over learned CNN features. Efficient retrieval of similar binary codes in large databases has been extensively studied. Multi-Index Hashing introduced by [Norouzi et al. 2014], achieving sub-linear search complexity through hash table partitioning, and remains a primary reference for Hamming-space indexing. IHashNet [Singh et al. 2020] is the closest work related to this study, combining CNN-based feature extraction with MIH indexing in a single system for iris recognition. The network extracts patch-level ordinal features (POFNet), combined with 1×1 trainable convolutions, producing 512-dimensional vectors that are converted into 512-bit binary codes called iris bar codes. This work differs from IHash-Net in two aspects. First, it evaluates multiple pre-trained CNN architectures and classical descriptors as feature extractors, rather than training a dedicated iris network. Second, it explicitly investigates how MIH affects the trade-off between hit rate and penetration rate across extractors with different code lengths.

3. Methodology

Dataset. This work uses the CASIA-Iris-Lamp subset of CASIA-IrisV4[1], containing 16212 grayscale images with 640×480 pixels from 411 individuals, captured under varying illumination conditions. Images are organized by subject and eye side, then split into a gallery of 12966 images for index construction and 3246 query images, with an 80/20 split. The same subject appears in both sets, but no single image serves as both a gallery entry and a query.

Multi-Index Hashing. Multi-Index Hashing (MIH) [Norouzi et al. 2014] indexes binary codes by splitting each B-bit code into m disjoint segments of B/m bits and storing each segment in an independent hash table. A brute-force search over N codes costs $O(NB)$, and MIH reduces this by exploiting the pigeonhole principle: if two codes differ in at most r bit positions, then at least one of the m segments must agree within a radius r' defined as follows:

$$r' = \left\lfloor \frac{r}{m} \right\rfloor \tag{1}$$

[1]Available at: `https://hycasia.github.io/dataset/casia-irisv4/`. Accessed: June 10, 2026.

A query therefore enumerates only the segments within the radius r' in each table, collects the union of candidate identifiers, and reranks them by full Hamming distance to discard false positives. Search complexity becomes sublinear when codes are uniformly distributed in Hamming space.

In this work, each B-bit code is split into $m = B/32$ segments of 32 bits, each feeding an independent hash table. This fixed-segment strategy equalizes per-table search difficulty across multiple extractors with different code lengths, so comparisons are not affected by segment size. At query time, all segments within the Hamming radius $r' = 2$ of each table segment are enumerated, and the union of matched identifiers forms the candidate set, which is then reranked by full Hamming distance to the query.

Feature Extractors. In this work, the 5 CNN models used as feature extractors are VGG16 [Simonyan and Zisserman 2014], ResNet50 [He et al. 2016], ConvNeXt-Tiny [Liu et al. 2022], MobileNetV2 [Sandler et al. 2018], and InceptionV3 [Szegedy et al. 2016]. All models are pre-trained on ImageNet with frozen weights. The classification head is discarded, and global average pooling is applied to the last feature map, yielding a fixed-length vector. Input images are resized to 224×224 pixels for most models and 299×299 for InceptionV3, then replicated across the three color channels and normalized with ImageNet mean and standard deviation. Feature dimensions are 512 for VGG16, 768 for ConvNeXt-Tiny, 1280 for MobileNetV2, and 2048 for both ResNet50 and InceptionV3. As classical baselines, SIFT [Lowe 2004] and ORB [Rublee et al. 2011] are evaluated. Both produce a variable number of local descriptors per image, one per keypoint, which are aggregated into a single global vector by averaging. SIFT yields a 128-dimensional real-valued vector, and ORB descriptors are already binary, so the per-bit mean is thresholded at 0.5, producing a 256-bit code.

Input Binarization. Real-valued vectors from CNNs and SIFT are binarized using each image's own feature vector median. A dimension is set to 1 if its value meets or exceeds the median and 0 otherwise, yielding a code of the same length as the original vector. ORB codes are indexed directly without further binarization.

Metrics. To evaluate the proposed pipeline, we adopt three metrics. The hit rate (HR) is the fraction of queries for which the correct result appears in the MIH candidate set, measuring index recall. The penetration rate (PR) is the average fraction of the gallery returned as candidates per query, so smaller values indicate greater search space reduction. The rank-k identification rate is the fraction of queries for which the match ranks among the k nearest candidates by Hamming distance [Arora et al. 2022], and unlike HR, it evaluates the quality within the retrieved set.

4. Experiments and Results

Experimental Setup. The experiments fix the per-segment Hamming radius at $r' = 2$. By default real-valued vectors are binarized by the median, and ORB uses its native binary code. The implementation uses Python 3.8+ with PyTorch and torchvision for CNN feature extraction and OpenCV for SIFT and ORB. All experiments were executed on the CPU. The four reported experiments are extractor comparison across all 7 extractors, impact of the number of MIH tables on the HR/PR trade-off, thermometer binarization as an alternative to median, and feature separability.

Recognition Performance Results. Table 1 reports HR, PR, rank-1 and rank-5 for all 7 extractors evaluated on subject identification. All CNN extractors achieve an HR of 100%, while SIFT and ORB reach 99.78% and 99.54%, respectively. However, HR alone is misleading in this regime: each query has approximately 16 genuine matches in the gallery, and the index needs to recover only one of them to register a hit. The rank-k metric shows the recognition gap between the extraction methods. For example, CNN extractors achieve rank-1 rates between 99.17% and 99.45%, while ORB reaches only 51.45% and SIFT only 10.60%. This pattern indicates that both SIFT and ORB recover the correct result within a large candidate pool, but consistently fail to rank it first. For CNNs, rank-1 differences are considerably small, being under 0.3%, so the main reference becomes the PR. VGG16 achieves the lowest PR with only 18.02%, while the highest PR is shown by ResNet50 with 83.96% of gallery access per query. This result is partly explained because ResNet50's 2048-bit code requires 64 tables, which increases the candidate union across lookups. InceptionV3 uses the same code length and number of tables as ResNet50 but achieves a more selective PR of 52.81%, suggesting that feature distribution also influences index selectivity beyond code length alone.

Extractor	Tables	HR (%)	PR (%)	rank-1 (%)	rank-5 (%)
SIFT	4	99.78	64.68	10.60	27.30
ORB	8	99.54	50.41	51.45	72.61
VGG16	16	100.00	18.02	99.45	99.78
ConvNeXt	24	100.00	50.06	99.23	99.57
MobileNetV2	40	100.00	43.30	99.26	99.60
InceptionV3	64	100.00	52.81	99.17	99.63
ResNet50	64	100.00	83.96	99.23	99.66

Table 1. Recognition Performance of extractors on subject identification with median binarization, 32-bit segments, and $r' = 2$.

Impact of Amount of Hash Tables. Table 2 varies $m \in \{4, 8, 16, 32\}$ for VGG16, fixing all other parameters to isolate the effect of MIH parameterization on the HR/PR trade-off. With $m = 4$, each segment spans 128 bits and the per-table radius $r' = 2$ is overly strict and HR falls to 1.66%, meaning the index misses the right match in almost all queries. At $m = 8$, with 64-bit segments, HR rises to 76.96%, but the index still discards 23% of the correct answers. At $m = 16$, with 32-bit segments, HR reaches 100% and PR of 18.02%. When increasing to $m = 32$, with 16-bit segments, HR remains at 100%, but the PR raises to 99.80%, making the search equivalent to a linear scan.

m	Segment (bits)	HR (%)	PR (%)	rank-1 (%)	rank-5 (%)
4	128	1.66	0.00	1.66	1.66
8	64	76.96	0.03	76.86	76.96
16	32	100.00	18.02	99.38	99.75
32	16	100.00	99.80	99.51	99.82

Table 2. Impact of the number of tables on VGG16 on subject identification with $r' = 2$.

Thermometer Binarization Results. As an extension of the parameterization analysis, the thermometer binarization is evaluated on VGG16 under the same fixed-segment pattern, with $m = B/32$ and $r' = 2$. For thermometer encoding with L levels, each dimension produces L bits, yielding a code of $512L$ bits and $m = 16L$ tables. Table 3 reports results for $L \in \{8, 10, 12, 14, 16, 18, 20\}$. HR reaches 100% across all levels, and rank-1 improves monotonically from 99.01% at $L = 8$ to a peak of 99.45% at $L = 18$, indicating that more levels preserve more ordinal information from the original feature vector. However, PR stands at 100% for every L size. The union of candidates across 128 to 320 tables covers the entire gallery per query, so it's also equivalent to a linear scan. This extends the trend observed in Table 2, where $m = 32$ tables already pushed PR to 99.80% for a 512-bit median code. Thermometer codes, by requiring far more tables to maintain 32-bit segments, push that trend to its limit. Median binarization therefore offers a better trade-off for this dataset, with a comparable rank-1 accuracy of 99.45% at $m = 16$ and a PR of 18%.

L	Tables	HR (%)	PR (%)	rank-1 (%)	rank-5 (%)
8	128	100.00	100.00	99.01	99.35
10	160	100.00	100.00	99.20	99.51
12	192	100.00	100.00	99.23	99.54
14	224	100.00	100.00	99.32	99.66
16	256	100.00	100.00	99.35	99.69
18	288	100.00	100.00	99.45	99.85
20	320	100.00	100.00	99.38	99.85

Table 3. Thermometer binarization on VGG16 and subject identification, with 32-bit segments and $r' = 2$.

5. Conclusion

This work evaluated 7 feature extractors, with 5 pre-trained CNNs and 2 classical descriptors, combined with median binarization and MIH for iris identification on the CASIA-Iris-Lamp dataset. Experiments compared extractors under a fixed-segment parameterization of $m = B/32$ and $r' = 2$, investigated the HR/PR trade-off as a function of the number of MIH tables, and examined thermometer binarization as an alternative to the median. CNN extractors achieved rank-1 identification rates near 99%, while SIFT and ORB reached only 10.60% and 51.45%, respectively, despite similar HR. This confirms that HR alone is not a reliable indicator of recognition quality under these conditions, and that mean-aggregated local descriptors are insufficient for iris identification using MIH. Among the CNNs, VGG16 stood out for combining the highest rank-1 rate of 99.45% with the lowest PR of 18.02%, making it the most efficient extractor for MIH-based retrieval in this setting. Varying the number of tables showed that the segment size determines the HR/PR balance. Segments larger than 32 bits cause the index to miss the correct results, while smaller segments recover the entire gallery. Thermometer binarization improved rank-1 consistently with the number of levels, but produced a PR of 100% across all configurations, as the large number of tables required by longer codes saturates the candidate union. Median binarization achieved the same peak rank-1 with a lower penetration rate, making it the more suitable choice for MIH indexing in this context.

Acknowledgements

This research was funded by CNPq (grant 307770/2025-7).

References

Arora, G., Vichare, S., and Tiwari, K. (2022). Irisindexnet: Indexing on iris databases for faster identification. In *Proceedings of the 5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD)*, pages 10–18.

Daugman, J. (2004). How iris recognition works. *IEEE Trans. Cir. and Sys. for Video Technol.*, 14(1):21–30.

Gangwar, A., Joshi, A., Joshi, P., and Raghavendra, R. (2019). Deepirisnet2: Learning deep-iriscodes from scratch for segmentation-robust visible wavelength and near infrared iris recognition.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. (2022). A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110.

Nguyen, K., Proença, H., and Alonso-Fernandez, F. (2024). Deep learning for iris recognition: A survey. *ACM Comput. Surv.*, 56(9).

Norouzi, M., Punjani, A., and Fleet, D. J. (2014). Fast exact search in hamming space with multi-index hashing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(6):1107–1119.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, A., Gaurav, P., Vashist, C., Nigam, A., and Yadav, R. P. (2020). Ihashnet: Iris hashing network based on efficient multi-index hashing. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.