Iris Recognition Using Low-Level CNN Layers Without Training and Single Matching

JORGE E. ZAMBRANO\(^1\), (Graduate Student Member, IEEE),
DANIEL P. BENALCAZAR\(^1\), (Member, IEEE),
CLAUDIO A. PEREZ\(^1\), (Senior Member, IEEE),
AND KEVIN W. BOWYER\(^2\), (Life Fellow, IEEE)
\(^1\)Department of Electrical Engineering and Advanced Mining Technology Center, Universidad de Chile, Santiago 8370451, Chile
\(^2\)Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA

Corresponding author: Claudio A. Perez (clperez@ing.uchile.cl)

This work was supported in part by the Agencia Nacional de Investigacion y Desarrollo (ANID) under Grant FONDECYT 1191610, Center AFBI0004, Center ANID/BASAL FB210024, Becas/Doctorado Nacional under Grant 21191614; and in part by the Department of Electrical Engineering and Advanced Mining Technology Center, Universidad de Chile.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Bioethics Committee, Facultad de Ciencias Fisicas y Matematicas, Universidad de Chile under Resolution No. 011.

**ABSTRACT** Iris is one of the most accurate biometrics. This has led to the successful development of large-scale applications. However, with population growth, and new international applications, datasets are constantly increasing in size, requiring more robust and faster methods. Many descriptors and feature extractors have been developed to extract features that represent the iris biometric pattern. Most of them have been designed by human experts and require a bit-shifting process to increase their robustness to eye rotations, at the expense of increased matching time. We propose a fast iris recognition method that requires a single matching operation and is based on pre-trained image classification models as feature extractors. Our approach uses the filters of the first layers from Convolutional Neural Networks as feature extractors and does not require fine-tuning for new datasets. Since our selected features extracted from convolutional layers encode the iris surface, they have the advantage of not being restricted to specific spatial positions. Thus, it is not necessary to perform a bit-shifting process in the matching stage, eliminating a significant number of computations. Additionally, to mitigate the effect produced by the mask border in rubber-sheet images, we propose filtering the feature map tensors by masking their channels and selecting the most relevant features. Our method was assessed on the publicly available datasets CASIA Iris Lamp and CASIA Iris Thousand, and showed significant improvement both in accuracy and in matching time.

**INDEX TERMS** Biometrics, bit-shifting, deep-learning features, feature extraction, iris recognition, pupil dilation.

I. INTRODUCTION Iris Recognition (IR) has become one of the most accurate approaches for biometric identification. The iris tissue forms complex patterns that are stable in time which makes it one of the most successful biometric methods [1]–[4]. Furthermore, the high level of accuracy (ACC) that can be obtained has led to the successful development of large-scale applications; for example, India’s Unique ID program [5], and the United Arab Emirates’ border-crossing surveillance [6]. However, with population growth and new international applications, such as those used for identification at border controls [7], [8], datasets are constantly increasing in size, requiring more robust and faster methods.

Since the pioneering work by John Daugman, which uses Gabor phase-quadrant features as descriptors [9], [10], iris recognition has progressed not only in accuracy, but also in the number of available datasets for research and evaluation [11], [12]. Many descriptors and feature extractors have been developed to extract the best features to represent...
this biometric pattern. Most of them have been designed by human experts and are based on experimental results [13]–[15]. Additionally, these classical “iris code” methods require a bit-shifting process to increase their robustness to eye rotations, at the expense of matching time [16], [17]. For example, Czajka et al. [18] and Fang [19] employed 33 matching operations per comparison (16 at the right, 16 at the left and at the center one) to cover rotations between ±11.25°.

With a deep learning approach, filters are no longer created by human experts, but, rather, an optimization process is performed to find the best coefficients, using a training process [20]–[22]. In addition, it is common to train a classifier, such as a Support Vector Machine (SVM), Multi Perceptron Layer (MPL), or Random Forest (RF), with a training stage [14], [23], [24]. The training stage could be a limitation since, for IR, there is not a standard dataset with enough images to adjust millions of parameters [25].

One approach to solve these limitations has been to extract iris features using publicly-available models, trained for natural image classification [25]–[27], and face recognition [28]. Most of these methods use the rubber sheet model as the input, and apply a mask to eliminate non-iris features, such as eyelids, eyelashes, and reflection artifacts [4], [16], [29], [30]. This approach has been very successful, however, there are still some limitations, including fine-tuning for each new iris dataset. Additionally, the effect of the mask, when the extracted features are not fixed at specific spatial locations, has not been well studied. The masking step may introduce errors since extracted features might be contaminated with the shape of the mask.

In this article, we propose a fast IR method that requires a single matching operation and is based on pre-trained image classification models as feature extractors [25]–[28]. Our approach uses the filters of the early layers from Convolutional Neural Networks (CNNs) as feature extractors, and our method does not require fine-tuning for new datasets. Since features extracted from convolutional layers encode the iris surface, they have the advantage of not being restricted to specific spatial positions. Thus, it is not necessary to perform a bit-shifting process in the matching stage, eliminating a significant number of computations. Additionally, our method aims to mitigate the masking effect produced by the mask border in rubber sheet images. To reduce this effect, we propose filtering the feature map tensors by masking their channels and selecting the most relevant features.

The main contributions of this paper are the following: (1) Developing an IR method based on selecting a layer from a CNN for feature extraction that does not require a training process; (2) Developing a method that does not need bit-shifting, thus reducing the matching time; (3) Proposing a novel method to reduce the impact of the mask on the extracted features; (4) Evaluating the performance of our method on publicly-available iris datasets such as CASIA Iris Lamp, and CASIA Iris Thousand; and (5) Improving accuracy on datasets that contain subjects with significant dilation changes.

II. RELATED METHODS

Daugman proposed the use of a Gabor phase-quadrant descriptor to extract features from iris images, obtaining high matching efficiency, and popularizing the iris as a reliable biometric [9], [10], [16]. More descriptors have been proposed for feature extraction, leveraging the advantages of the complex texture of the iris. Approaches based on Haar wavelets [31], wavelet packets [32], spatial filter banks [33], directional filter banks [34], Discrete Cosine and Fourier Transform [35], [36], Local Binary Patterns [37], and Binarized Statistical Image Features (BSIF) [18], have been investigated, achieving both low false matching and high recognition rates [18]. Not only has the texture provided by the 2D image been used, but also the 3D information corresponding to the relief in the iris tissue has been explored recently with excellent performance [29], [38].

With the arrival of Deep Learning techniques, new IR approaches have been developed. Deep Iris was presented by Liu et al. [39] in 2016 as the first attempt to solve the problem using CNNs. The model consists of a CNN, and a pairwise filter bank for iris verification. Gangwar and Joshi [40] proposed two deeper architectures, called DeepIrisNet A, and B. Both attained superior performance on the ND-IRIS-0405, and ND-CrossSensor-Iris-2013 datasets [41], [42]. Zhao and Kumar [43] presented a network called UniNet, based on fully convolutional networks. They introduced a loss function related to a variation of the Triplet Loss [44], to focus on the bit-shifting, and non-iris masking operations in the matching stage [43]. Wang and Kumar [45] proposed a model based on dilated convolutional kernels and residual network learning to obtain the more representative features from iris images.

Employing off-the-shelf weights, as well as fine-tuning techniques, has been explored to avoid the problem of requiring large iris datasets to adjust millions of parameters in complex architectures [25]. Mineae et al. [26] investigated the convolutional layers of a pre-trained VGG for iris feature extraction, and they evaluated different numbers of components in PCA carried out by an SVM, to achieve greater accuracy. Nguyen et al. [25] went deeper, employing five pre-trained image classification models for feature extraction, using pre-selected layers, and training a multiclassification SVM method. Boyd et al. [28] improved Nguyen’s approach by adding a fine-tuning stage and using different off-the-shelf weights. They relied on a One versus Rest SVM as a classifier before applying PCA in the features extracted to reduce the dimensionality. Zhao et al. introduced a method based on the capsule network architecture [27]. They built several convolution structures with various depths according to different outputs of pre-trained classic networks to dock with the capsule structure. The results in three datasets exceeded the baseline [27].
It is worth noting that all the approaches cited above depend on a training stage, either in the network itself, the classifier, the dimensionality reduction, or in the fine-tuning stage. Furthermore, the effect of the masking process on the selected features has not been considered in previous models.

III. METHODOLOGY
Our method consists of six steps. The first two include: iris segmentation and normalization, followed by image preprocessing that uses masking and image enhancement. The next four steps are performed in a loop assessing the IR performance for features extracted from all the convolutional layers of a CNN. We include the most important CNN models designed for image classification as feature extractor engines, such as, DenseNet [46], Inception [47], Inception ResNet [48], NASNetMobile [49], ResNet [50], and Xception [51]. However, our method could be applied to any other CNN. IR tests are performed using the features extracted from one convolutional layer at a time on a standard iris dataset. Then, for each model, we compare the IR results using all the convolutional layers and select the layer that provides the highest accuracy as the best set of feature extractors. Assessing each convolutional layer is done using a random subset of the CASIA Iris Lamp dataset, as is explained in section H. Finally, we tested our IR method on various datasets with the best convolutional layer selected to extract iris features. The experiments and the reported results were executed on an Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz computer, using Python 3.8. The steps are detailed as follows:

A. IRIS SEGMENTATION AND NORMALIZATION
We used the implementation provided by Wang et.al. [52] to obtain the iris mask, and the iris and pupil contours. Then, circles were fitted to the contours to obtain the center and the radius, which were used to normalize the irises and their masks using Daugman’s Rubber Sheet model.

B. IMAGE PREPROCESSING
Before feature extraction, iris rubber sheets were converted from grayscale to RGB images to make them compatible with the CNN feature extraction process. This was performed by tripling the grayscale single channel to obtain an image with three channels. Images are resized to $224 \times 224$ pixels to become the input of each model in the Keras framework [53]. Next, we applied contrast-enhancement to the rubber sheets using the CLAHE method [54]. In this method, the images are converted from the RGB into the HSV domain, where the third channel (Value) is equalized with the parameters’ Clip Limit at 10.0, and the Tile Grid Size at (8, 8). Then, images are returned to the RGB space to be masked. Fig. 1 shows an example of this process. As shown, the texture in (c) is enhanced compared to that in (a). Finally, the output of the image preprocessing step is used as the input to the CNN models in the loop to find the best features.

FIGURE 1. (a) Iris Rubber Sheet. (b) Mask. (c) Enhanced and masked Iris Rubber Sheet after the image preprocessing step.

C. FEATURE EXTRACTION
After image preprocessing, feature extraction is performed using some of the most reliable models for image classification: DenseNet [46], Inception [47], Inception ResNet [48], NASNetMobile [49], ResNet [50], and Xception [51]. The weights for the CNN models were obtained from the ImageNet dataset [55]. These extracted features are considered to be feature maps that will become feature vectors after reducing the mask impact on the extracted features in the next step.

D. REDUCING THE MASK EFFECT
It is important to mask the iris rubber sheets to avoid including irrelevant information, such as eyelids, eyelashes, light reflections, or other artifacts [15], [56], [57]. Nevertheless, when the iris image is masked, additional information is placed in the image, e.g., the shape of the mask. The contour of the mask may be a strong attribute that is included in the feature maps [58]–[60]. For example, a bias may be introduced since irises with similar masks could be considered to be similar, although belonging to different subjects. Conversely, some datasets contain images that were captured under almost the same conditions, thus, generating similar masks for the same subject. In this case, not only are the features matched, but also the mask shape is affecting the results.

To reduce this bias, we propose including just the elements in the feature map tensors that belong to iris features outside of the mask. The first step consists of obtaining the dimensions of the final feature map used $[w, h, ch]$ (width, height, channels). It is important to know just the width and the height since all the channels will be filtered in the same way. Next, a mesh grid is created over the original mask with the same feature map dimensions, and all the cells that belong to a portion of the mask are disabled, as is shown in Fig. 2. This process creates a new mask with dimensions of $[w, h]$ which is stored with the feature map tensor. The new masks will be used in the next step for filtering the corresponding feature maps. This new mask will not place additional information in the image, but it will eliminate elements of the feature map tensor that contain information about the mask placed on the iris rubber sheet.

E. ENCODING AND MATCHING FEATURE VECTORS
To compare two irises, the feature maps and their masks are loaded, and a unique mask is created by the AND logical
FIGURE 2. Process of creating a new mask for feature map filtering. First, a mesh grid is placed over the original mask. Just the cells without any part of the mask are used to create the new mask.

Feature Vectors are normalized to obtain a mean of zero and a standard deviation of one. Then, we select the most important features contained in the vector using a $\lambda$ value. Fig. 3 depicts the selection process, with just the values inside the interval $[-\lambda; +\lambda]$ considered, while the rest of them are ignored. It is done by creating a mask designed for the vectors using the expressions:

$$M_{V_1} = |F_{V_1}| \leq \lambda,$$

(1)

$$M_{V_2} = |F_{V_2}| \leq \lambda,$$

(2)

$$M_V = M_{V_1} \& M_{V_2},$$

(3)

where $M_{V_1}$ and $M_{V_2}$ represent the available positions in the Feature Vectors $F_{V_1}$ and $F_{V_2}$ according to the $\lambda$ value. $M_V$ corresponds to the unified mask, which was obtained with the logical AND operator. The last step of the selection process is to obtain the final values using the unified mask in both feature vectors.

The feature selection process is essential because it reduces the dimensionality of the feature vectors, avoiding outlier values and reducing the computational cost [61]. Once the best convolutional layers have been selected in each model, the IR performance is measured by evaluating different values of $\lambda$. Here, the best results were obtained with values close to 1 in most of the models, so we decided to use $\lambda = 1$ from then on. We used the same subset of the CASIA Iris Lamp dataset to select the best $\lambda$ value as that used for layer selection. Fig. 4 shows an example of the Accuracy obtained in an experiment selecting features from Xception with different values of $\lambda$.

To encode the new feature vectors, we created binarized vectors using a threshold of zero; thus, both vectors could be compared using the Hamming Distance. The length of the encoded features depends on the backbone used, the selected layer, and then, on the new mask and the feature selection process.

FIGURE 3. Feature Vector Normalized Histogram ($\mu = 0; \text{std} = 1$). Just the values inside the red box are considered. Then, features are binarized using a threshold (zero).

FIGURE 4. IR accuracy obtained by evaluating different values of $\lambda$ from 0.5 to 4.0 with features extracted from Xception.

F. MEASURING PERFORMANCE

In this paper, we report using the False Reject Rate (FRR) that yielded a False Accept Rate (FAR) equal to 0.1% and the Accuracy (FAR = FRR) [25], [27]. Both metrics are used for selecting the best convolutional layer for various CNNs in the loop process. For cases where the maximum ACC and minimum FRR does not coincide, the best layer was selected by finding the greatest difference between these two metrics.

G. CNN ARCHITECTURES

We investigated six of the most important models for image classification that were trained for the Large-Scale Visual Recognition Challenge (ILSVRC) [55]. The models were implemented on Keras 2.3.1 using the pre-trained ImageNet weights [62]. Table 1 lists the six models and the layer types explored in each CNN. Convolutional layers were used in Inception V3, Inception ResNet V2, and Xception. Concatenation and Residual Layers are used for DenseNet 201 and ResNet 50, respectively. It is worth noting that Concatenation Layers correspond to Dense Blocks, which are formed by Convolutional Layers [46]. In the same way, each Residual Layer is formed by Convolutional, Activation
TABLE 1. Models used for feature extraction, and the type of layer explored in each CNN. The third column shows the number of layers explored. Concatenation and Residual layers include the convolutional layers explored [46], [50]. For NASNetMobile we just use Normal Cells because Reduction Cells return feature maps reduced by a factor of two, thus reducing information [49].

| CNN               | Kind of Layer | Count |
|-------------------|---------------|-------|
| DenseNet 201      | Concatenation | 98    |
| Inception V3      | Convolutional 2D | 94    |
| Inception ResNet V2 | Convolutional 2D | 203   |
| NASNetMobile      | Separable Normal | 120   |
| ResNet 50         | Residuals     | 52    |
| Xception          | Convolutional | 111   |

Function ReLU, and Batch Normalization Layers [50]; here just the Convolutional Layers are considered. Finally, NASNetMobile is composed of Normal and Reduction Cells [49]. Reduction Cells were not used because they return a feature map whose height and width are reduced by a factor of two, while Normal Cells return a feature map with the same dimensions as the previous one [49].

H. DATABASES
For the CNN layer selection, we evaluated 678 sets of features, each one extracted from one of the various layers shown in Table 1. For this evaluation, we used a subset of CASIA Iris Lamp [63], with 30% of the classes on which it was not used in the test set, to compare our results with those published previously. This subset with 30% of the classes to select the best layers was subject disjoint. The 678 sets of features from Table 1 were evaluated using the same subset from the CASIA Iris Lamp dataset for a fair comparison, thus allowing the selection of the best layer for each model.

Once the best layer was selected for each model, we used the partitions of CASIA Iris Thousand, and CASIA Iris Lamp as the previously published results for the final assessment of our method [25], [27]. These partitions were formed to compare our results to those previously published using the same conditions. We selected 30% of the classes for CASIA Iris Thousand to obtain the same number of images as in Nguyen’s work [25], in the same way. For the CASIA Iris Lamp dataset, 1,500 images were selected as the test partition as in Zhao’s work [27].

Additionally, we selected the partitions for both CASIA Iris Thousand, and Iris Lamp, to obtain two datasets that contain iris images with significant dilation changes for the same subject. We measured and sorted the difference between the smallest and the largest dilation level for each class. Then, the images of the subjects with the largest dilation differences were selected. We decided to use 1,500 iris images from each dataset. Finally, using the iCAM TD100 sensor, we created a dataset with 20 classes to study the effect of pupil dilation. Iris images were obtained after a subject was in a room in absolute darkness. Images were captured while the illumination was increased and decreased, to make the pupil dilate and constrict. The largest dilation was 0.60 and the smallest was 0.22, obtained on different subjects. The largest and smallest dilation range in a subject was 0.34 and 0.14, respectively. The iris images were obtained with the approval of the Bioethics Committee, Facultad de Ciencias Fisicas y Matematicas, Universidad de Chile (resolution No.011, May 19, 2019), and the signed informed consent was obtained from all subjects. After that, iris images were sorted according to their dilation level and the smallest number of available images in a class was selected, obtaining 30 images per class. Images in the rest of the classes were sampled uniformly according to their dilation increment. The iris dataset will be available on GitHub.1

IV. RESULTS AND DISCUSSION
A. LAYER SELECTION RESULTS
To decide the best layer to use as a feature extractor, ACC and FRR at FAR = 0.1% were obtained for all convolutional layers for each of the six CNNs in Table 1. Both metrics were depicted as a function of the layer position within each CNN showing the layers that provide the best results (red star) as feature extractors; see Figs. 5 - 10. These figures show the results obtained for ACC and FRR (at FAR = 0.1%) for all convolutional layers. For the CNNs DenseNet 201, NASNetMobile, ResNet 50, and Xception, the highest ACC and the lowest FRR were reached for the same layer. For Inception V3 and Inception ResNet V2, the maximum ACC and minimum FRR did not coincide, and the best layer was selected as the one with greatest difference between these two metrics. Table 2 summarizes the selected layer for each of the CNN models, showing the selected layer name, number, relative position, and fraction within the layers. The name of each layer is based on the Keras 2.3.1 framework [62]. Although the best layer is selected for each CNN, it can be observed in Fig. 5 - 10 that there are several possible layers that will yield similar IR results.

1https://github.com/JorgeZam/Dilation-Dataset-iCAMTD100

FIGURE 5. ACC (%) and %FRR (@FAR = 0.1%) as a function of the layer number for feature extraction in DenseNet 201. The best results were obtained for the selected layer number 14.
proposed method, therefore, allows finding many possible solutions using various layers as feature extractors from each CNN.

It should be noted that the best selected layers for iris feature extraction are located within the first 33% of the CNN architecture. It has been recognized [64] that the first layers in the CNN architecture extract general abstract patterns compared to those of the final layers that code more complex image information belonging to the ImageNet dataset [64][55]. Table 2 shows in the fourth column the fraction of the layers used for each of the CNN architectures on Table 1. Requiring only the first 33%, or less, of each CNN architecture is also an advantage in terms of the computational time required to extract features.

B. IRIS RECOGNITION AND MATCHING TIME RESULTS

Our results were compared to those of Zhao [27]. We used a partition with the same number of images as in [27] for a fair comparison, as is detailed in the Methods section. Zhao reported the ACC and the Equal Error Rate (EER). We are also reporting the total number of errors, which is the sum
of the false negatives (FN), plus the false positives (FP), and the matching time. In addition, we compare to Czajka’s approach [18] implemented in MATLAB [65]. The results are presented on Table 3. The best results are highlighted in bold text.

As can be observed on Table 3 regarding the CASIA Iris Lamp dataset, our method overtakes all the best previously published results [16], [27], and those obtained using the available GitHub [65]. The number of errors obtained by our method is also the least. From our six alternatives, DenseNet 201 reached the best result with the smallest number of errors. It can be observed in the last column of Table 3 that the lowest matching time was obtained with NASNetMobile, and Inception models. The most accurate model is DenseNet 201, while the fastest is NASNetMobile.

### TABLE 3. IR results in the CASIA Iris Lamp test partition (the test partition according to [27]). The first two rows correspond to the baseline, and the best model reported in [27]. The third row corresponds to the IR approach available in [65]. The six last rows are of our method.

| Method                      | ACC   | FP + FN | EER   | Matching Time [h:m:s] |
|-----------------------------|-------|---------|-------|-----------------------|
| Baseline (Daugman) [27]     | 92.4  | —       | 3.63  | —                     |
| VGG16 + DRDL [27]           | 93.87 | —       | 1.21  | —                     |
| BSIF Domain Specific [18]   | 98.47 | 17.151  | 1.5260| 12:40:52              |
| DenseNet201 [Ours]          | 99.56 | 4.022   | 0.4352| 00:36:30              |
| InceptionV3 [Ours]          | 99.27 | 6.739   | 0.7292| 00:10:27              |
| InceptionResNetV2 [Ours]    | 99.02 | 9.031   | 0.9773| 00:08:23              |
| NASNetMobile [Ours]         | 98.50 | 13.893  | 1.5034| 00:07:59              |
| ResNet50 [Ours]             | 99.34 | 6.091   | 0.6591| 00:54:59              |
| Xception [Ours]             | 99.42 | 5.318   | 0.5755| 01:50:31              |

Using the test partition of CASIA Iris Thousand in order to compare our results to those previously published [25], and to those obtained using the available GitHub [65], we obtained the results presented in Table 4. In this table, we used the FAR = 0.1% to measure the ACC to be able to compare our results to those previously published. Not only was the accuracy obtained higher, but also the matching time was significantly lower than that required by BSIF.

The most accurate models are DenseNet 201, Inception V3, and Xception. The main difference among these models can be observed in the false matches (number of errors FP + FN), with DenseNet having a better performance. The fastest model is NASNetMobile, again, as in its use with the CASIA Iris Lamp dataset, but with the lowest accuracy. The best metrics results appear in bold text on Table 4.

### C. DILATION ROBUSTNESS RESULTS

Our method was tested on three datasets formed by subjects with significant pupil dilation changes. These three datasets were described in the previous section. Results show the FRR at FAR = 0.1%, the Accuracy, and the number of False Positives and Negatives detected. The matching time was reported as well, demonstrating that our methods are faster compared to the baselines [18], [66]. Table 5 shows the results achieved on our dataset. Although the results of IR accuracy are similar, our models show better results in the numbers of FP and FN. The matching time required was also reduced, with the highest being 33 minutes, and the lowest at 109 seconds. The best performances were obtained by DenseNet 201 and ResNet 50, and the lowest required matching time was achieved by NASNetMobile. The best results appear in bold text. Table 6 and Table 7 present the results obtained on both the CASIA Iris subsets designed for pupil dilation, as was explained in the Methods section, part H. CASIA Iris Thousand, besides having subjects with significant dilation changes, included intra-class variations.
TABLE 5. IR results on our dataset with significant dilation changes. The first two rows correspond to results of the baselines [18], [66]. The six last rows show the results obtained with our method.

| Method                   | FRR   | ACC   | FP + FN | Matching Time [h:m:s] |
|-------------------------|-------|-------|---------|-----------------------|
| OS Iris V4.1 [66]       | 0.471 | 99.7129 | 1.637   | 03:21:15              |
| BSIF Domain Specific [18]| 0.931 | 99.5359 | 834     | 02:09:19              |
| DenseNet201 [Ours]      | 0.000 | 99.9983 | 3       | 00:09:49              |
| InceptionV3 [Ours]      | 0.011 | 99.9783 | 39      | 00:02:25              |
| InceptionResNetV2 [Ours]| 0.000 | 99.9883 | 21      | 00:01:54              |
| NASNetMobile [Ours]     | 0.034 | 99.9494 | 91      | 00:01:49              |
| ResNet50 [Ours]         | 0.000 | 99.9983 | 3       | 00:20:46              |
| Xception [Ours]         | 0.023 | 99.9661 | 61      | 00:33:38              |

TABLE 6. IR results for subset of CASIA Iris Thousand with significant dilation changes.

| Method                   | FRR   | ACC   | FP + FN | Matching Time [h:m:s] |
|-------------------------|-------|-------|---------|-----------------------|
| OS Iris V4.1 [66]       | 32.250| 88.4300| 130.076 | 10:14:50              |
| BSIF Domain Specific [18]| 21.632| 94.4149| 62.790  | 12:45:21              |
| DenseNet201 [Ours]      | 9.667 | 96.7224| 36.848  | 00:52:48              |
| InceptionV3 [Ours]      | 9.842 | 96.5751| 38.505  | 00:14:23              |
| InceptionResNetV2 [Ours]| 10.158| 96.1026| 43.816  | 00:11:16              |
| NASNetMobile [Ours]     | 16.371| 95.3253| 52.555  | 00:10:41              |
| ResNet50 [Ours]         | 10.586| 95.7383| 47.912  | 01:25:03              |
| Xception [Ours]         | 12.092| 96.1908| 42.825  | 02:29:44              |

such as eyeglasses, and specular reflections [12], [63]. For that reason, the performance achieved is lower than in the CASIA Iris Lamp version. Our results on lines 3-8 of Tables 6 and 7 show lower FRR compared to those of the baselines, OS Iris and BSIF [18], [66]. Our Accuracies are greater than those obtained with [18], [66]. Also, our method shows lower numbers of errors (FP + FN) on both Tables, and a lower matching time using NASNetMobile. The matching time was reduced from more than 10 hours to approximately 10 minutes.

D. FEATURE MAPS FILTERING - MASK EFFECT
In our method the feature maps were filtered after the feature extraction, and the best features were selected to reduce the effect of the mask contours. Table 8 shows the results of IR using our proposed method with filtering of the feature maps (the same as those of Table 4) compared with no filtering of the feature maps on the CASIA Iris Thousand dataset, test partition. In this last case, no filtering of the information of the mask contour in the channels of the Feature Map Tensors was performed for any of the CNN models. In Table 8 it can be observed that the accuracies increased, and the number of errors (FP + FN) decreased with our method by 13.10% and 99.08%, respectively, for the most accurate model (DenseNet 201). On the other hand, the improvements in accuracy and decrease in false matches (number of errors FP + FN) for our faster model (NASNetMobile) were 11.86% and 98.85%. Therefore, our method for minimizing the effect of the mask significantly improves the accuracy and decreases the number of errors.

E. FEATURE EXTRACTION TIME
As shown in Table 2, the fraction of layers (number of layers used/total number of layers) in the CNN architecture used for feature extraction represents less than 33% in the six proposed backbones. This ensures that the propagation time through the CNN layers in our method will be reduced compared to the time needed by the methods that use all the CNN layers. Table 9 shows the measured time required to extract features from a group of 5,000 images from the CASIA Iris Thousand, using the whole backbone (second column) compared to those of our method using the fraction of layers for the six backbones (detailed on Table 2). When our method was used with only a fraction of the architecture, the time required for feature extraction is even shorter. As shown in Table 9 the time required for feature extraction is reduced by 59.33% using our most accurate method (DenseNet 201). In the same way, our faster method in the matching stage (NASNetMobile) reduced the time for feature extraction by 47.18%. This improvement can have a significant impact in real large-scale applications.
TABLE 9. Times required for feature extraction using the whole backbone, and with our method using a fraction of the architecture as specified on Table 2 for each of the six CNNs. The fraction of the architecture used is shown in the fourth column on Table 2. The test was performed on a dataset of 5,000 iris images from the CASIA Iris Thousand.

| Backbone         | Using the whole backbone [m:s] | Using a fraction of the architecture as specified on Table 2 [m:s] |
|------------------|--------------------------------|---------------------------------------------------------------|
| DenseNet201      | 5.59                           | 3.33                                                          |
| InceptionV3      | 1:50                           | 1:23                                                          |
| InceptionResnetV2| 3:53                           | 1:32                                                          |
| NASNetMobile     | 3:31                           | 1:49                                                          |
| ResNet50         | 3:52                           | 1:54                                                          |
| Xception         | 4:13                           | 2:34                                                          |

TABLE 10. Computed hamming distance for the maximum rotations of ±11.25°. The test was performed on the test partition of the CASIA Iris Lamp dataset.

| Backbone         | HD at 11.25° | HD at -11.25° |
|------------------|--------------|---------------|
| DenseNet201      | 0.2946       | 0.2943        |
| InceptionV3      | 0.3878       | 0.3879        |
| InceptionResnetV2| 0.3992       | 0.3990        |
| NASNetMobile     | 0.3327       | 0.3342        |
| ResNet50         | 0.3349       | 0.3347        |
| Xception         | 0.2600       | 0.2587        |

F. SINGLE MATCHING AND SMALL IRIS ROTATIONS

To test the robustness of our method to small iris rotations, we performed horizontal displacements on the rubber sheet images equivalent to rotations in the range ±11.25° [18], [19]. Subsequently, the features maps were extracted and processed as explained previously. Then, we computed the Hamming Distance (HD) among the vectors representing the original image and those vectors from the rotated images, for each model. As expected, with no rotation the HD was zero, and the maximum HD was obtained for rotations of ±11.25°. We used the images of the CASIA Iris Lamp dataset test partition, to compute the HD. Table 10 shows the maximum HDs obtained for the maximum rotations (±11.25°), for each model. All the maximum HDs are under the decision threshold used for our results on Table 3 for CASIA Iris Lamp. These results are compatible with the fact that our method requires only a single matching instead of bit-shifting.

G. RUBBER SHEET CONTRAST ENHANCEMENT

In our method, rubber sheet images are preprocessed to enhance the texture of the iris tissue. Table 11 shows the accuracies using the proposed rubber sheet contrast enhancement, compared to those without the enhancement. In this test we used the CASIA Iris Lamp dataset test partition. Results on Table 11 show that accuracies increased, and the number of errors decreased with our contrast enhancement by 0.21% and 32.16%, respectively, for the most accurate model (DenseNet 201). The improvements in accuracy and decrease in false matches for our faster model (NASNetMobile) were 0.55% and 26.21%, respectively.

TABLE 11. Contrast enhancement results for CASIA Iris Lamp test partition.

| Backbone         | With Contrast-Enhancement | Without Contrast-Enhancement |
|------------------|---------------------------|------------------------------|
|                  | ACC | FP + FN | ACC | FP + FN |
| DenseNet201      | 99.56 | 4,022 | 99.35 | 5929 |
| InceptionV3      | 99.27 | 6,739 | 99.21 | 7,290 |
| InceptionResnetV2| 99.02 | 9,031 | 98.05 | 10,331 |
| NASNetMobile     | 98.50 | 13,893 | 97.96 | 18,828 |
| ResNet50         | 99.34 | 6,091 | 98.88 | 10,318 |
| Xception         | 99.42 | 5,318 | 97.91 | 19,275 |

V. CONCLUSION

We have proposed a novel method for IR, based on feature extraction by a CNN, that does not require a training stage. Iris features are extracted by using fewer than 33% of the convolutional layers of the most important pre-trained CNNs for image classification. This ensures that the propagation time through the CNN layers is reduced compared to that with methods that use all the CNN layers, as was shown on Table 9. Feature maps obtained were processed to reduce the effect of the mask contours on the rubber sheet images, by filtering the feature maps channels, and selecting the best features. It was also shown, on Table 8, that results in IR improve by using our proposed filtering of the feature maps, compared to when just using the mask on the rubber sheet. The proposed IR method does not require fine-tuning to be tested on different datasets since instead of using a classifier, a simple and fast matching among the codes is performed. Our approach requires only a single matching operation since the abstract features that are extracted from the CNN have the advantage of not being tied to specific spatial positions. This allows our method to reduce the matching time significantly compared to OSIris [66] and BSIF Domain-Specific [18] implementations. For example, on a dataset that contains 1,500 images, which involves 1,124,250 comparisons, the baselines registered more than 10 hours for the matching stage. In contrast, our fastest (based on NASNetMobile backbone), and slowest (based on Xception backbone) models for the same number of comparisons require approximately 10 minutes and 2.5 hours, respectively. The performance obtained by all our implemented models was above that of the previously published results on the CASIA Iris Lamp and CASIA Iris Thousand datasets, using the same partitions and datasets for a fair comparison. Among the backbones tested, DenseNet and Inception networks were the best models for iris feature extraction, obtaining the highest IR performance. In addition, our approach has improved IR results on datasets with large changes in pupil dilation, based on CASIA Iris Lamp and CASIA Iris Thousand, as well as on our own dataset, which was created specifically with eyes with wide pupil dilation differences.

REFERENCES

[1] J. Daugman, “Recognising persons by their iris patterns,” in Advances in Biometric Person Authentication, S. Z. Li, J. Lai, T. Tan, G. Feng, and Y. Wang, Eds. Berlin, Germany: Springer, 2005, pp. 5–25.
[2] J. Daugman, “Statistical richness of visual phase information: Update on recognizing persons by iris patterns,” Int. J. Comput. Vis., vol. 45, no. 1, pp. 25–38, Oct. 2001.

[3] H. Mehrrota, M. Vatsa, R. Singh, and B. Mahi, “Does iris change over time?” PLoS ONE, vol. 8, pp. 1–8, Nov. 2013, doi: 10.1371/journal.pone.0078333.

[4] D. P. Benalcazar, D. Bastias, C. A. Perez, and K. W. Bowyer, “A 3D iris scanner from multiple 2D visible light images,” IEEE Access, vol. 7, pp. 61461–61472, 2019.

[5] J. Daugman, “600 million citizens of India are now enrolled with biometric ID,” SPIE newsroom, vol. 7, May 2014. [Online]. Available: https://scholar.googleusercontent.com/scholar.bib?q=info:3F7yO3PsYmAJ:scholar.google.com&output=citation&scisf=4&ct=citation&cd=-1&hl=es

[6] J. E. Zambrano et al.: Iris Recognition Using Low-Level CNN Layers Without Training and Single Matching
