Alexandria Engineering Journal 112 (2025) 661–671

Contents lists available at ScienceDirect

Alexandria Engineering Journal

journal homepage: www.elsevier.com/locate/aej

Original article

GazeNet: A lightweight multitask sclera feature extractor

Matej Vitek *, Vitomir Štruc, Peter Peer

University of Ljubljana, Ljubljana, Slovenia

ARTICLE INFO

Keywords:
Biometrics
Ocular biometrics
Sclera recognition
Lightweight
Feature extraction

ABSTRACT

The sclera is a recently emergent biometric modality with many desirable characteristics. However, most literature solutions for sclera-based recognition rely on sequences of complex deep networks with significant computational overhead. In this paper, we propose a lightweight multitask-based sclera feature extractor. The proposed GazeNet network has a computational complexity below 1 GFLOP, making it appropriate for less capable devices like smartphones and head-mounted displays. Our experiments show that GazeNet (which is based on the SqueezeNet architecture) outperforms both the base SqueezeNet model as well as the more computationally intensive ScleraNET model from the literature. Thus, we demonstrate that our proposed gaze-direction multitask learning procedure, along with careful lightweight architecture selection, leads to computationally efficient networks with high recognition performance.

1. Introduction

Biometric recognition technology focuses on recognising individuals based on their body and behavioural characteristics, such as the face [1], fingerprints and fingermarks [2], DNA [3], retina [4], iris [5], periocular region [6], or ears [7,8]. Sclera recognition is a subfield of biometric identity recognition that instead utilises the vascular structures in the sclera (i.e. the white region of the eye). However, existing sclera recognition approaches often rely on excessively large and complex network architectures [9,10] with significant memory and computational cost. Sclera recognition is typically performed in four distinct steps [10]: sclera segmentation, vessel enhancement/segmentation, feature extraction, and identity matching. The identity matching is generally performed via simple vector distance computation with very low computational cost. The segmentation steps, on the other hand, tend to employ encoder–decoder architectures [11]. This double-network design, along with its often-employed skip connections, causes even the more lightweight models (such as RITnet [12,13]) to have a relatively high computational complexity and thus these models may require further compression for use on less capable hardware [14]. The feature extraction step, in contrast, commonly relies on classification-style models, which typically have a single-downstream-network design, allowing for lower computational complexity. For reference, RITnet [12] (designed to process 640 × 400 pixel images) is considered a lightweight segmentation solution and requires 16 GFLOPs (i.e. 16×10^9 floating point operations) for a single forward pass [14]. ScleraNET [9] (designed to process 400 × 400 pixel images), on the other hand, was not designed to be lightweight, but as we show in this paper, it still requires only 7.73 GFLOPs (12.5 GFLOPs if we processed 640 × 400 pixel images for consistency) for a single image. However, with model compression [14], the FLOP requirement of segmentation can be reduced, bringing its complexity on par with the feature extraction step. Since feature extraction relies on classification architectures and lightweight classification architectures already exist [15–22], our focus in this paper is on the selection of the appropriate architecture and the subsequent adaptations that can be made to improve its performance in the task of open-set sclera feature extraction. With this careful architecture design, we reduce the computational complexity of feature extraction without a performance cost and once again make segmentation the computational bottleneck of the entire recognition pipeline.

Specifically, in this paper, we propose two versions of GazeNet, a lightweight feature extraction network based on the SqueezeNet [18] architecture. The two versions (1.0 and 1.1) are of different sizes, similar to the original SqueezeNet [18]. The proposed GazeNet architecture contains two objective heads, designed for training in a multitask learning setup. We show that such multitask learning leads to better generalisation capabilities of the model, particularly with smaller, more lightweight networks. The results presented in this work show that the FLOPs for feature extraction can be decreased substantially relative to the literature approach without negatively impacting the overall recognition accuracy. It should be noted, however, that several lightweight networks we initially tested performed significantly worse. Among those were several visual transformers, specifically TinyViT [15], MetaFormer [16], and LVT [17], all of which failed

[*] Corresponding author.
E-mail address: matej.vitek@fri.uni-lj.si (M. Vitek).

https://doi.org/10.1016/j.aej.2024.11.011
Received 11 April 2024; Received in revised form 18 October 2024; Accepted 1 November 2024
Available online 14 November 2024

to converge in training. This is perhaps unsurprising since transformer architectures tend to require larger training datasets than CNNs [23], although steps have recently been proposed to alleviate this issue in vision transformers [24], in particular. However, even several lightweight CNNs, such as EfficientNet [19], MobileNet [20,21], and RegNet [22] performed poorly. Therefore, particularly at lower model complexities, carefully selecting the base architecture is crucial for successful recognition.

In summary, our paper makes the following contributions:

- We propose GazeNet, a novel two-headed lightweight network design based on the SqueezeNet architecture. GazeNet is trained to predict both the subject identity and the gaze direction of the image. The gaze direction is a crucial image characteristic in sclera recognition, as it determines which parts of the sclera vasculature are exposed in the image. As such, training the model to extract gaze-related features leads to better results, particularly in the lightweight scenario, where *(i)* parameter and feature efficiency is crucial and *(ii)* explicit vessel segmentation is not performed.
- Following established evaluation methodology [9], we comprehensively investigate how various image and subject characteristics affect recognition performance in the lightweight setting. We affirm the slight decrease in recognition accuracy on older subjects, while no such correlation is found in respect to subject gender, following similar observations made in [9]. In our evaluation, we show that GazeNet matches or outperforms all other methods, including the SqueezeNet architecture it is based on. GazeNet also displays a high degree of robustness to various perturbations in the recognition process, such as reduced image resolution or the removal of gaze directions in the subject galleries.
- We explore the correlation between model complexity and performance, and we show that our proposed GazeNet network lands in the sweet-spot of state-of-the-art performance and a sufficiently low FLOP count to be viable for deployment on modern lightweight devices (such as smartphones). As part of the experimental work we also perform a detailed analysis of the computational complexity of the GazeNet, SqueezeNet, and ScleraNet networks, as well as the classical feature extractor methods SIFT, SURF, ORB, and dense SIFT.

The remainder of the paper is structured as follows. Section 2 overviews the existing literature in the fields of sclera recognition and lightweight deep network development. In Section 3, the multitask GazeNet architecture, which is the chief contribution of this paper, is explained in detail. The same section also contains a figure overviewing the architecture, as well as a detailed analysis of the computational complexities of its components and of the other evaluated approaches. Section 4 contains the experimental results of the paper, including a comprehensive investigation into how various factors affect sclera recognition accuracy. The section compares our GazeNet network to the existing solutions from the literature (both classical hand-crafted and CNN-based), and it also studies the impact of our proposed training process in the various experimental settings through the comparisons of SqueezeNet and GazeNet. Finally, Section 5 concludes our paper with a summary of the presented results and the future outlook.

2. Related work

2.1. Sclera recognition

As discussed in the previous section, typical sclera recognition systems consist of three distinct steps that require significant computational power: (*i*) sclera segmentation, (*ii*) vessel segmentation/enhancement, (*iii*) feature extraction and identity matching. Each of these steps typically requires its own solution. The first step, sclera segmentation, extracts the relevant region of interest (ROI) from the image. It is so far the most studied step in the literature, and a recent source of sclera segmentation solutions is the Sclera Segmentation Benchmarking Challenge (SSBC), which was organised several years in a row at major biometric conferences [25–29]. The 2020 edition of the SSBC [11] showed the superiority of deep models over classical hand-crafted approaches to sclera segmentation. Research of the remaining two steps is scarcer. The idea of the explicit vessel segmentation/enhancement in the second step is to isolate the relevant informative structures in the ROI extracted in the previous step and eliminate the (mostly just noisy white) remainder of the sclera region. This step is explored more in-depth in [10], which points to superior performance of deep networks, although, as shown in [9], when we use deep networks, the step may not be crucial for successful performance. This is important when trying to design a lightweight solution, as it allows us to omit a key source of computational complexity in the entire recognition system. The final step extracts the key features of the vascular structures obtained in the previous steps into a feature vector, which can be compared for the purposes of identity matching. This step is explored in [9,30], where deep networks are again shown to be the best approach. In this work, we provide an extended analysis of feature extraction. We discuss various feature extraction techniques and once again show the superiority that deep recognition models exhibit over classical solutions.

2.2. Lightweight deep models

While the sclera has been gaining popularity as a biometric modality in recent years, most of the current research relies on heavy and computationally complex deep networks [11]. However, in real-world applications, there is growing demand for more computationally efficient solutions as biometric systems are being deployed more commonly on more lightweight devices, such as smartphones or head-mounted displays [13]. In the field of convolutional networks, model compression has been a common avenue of research, dating back all the way to the very inception of convolutional neural networks in the form of filter pruning [31]. Several other mechanisms of model compression have been conceived since, most prominently knowledge distillation [32,33], approximate multiplication [34], quantisation [35], weight sharing [36], low-rank approximation [37], and the Winograd transformation [38].

Many of the sclera segmentation models from the literature, such as the models from SSBC [11,28,29] competitions, are complex in terms of both computation and memory requirements, since they were not designed with less capable hardware in mind. Lightweight solutions are scarcer, although the OpenEDS competition [13] recently organised by Facebook focused on model complexity alongside its performance, and produced several lightweight solutions. However, due to the heavy weight and lower bound the organisers imposed on the memory footprint part of the scoring criterion, all the most successful performers (according to their score) were lightweight models that were designed from scratch to have precisely 1MB worth of parameters. Such lightweight design from scratch has been shown to outperform the model compression approaches listed above in certain cases [39], however it has the downside that it predefines a specific model size and performance when the model is designed, whereas model compression methods often allow for more control of the trade-off between model size and performance. For instance, this requirement of 1MB is already quite outdated in the case of mobile devices, as modern smartphones normally have several GB of memory available. Additionally, this approach does not allow any discussion of over- or under-parameterisation of the segmentation models, i.e. whether the model sizes and complexities could be reduced without a (significant) loss of accuracy in the given task, or whether the addition of extra parameters (usually in the form of extra layers or filters) would improve performance. This aspect is explored in more detail in [14],

Table 1
The ScleraNET architecture. Note that the convolutional layers (aside from the initial one) implement padding to maintain the feature map size, which is only downsized in the max-pooling layers. The first dotted line denotes the cut-off point where the network is pruned after training to function as a feature extractor (rather than a closed-set classifier) – this is why the FLOP computation excludes the layers after this line. The second dotted line separates the two heads that are connected in parallel to the feature extractor part of the network.

	Layer	Layer size*	Output size	FLOPs [M]	#Params. [k]
	Input		400 × 400 × 3		
	Conv	7 × 7/2 (128)	197 × 197 × 128	730	19
	MaxPool	2 × 2/2	98 × 98 × 128		
	Conv	3 × 3/1 (128)	98 × 98 × 128	1416	148
	Conv	3 × 3/1 (128)	98 × 98 × 128	1416	148
	MaxPool	2 × 2/2	49 × 49 × 128		
	Conv	3 × 3/1 (256)	49 × 49 × 256	708	295
	Conv	3 × 3/1 (256)	49 × 49 × 256	1416	590
	MaxPool	2 × 2/2	24 × 24 × 256		
	Conv	3 × 3/1 (512)	24 × 24 × 512	679	1180
	Conv	3 × 3/1 (512)	24 × 24 × 512	1359	2360
	MaxPool	2 × 2/2	12 × 12 × 512		
	AvgPool	12 × 12/1	512		
				$\Sigma = 7724$	$\Sigma = 4740$
ID	Dense	512	512		
	Dense	120	120		
Gaze	Dense	512	512		
	Dense	4	4		

* For convolutional/pooling layers: filter size/stride (number of filters).
For dense layers: number of neurons in layer.

Table 2
The GazeNet 1.0 architecture. The model is cut off at the first dotted line after training to function as a feature extractor (rather than a closed-set classifier), so the prediction heads are not included in the FLOP computation. The second dotted line separates the two heads that are connected in parallel to the feature extractor part of the network.

	Layer/Module	Filters*	Output size	FLOPs [M]	#Params. [k]
	Input		400 × 400 × 3		
	Conv	7 × 7/2 (96)	197 × 197 × 96	548	14
	MaxPool	3 × 3/2	98 × 98 × 96		
	Fire	16, 64, 64	98 × 98 × 128	113	12
	Fire	16, 64, 64	98 × 98 × 128	118	12
	Fire	32, 128, 128	98 × 98 × 256	433	45
	MaxPool	3 × 3/2	49 × 49 × 256		
	Fire	32, 128, 128	49 × 49 × 256	118	49
	Fire	48, 192, 192	49 × 49 × 384	251	105
	Fire	48, 192, 192	49 × 49 × 384	266	111
	Fire	64, 256, 256	49 × 49 × 512	452	189
	MaxPool	3 × 3/2	24 × 24 × 512		
	Fire	64, 256, 256	24 × 24 × 512	113	197
				$\Sigma = 2412$	$\Sigma = 734$
ID	Conv	1 × 1/1 (120)	24 × 24 × 120		
	AvgPool	24 × 24/1	120		
Gaze	Conv	1 × 1/1 (4)	24 × 24 × 4		
	AvgPool	24 × 24/1	4		

*For convolutional/pooling layers: filter size/stride (number of filters).
For Fire modules: $s_{1\times1}$, $e_{1\times1}$, $e_{3\times3}$.

where even the winner of OpenEDS [13], RITnet [12], is shown to be overparameterised in several sclera and ocular segmentation tasks.

Due to the encoder–decoder design common in segmentation models, it is quite difficult to find lightweight segmentation models in the context of sclera segmentation. As such, we need to rely on model compression methods described in the first paragraph of this section for the segmentation models [14]. In the feature extraction domain, on the other hand, we mainly rely on network architectures designed for image classification. Many such architectures are already designed in a lightweight manner, some of the most prominent being EfficientNet [19], MobileNet [20,21], SqueezeNet [18], ShuffleNet [40,41], and RegNet [22]. Recently, vision transformers [42] have become another common approach in image classification, and lightweight variations, such as TinyViT [15], MetaFormer [16], and LVT [17], soon followed with performances comparable to classical convolutional designs. However, these lightweight classification models have not yet been adapted to the task of open-set sclera vessel feature extraction that we require in sclera recognition. In this work we show that carefully selected lightweight architectures can be adapted to this task without a negative impact on the accuracy of recognition relative to more heavyweight feature extractors from the literature.

3. Methodology

This paper loosely follows the experimental design of [9], utilising the same AlexNet-like convolutional neural network ScleraNET [9]. We summarise the architecture of ScleraNET along with its required FLOPs to process a single 400 × 400 pixel image in Table 1. Note that the model's two heads are cut off after training to turn it into a feature extractor, applicable to open-set recognition. For details on the architecture, we refer the reader to [9,10]. It is important to note that we count *one multiplication and one addition* as a single operation when reporting the computational complexity of the models, as modern processor architectures implement such a pair as a single MAC (multiply-accumulate) instruction [43].

The lightweight recognition results of this paper are obtained with the 1.0 and 1.1 versions of our proposed GazeNet network. GazeNet is based on SqueezeNet [18], which is another AlexNet-like convolutional neural network that replaces convolutional layers in the downstream blocks with *fire* modules. A fire module consists of a 1 × 1 *squeeze* convolution layer with $s_{1\times1}$ filters, and an *expand* layer with $e_{1\times1}$ 1 × 1 filters and $e_{3\times3}$ 3 × 3 filters. The key idea of the fire module is to use 1 × 1 convolutions to *(i)* partially replace 3 × 3 convolutions, and *(ii)* limit the number of input channels into 3 × 3 convolutions. The network ends with a 1 × 1 convolutional layer with the same number of output channels as there are classes in the training dataset, followed by a global average pooling (GAP) layer. These two layers replace the fully connected layers in AlexNet, and are the two layers we cut off after training to turn the network from a closed-set classifier into an open-set feature extractor. For additional strategies behind the network design, as well as implementation details, we refer the reader to the original paper [18].

Our proposed GazeNet network, displayed in Fig. 1, relies on multitask learning. The network has a similar design to the SqueezeNet architecture, however it contains two predictive heads, each of which consists of a 1 × 1 convolution and a GAP layer. One of the heads is tasked with predicting the subject identity, while the other predicts the gaze direction in the input image. Note again that the heads are only present during learning and are cut off afterwards to turn the network into a feature extractor. As shown in [9], the gaze direction heavily affects matching accuracy, since different vascular structures are visible in different gaze directions. As such, in this paper we posit (and show through our experimental work) that such two-task training allows the model to better learn the crucial features, leading to higher performance and better generalisation. The overall architecture of our proposed network, along with its FLOP count, is detailed in Table 2.

The original SqueezeNet network comes in two different versions of different sizes: 1.0 and 1.1. For consistency's sake, we mirror this approach and propose corresponding 1.0 and 1.1 versions of our GazeNet network. As we can see from Table 2, the GazeNet 1.0 architecture requires 2.41 GFLOPs to process a single image, which is roughly $\frac{1}{3}$ of the computational complexity of ScleraNET. The 1.1 version reduces the number and size of the filters in the initial convolutional layer from 96 filters of size 7 × 7 to 64 filters of size 3 × 3 and it moves the max-pooling layers from after the 3rd and 7th fire blocks to after the 2nd and 4th fire blocks in order to downsize the input faster. With these changes, the GazeNet 1.1 version can process a single 400 × 400

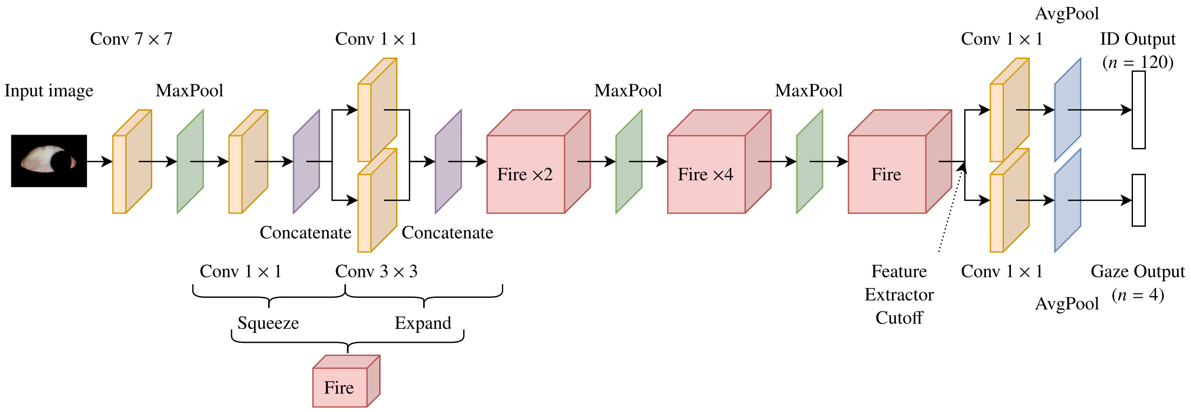


Fig. 1. The proposed GazeNet network, based on the SqueezeNet [18] architecture. Note the two output heads, one for predicting subject ID and one for predicting the gaze direction.

Table 3
The GazeNet 1.1 architecture. The model is cut off at the first dotted line after training, so the prediction heads are not included in the FLOP computation. The second dotted line separates the two heads that are connected in parallel to the feature extractor part of the network.

	Layer/Module	Filters*	Output size	FLOPs[M]	#Params. [k]
	Input		$400 \times 400 \times 3$		
	Conv	3 × 3/2 (64)	$199 \times 199 \times 64$	68	2
	MaxPool	3 × 3/2	$99 \times 99 \times 64$		
	Fire	16, 64, 64	$99 \times 99 \times 128$	110	11
	Fire	16, 64, 64	$99 \times 99 \times 128$	120	12
	MaxPool	3 × 3/2	$49 \times 49 \times 128$		
	Fire	32, 128, 128	$49 \times 49 \times 256$	108	45
	Fire	32, 128, 128	$49 \times 49 \times 256$	118	49
	MaxPool	3 × 3/2	$24 \times 24 \times 256$		
	Fire	48, 192, 192	$24 \times 24 \times 384$	60	105
	Fire	48, 192, 192	$24 \times 24 \times 384$	64	111
	Fire	64, 256, 256	$24 \times 24 \times 512$	109	189
	Fire	64, 256, 256	$24 \times 24 \times 512$	113	197
				$\Sigma = 870$	$\Sigma = 721$
ID	Conv	1 × 1/1 (120)	$24 \times 24 \times 120$		
	AvgPool	24 × 24/1	120		
Gaze	Conv	1 × 1/1 (4)	$24 \times 24 \times 4$		
	AvgPool	24 × 24/1	4		

*For convolutional/pooling layers: filter size/stride (number of filters).
For Fire modules: $s_{1\times1}$, $e_{1\times1}$, $e_{3\times3}$.

Table 4
The steps of the SIFT algorithm and their required FLOPs to process a single 400 × 400 pixel image. The FLOPs are estimated using formulas from [48]. The dotted line separates the feature extraction and feature vector matching stages. The matching stage assumes a template image in the gallery with its SIFT descriptors pre-computed.

Step	MFLOPs
Gaussian Blurring	48
Difference of Gaussian	2
Scale-space Extrema Detection	50
Keypoint Detection	0.2
Orientation Assignment	23
Keypoint Descriptor Generation	24
	$\Sigma = 147$
KNN-matching	8
	$\Sigma = 155$

pixel image with just 870 MFLOPs, as seen in Table 3. This is roughly $\frac{1}{3}$ of GazeNet 1.0 computational complexity, and an entire order of magnitude less than the complexity of ScleraNET.

Additionally, we compare our methods with classical descriptor-based methods SIFT [44], dense SIFT [45], SURF [46], and ORB [47]. The descriptor-based methods additionally rely on KNN when comparing the descriptor vectors of two images (for details see [44]). This matching process, unlike the cosine distance, has a non-negligible computational cost relative to the cost of feature extraction.

The computational complexity of SIFT is difficult to compute, as it depends on many parameters, as well as unpredictable factors, such as the number of keypoints detected in the image. Table 4 provides the number of FLOPs required to process a 400 × 400 pixel image using SIFT, estimated using the formulas from Table 2.1 in [48]. The formula for the KNN part is not provided in this work, but we can derive it using the same notation. There are $(\alpha \cdot \beta + \gamma) \cdot N^2$ descriptors computed in SIFT, where $N \times N$ is the size of the input image, α is the proportion of extrema among all pixels, β is the proportion of detected keypoints among all extrema, and γ is the fraction of keypoints added in the orientation assignment stage among all pixels, following the parameters' definitions from [48]. The concrete values of α, β, and γ used in our computation were determined experimentally on our input data. Since KNN needs to compute the distance between each pair of descriptors, it needs to compute roughly

$$\frac{((\alpha \cdot \beta + \gamma) \cdot N^2)^2}{2}$$

distances. By default, Euclidean distance is used in descriptor comparisons, which for two vectors of length n requires n subtractions, n multiplications (squares) and n additions (accumulation). Since a multiplication and addition comprise a single MAC processor instruction, this amounts to $2n$ total operations. Note that, depending on the processor architecture, subtractions may be significantly faster to compute than MAC operations, or the subtraction, squaring, and accumulation may even be implemented as a single EDAC (Euclidean Distance and Accumulate) operation, therefore this estimate may differ. The descriptor length in SIFT is $\frac{b \cdot x^2}{4}$, where b is the number of histogram bins and $2x \times 2x$ is the descriptor neighbourhood size used in the descriptor computation, again following the parameters' definitions from [48]. Putting it all together, the formula for the total number of FLOPs in KNN is:

$$\frac{((\alpha \cdot \beta + \gamma) \cdot N^2 \cdot x)^2 \cdot b}{4} \tag{1}$$

Table 5 adapts the formulas from Table 4 to dense SIFT by replacing $(\alpha \cdot \beta + \gamma) \cdot N^2$ (which corresponds to the total number of keypoints) with N, since instead of detected keypoints we use a $\sqrt{N} \times \sqrt{N}$ grid of points.

Table 5
The required FLOPs to process a single 400 × 400 pixel image using dense SIFT. The formula was adapted from the formulas in [48], replacing the detected keypoints in the SIFT algorithm with a dense grid of points. The dotted line separates the feature extraction and feature vector matching stages.

Step	MFLOPs
Gridpoint Descriptor Generation	39
KNN-matching	21
	$\Sigma = 60$

4. Experiments

Our experimental work in this paper follows a similar evaluation protocol to the one used in [9]. However, since our focus is on lightweight recognition systems, we adopt the approach from Case Study 3 in [9] to minimise the computational overhead introduced by the segmentation stages. Specifically, we evaluate our recognition models on segmented sclera images, without the additional explicit vessel segmentation step. The experimental dataset, train/test split, evaluation protocol, performance metrics, and baseline implementations follow the ones outlined in [9]. Throughout this section we additionally mark all the metrics with either ↑ (meaning higher is better) or ↓ (meaning lower is better) and the figures in this section are best viewed online zoomed-in and in colour. The experiments in this section compare the performance of the 1.0 and 1.1 versions of the lightweight GazeNet and SqueezeNet networks, the performance of the heavier ScleraNET model and the classical descriptor methods from [9]. Note that any comparisons of the corresponding (1.0 or 1.1) versions of GazeNet and SqueezeNet directly evaluate the impact of our proposed multitask training method, as the two network architectures are equivalent in all other respects. An additional case study is added, which explores the trade-off between recognition accuracy and computational complexity, and the final case study investigates the impact of our gaze-direction multitask learning process when faced with limited training data. To facilitate reproduction and further studying of these results, we make the code for all experimental work and model construction publicly available.[1]

4.1. GazeNet implementation details

We use the publicly available PyTorch implementation of SqueezeNet[2] [18] and append the parallel gaze prediction head. To ensure a fair comparison, we train the models on the same SBVPI data as the existing ScleraNET model [9], augmented via random rotations (up to 15°), horizontal and vertical translations (up to 10% of the image size), image rescaling (to between $\frac{1}{1.2}\times$ and 1.2× of the original size), shearing (up to 5°), brightness jitter (up to 0.1), and contrast jitter (up to 0.05). We train the models for 200 epochs, using the RMSprop optimiser with the categorical cross-entropy loss, defined as:

$$CCE = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic}\log p_{ic}, \tag{2}$$

where N is the total number of samples, C is the number of classes, y_{ic} are the true class labels (1 if sample i belongs to class c, 0 otherwise), and p_{ic} are the predicted probabilities that sample i belongs to class c (between 0 and 1). Since our two heads are trained on significantly different numbers of classes (there are 120 different identities, but only 4 different gaze directions in the training data), we additionally balance the two losses before combining them. As such, each of our predictive heads in fact uses class-balanced categorical cross-entropy, defined as:

$$CBCCE = -\frac{1}{N}\frac{1}{\log C}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic}\log p_{ic}. \tag{3}$$

The final loss is then obtained by combining the two predictive heads' losses, as the following:

$$L = \alpha \cdot CBCCE_i + (1-\alpha)\cdot CBCCE_g, \tag{4}$$

where $CBCCE_i$ is the identity-head loss, $CBCCE_g$ is the gaze-head loss, and α is the weighting parameter. The use of $CBCCE$ from Eq. (3) instead of CCE from Eq. (2) ensures the two losses are appropriately balanced, while the α parameter allows us to manually offset this balance as desired. We used the value of $\alpha = 0.5$ (i.e. fully-balanced loss) in our experimental work, however preliminary experiments showed a relatively consistent performance across non-extreme values of α. Note that the value of $\alpha = 1$ implies the gaze loss is ignored, making our network roughly equivalent to the original SqueezeNet, while $\alpha = 0$ ignores the identity loss, leading to significantly degraded performance.

50-fold random hyperparameter search was used to establish the concrete values for the initial learning rate (on the interval 1×10^{-7} to 1×10^{-2}), final learning rate (1×10^{-9} to 1×10^{-3}, only when using learning rate scheduling, which had a 50% probability), weight decay (0 to 0.01), and momentum (0.5 to 1) for each version of the models. In the hyperparameter search, the highest validation accuracy was achieved by all models with: learning rates in the range 2.2×10^{-6} to 4.4×10^{-5} with no (or minimal) scheduling, weight decay in the range 3.6×10^{-4} to 6.6×10^{-3}, and momentum in the range 0.6 to 0.95. The relatively small best-performance hyperparameter ranges (aside from momentum) demonstrate the high degree of sensitivity to hyperparameter selection in the training process. The models are trained at a batch size of 16 on an NVIDIA RTX A5000 GPU. Note that the left and right eye are considered to be different identities both in training and throughout the rest of this section.

4.2. Case Study 1: Sclera recognition

We first look at the overall recognition accuracy of the recognition methods on the 400 × 400 pixel sclera images, with the scleras segmented by the SegNet model from [9], using a gallery of 4 template images per subject per eye (1 for each gaze direction). Table 6 shows the detailed numerical results of the overall recognition experiments, reporting the mean (μ) and standard deviation (σ), computed over a 10-fold cross-evaluation, while Fig. 2 maps the verification rate (VER) and false acceptance rate (FAR) values over different match-score thresholds in the form of receiver operating characteristic (ROC) curves. From the ROC curves in Fig. 2(a), we can see that the 1.0 and 1.1 versions of the lightweight GazeNet network slightly outperform the heavier ScleraNET model, as well as all classical descriptor methods, overall. As seen in Fig. 2(b), when the recognition system is tuned towards a lower amount of false accept errors (i.e. low values of FAR), the two lightweight networks significantly outperform all other methods.

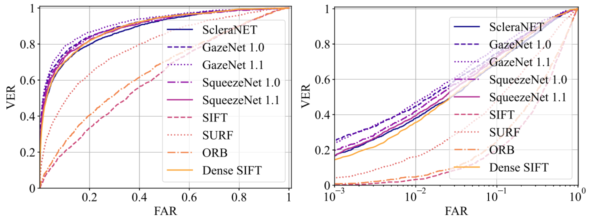
Since the networks were trained on greyscale segmentation maps, this case study can also be viewed as a test of the models' generalisation capabilities. Despite being trained on the vessel segmentation maps, the three deep networks all maintain a high level of performance on raw sclera images, implying the networks all learned a meaningful representation of the vascular structure, regardless of the exact form in which it is passed into the network. Of the four classical descriptor methods, only dense SIFT is competitive, while SIFT, SURF, and ORB all perform significantly worse.

[1] GitHub repository for the source code: https://github.com/MatejVitek/EyeZ-v2.

[2] Available from: https://pytorch.org/hub/pytorch_vision_squeezenet/.

Table 6
Results of the various recognition approaches with the best result in each column underlined. The lightweight networks achieve the best results overall, outperforming even the heavyweight ScleraNET CNN.

Model	VER@0.1FAR ↑	VER@1FAR ↑	EER ↓	AUC ↑
ScleraNET [9,10]	0.173 ± 0.003	0.381 ± 0.008	0.191 ± 0.002	0.889 ± 0.002
SqueezeNet 1.0 [18]	0.206 ± 0.004	0.427 ± 0.008	0.177 ± 0.003	0.902 ± 0.002
GazeNet 1.0 (ours)	$\mathbf{0.237 \pm 0.010}$	0.448 ± 0.009	0.174 ± 0.005	0.905 ± 0.004
SqueezeNet 1.1 [18]	0.170 ± 0.008	0.380 ± 0.011	0.190 ± 0.003	0.893 ± 0.003
GazeNet 1.1 (ours)	0.216 ± 0.005	$\mathbf{0.450 \pm 0.010}$	$\mathbf{0.172 \pm 0.003}$	$\mathbf{0.908 \pm 0.002}$
SIFT [44]	0.004 ± 0.002	0.031 ± 0.005	0.422 ± 0.006	0.611 ± 0.005
SURF [46]	0.044 ± 0.005	0.170 ± 0.005	0.293 ± 0.005	0.779 ± 0.004
ORB [47]	0.012 ± 0.001	0.054 ± 0.004	0.393 ± 0.006	0.650 ± 0.004
Dense SIFT [45]	0.144 ± 0.012	0.345 ± 0.012	0.192 ± 0.004	0.889 ± 0.003



(a) *Linear*

(b) *Semi-log*

Fig. 2. Results of the overall sclera recognition experiments. The CNN models and dense SIFT achieve the best overall performance.

4.3. Case Study 2: Recognition across subject characteristics

As subject characteristics can be an impactful source of performance differentials in sclera biometrics [49], we additionally study the impact of age and gender on the recognition performance. As in [9], we run the experiments in two settings:

- *within-group verification*, where both mated and non-mated [50] verification attempts are formed by pairing up images within a specific age/gender group;
- *between-group verification*, where non-mated attempts pair up images of different age/gender groups.

The recognition results of the two GazeNet models across different ages in Fig. 3 show that the models consistently perform worse on images belonging to older subjects, likely due to the decrease in the clarity of the vascular structure. No such conclusion can be drawn regarding gender (Fig. 4) – while the 1.0 architectures seems to perform marginally better on images from male subjects, there is not enough consistency in the differences to imply any significant impact of gender on the recognition accuracy. These results are in line with similar observations made in [9]. They may also partially be the result of unbalanced representation in the training data, since the training data distribution is skewed towards younger subjects, while being approximately gender-balanced [9], although we note that in [9] the age discrepancy was present even with descriptor methods, which are not trained on the data.

4.4. Case Study 3: Recognition with a limited gallery

In this case study we focus on the impact of gaze direction on recognition accuracy. Since the visible parts of the vascular structure differ greatly depending on the gaze direction, most of our experiments are performed with each subject having 4 template images with different gaze directions present in the gallery. Therefore, we study how removing one or more of those gaze directions from the subject's gallery affects the recognition accuracy. Specifically, we conduct experiments with 3/2/1 random gaze directions in each template, and a final set of experiments with only the frontal gaze in the template. The motivation behind the frontal-gaze setting is that the frontal gaze best exposes both parts of the sclera vasculature, implying that in this setting, methods that learn a holistic representation of the vascular structures will perform best.

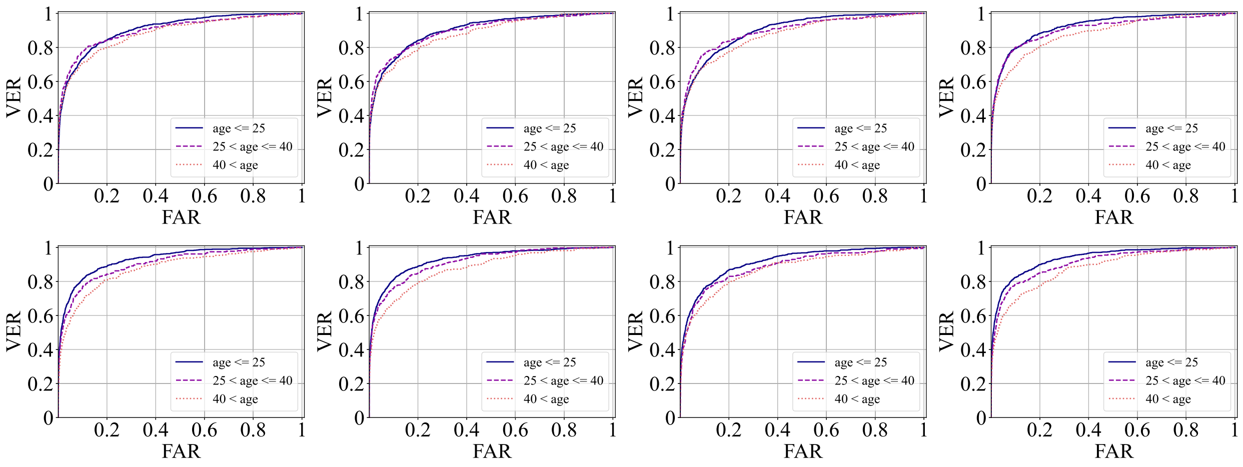
The results of this study are displayed in Fig. 5. While dense SIFT and ScleraNET perform comparably to the lightweight models with 3 gaze directions, once more gaze directions get removed, SqueezeNet 1.0 and both GazeNets all begin to noticeably outperform the other approaches. Additionally, all deep models significantly improve their results from the 1-random-gaze case to the frontal-gaze case, while with the descriptor-based methods there is no significant difference in performance between the two. This implies that the deep networks consistently learn the representation of the entire visible vasculature (which is most visible in the frontal gaze case) better than the descriptor methods, and the lightweight GazeNet and SqueezeNet once again display the best generalisation capabilities with more difficult variations of the problem task. In addition, the 1.0 and 1.1 versions of our GazeNet network again display the best performance overall, albeit with a small margin over SqueezeNet 1.0.

4.5. Case Study 4: Recognition with smaller resolution

Next we look at how different recognition methods performed with lower-resolution images. In the results of Fig. 6 we see that the lightweight networks outperform all other approaches (including ScleraNET) by a significant margin in the lower-resolution experiments. The 1.0 and 1.1 versions of SqueezeNet and GazeNet perform somewhat similarly down to the 192 × 192 resolution, while on the smaller resolutions the larger 1.0 outperforms the smaller 1.1 version. The 1.0 versions, in fact, maintain a stable performance even down to the 64 × 64 resolution. The superior performance of the lightweight networks over ScleraNET may point to the larger, more complex architecture focusing more on the details rather than the overall vascular structure, which is detrimental when the details are lost or made less distinguishable. Additionally, the ScleraNET network downsamples the input image twice as much as the SqueezeNet/GazeNet networks, which may also be detrimental with smaller image sizes. This observation is consistent with the fact that, while ScleraNET's performance starts degrading from the 192 × 192 resolution onwards, the 1.1 networks' performance remains stable until the 96 × 96 resolution, and only then begins to fall off. With both these observations in mind, we can see that shallower network architectures may in fact be better at dealing with lower-resolution input than their deeper heavyweight counterparts. Finally, our GazeNet again performs best overall, although in this case this best performance is achieved with the larger 1.0 version of the network.

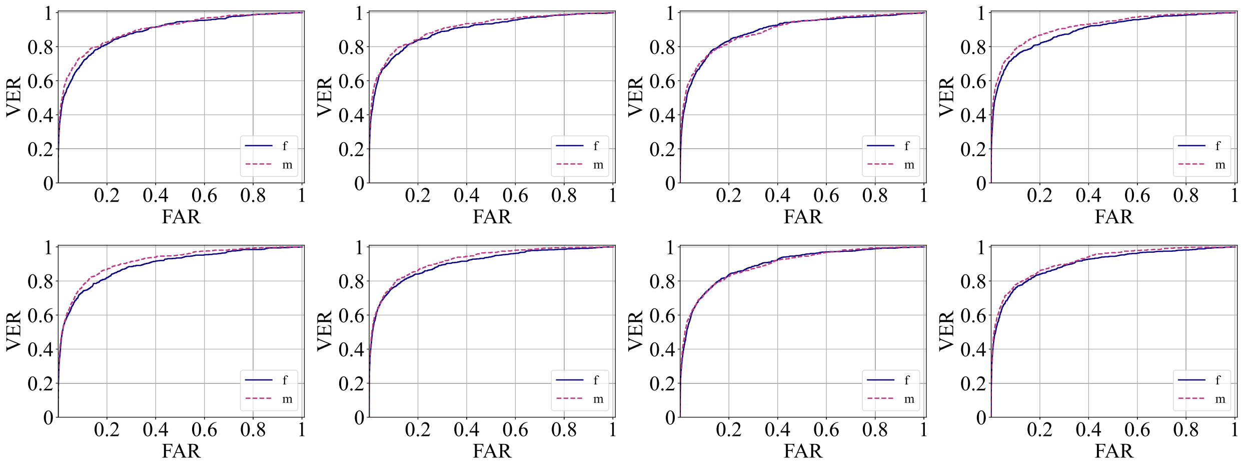
4.6. Case Study 5: Recognition across time complexities

The experimental work in the previous case studies was concerned only with the accuracy of the feature extraction methods. As this paper also focuses on the computational complexity of the recognition models, this case study explores the trade-off between model complexity and recognition accuracy. In Fig. 7 we show how well methods with



(a) *SqueezeNet 1.0* (b) *GazeNet 1.0* (c) *SqueezeNet 1.1* (d) *GazeNet 1.1*

Fig. 3. Verification results among different age groups. The top row contains results of the experiments with within-group non-mated attempts and the bottom row the results of the between-group experiments. The younger subjects consistently lead to higher recognition accuracy.



(a) *SqueezeNet 1.0* (b) *GazeNet 1.0* (c) *SqueezeNet 1.1* (d) *GazeNet 1.1*

Fig. 4. Verification results on male (m) and female (f) subjects. The top row contains results of the experiments with within-group non-mated attempts and the bottom row the results of the between-group experiments. The 1.0 architectures seem to perform slightly better on male subjects, while 1.1 do not exhibit any consistent differences in performance.

different computational complexities perform in feature extraction. The computational complexities of SURF and ORB are approximated from the formulas from Table 4, taking into account the differences between the algorithms described in their respective papers [46,47]. Since FLOP counts do not always correlate with practical performance, we provide a real-world comparison of the methods' execution times in Appendix.

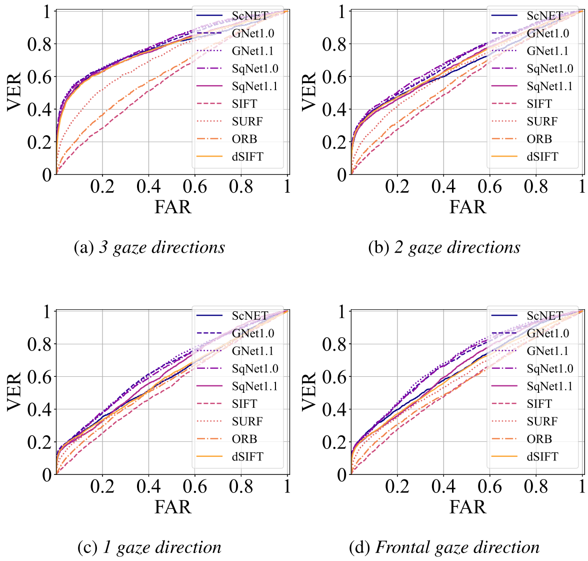
The plots of Fig. 7 once again demonstrate that, while the classical descriptor methods are less computationally intensive, this comes at a noticeable performance decrease. The descriptor-based methods also trend towards better performance from the more computationally intensive ones, with the exception of dense SIFT, which performs best, despite being on the low end of computational complexity. On the other hand, the ScleraNET network, despite requiring far more FLOPs to process an image, does not outperform the lighter-weight networks, and performs significantly worse than them at low FAR values. Additionally, our GazeNet 1.1 network performs the best overall, slightly outperforming the larger GazeNet 1.0 in most metrics, making GazeNet 1.1 the optimal choice in the accuracy/complexity trade-off (with the exception of dense SIFT, if excessively low computational complexity is required).

These results point to the fact that networks used in feature extraction in the task of sclera recognition can, with careful architecture design, be brought to the under-1-GFLOP domain without losing accuracy. What is more, at low FAR values, the lighter-weight networks in our experiments actually significantly outperform the heavyweight network. These observations are in line with similar observations made about sclera segmentation in [14], where lighter-weight models often outperformed their heavyweight counterparts.

4.7. Case Study 6: Recognition with limited training data

In the final case study, we perform an ablation study, investigating the effect of our novel training approach when the models are trained on limited training data. We train the two multitask models GazeNet 1.0 and 1.1, as well as their corresponding singletask counterparts SqueezeNet 1.0 and 1.1, on smaller parts of the training dataset, specifically at 10%, 33%, and 67% of the original training data.

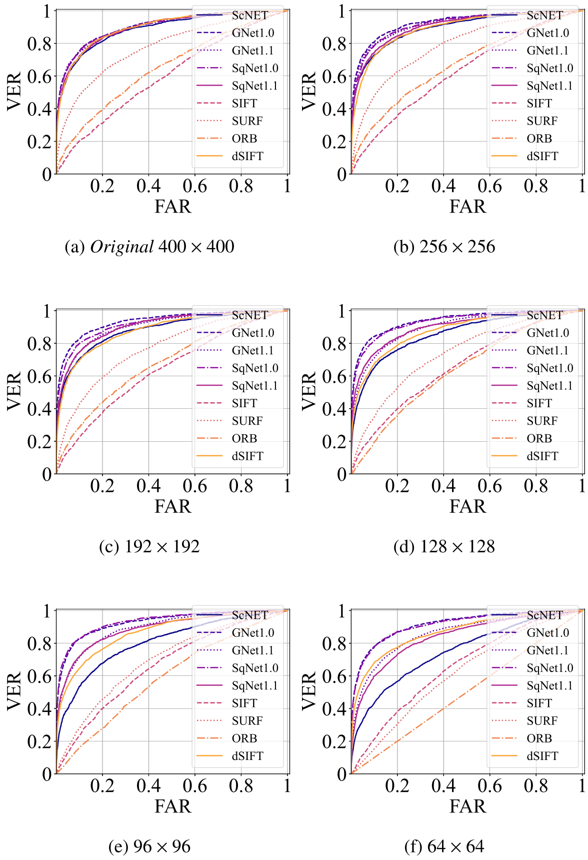
The results in Fig. 8 demonstrate that, particularly with the smaller 1.1 architectures, our multitask gaze-direction-based training approach leads to significant improvements in the model performance when trained on low amounts of data. Additionally, while the 10% datapoint is an outlier in several regards (mainly due to the intrinsic randomness of training at such low amounts of data), at the remaining datapoints (33%, 67%, 100%), both GazeNet models consistently improve their



(a) *3 gaze directions* (b) *2 gaze directions*

(c) *1 gaze direction* (d) *Frontal gaze direction*

Fig. 5. Verification results, given different gallery sizes. The first three experiments feature a random selection of 3/2/1 gaze directions in the gallery for each subject, while the last has only the frontal gaze. The deep models and dense SIFT perform comparably with the 3- and 2-random-view galleries, while in the single-view gallery experiments the lightweight models GazeNet 1.0 and 1.1, as well as SqueezeNet 1.0, outperform all other approaches.

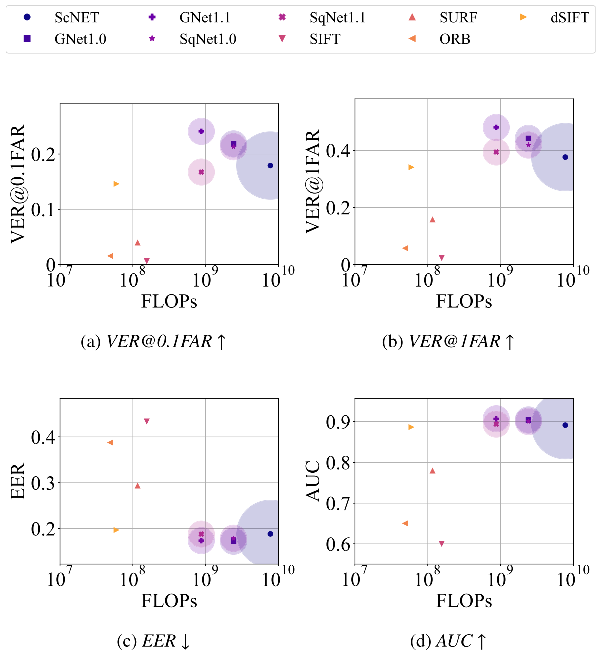


(a) *Original* 400 × 400 (b) 256 × 256

(c) 192 × 192 (d) 128 × 128

(e) 96 × 96 (f) 64 × 64

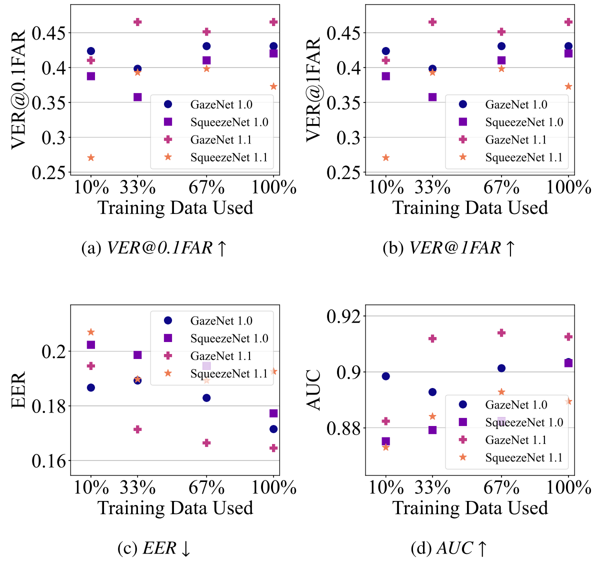
Fig. 6. Verification results on different input image resolutions. The first image gives the results of the original inputs, while the rest use input images downscaled with the Lanczos filter. The lightweight networks convincingly outperform all other approaches in the downsampled cases.



(a) *VER@0.1FAR* ↑ (b) *VER@1FAR* ↑

(c) *EER* ↓ (d) *AUC* ↑

Fig. 7. Trade-off between model complexity and recognition accuracy. The x-axis represents the models' computational complexities in FLOPs on a logarithmic scale. The areas of the translucent circles denote the number of parameters of the respective deep network (i.e. its memory footprint). The top row contains the metrics evaluating the performance at low amounts of false accepts, while the bottom row contains the metrics measuring the overall performance.



(a) *VER@0.1FAR* ↑ (b) *VER@1FAR* ↑

(c) *EER* ↓ (d) *AUC* ↑

Fig. 8. Verification results of models trained with limited amounts of training data. The 10% datapoint is an outlier, where the 1.0 architectures outperform the smaller 1.1 models and the performance at low FAR values differs significantly from the overall performance. However, at the remaining datapoints, the performance at low FAR values and overall performance are consistent, the 1.1 architectures consistently outperform their 1.0 counterparts, and all models aside from SqueezeNet 1.1 consistently improve their performance with more training data.

overall performance when more data is added to the training set, while SqueezeNet 1.1's performance stagnates and even drops slightly from the 67% to the 100% datapoint.

This ablation study demonstrates the positive impact of our multi-task training process, particularly in the case where only low amounts of training data are available. Since SBVPI is currently, to the best of our knowledge, the only available dataset with manual markups of the sclera vessels and only contains roughly 130 such markups, the ability to train models with limited data is important.

5. Conclusion

In this paper, we developed two versions (1.0 and 1.1) of our lightweight feature extractor network GazeNet. We showed that the classical descriptor-based methods are significantly less computationally intensive than deep convolutional networks, however they also tend to perform noticeably worse than the deep networks, particularly in terms of generalisation and in the case of more challenging inputs. On the other hand, we showed that lightweight deep models, even down to an order of magnitude smaller than existing solutions, can perform feature extraction with a comparable (and often even greater) recognition accuracy. The lightweight GazeNet models showed even greater robustness to inputs with different image characteristics (resolution in Section 4.5 or the absence of gaze directions in the subject gallery in Section 4.4) than the heavyweight ScleraNET [9]. This may point to the more complex ScleraNET model focusing too much on the details, colour characteristics, etc. of the vessel images, while the lightweight networks focus solely on the general structure of the vessels, leading to better results in these scenarios. Our extensive experimental work also confirmed the slight negative correlation between subject age and recognition performance seen in [9], while no (or minimal) correlation was found in regards to subject gender. With all the results in mind, our GazeNet 1.1 represents the best overall feature extractor in terms of the performance/complexity trade-off in our experiments, although GazeNet 1.0 is the better solution with excessively low-resolution images or excessively low amounts of training data, and dense SIFT is a reasonable alternative if even GazeNet 1.1's computational complexity exceeds our requirements, although this will rarely be the case in practice, since even lightweight solutions in the segmentation stages require at least an order of magnitude more computations, making GazeNet 1.1's computational complexity negligible in comparison.

For future work, we plan to expand the experimental work to more datasets, such as MOBIUS [49,51], as we are interested in how well the small models adapt to more diverse data being present in training and evaluation. Additionally, in line with the experiments from [14], we intend to study the bias present in the feature extraction stage, as well as whether the bias in segmentation and feature extraction models actually translates to a corresponding bias in the overall recognition accuracy. Next, since a well-designed enrolment was shown to be critical to successful recognition, particularly when it comes to gaze direction (see Section 4.4), we plan to further study enrolment; more specifically, we would like to reconstruct the entire vascular structure from different-gaze-direction images of the sclera. Finally, while we found in this paper that a carefully chosen small network architecture, coupled with our gaze-direction training, already performs well, the use of a pruning procedure (like IPAD from [14]) could alleviate some of the difficulties in selecting the right architecture for the task.

CRediT authorship contribution statement

Matej Vitek: Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Vitomir Štruc:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Peter Peer:** Writing – review & editing, Supervision, Resources, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table A.7
The time required for a single forward pass of each feature extractor model. The best execution times are presented in bold. Note that this comparison is highly specific to the concrete employed hardware, software language and framework, and implementation details.

Model	Time measurement [ms]		
	Feature extraction	Distance computation	Total
ScleraNET	57.13 ± 1.00	**0.0004 ± 0.00001**	57.13 ± 1.00
GazeNet 1.0	15.26 ± 0.78	0.22 ± 0.002	15.48 ± 0.78
GazeNet 1.1	**4.86 ± 0.14**	0.22 ± 0.002	**5.08 ± 0.14**
SIFT	25.14 ± 0.18	1.53 ± 0.002	26.66 ± 0.18
SURF	16.10 ± 0.018	0.65 ± 0.0006	16.75 ± 0.018
ORB	5.35 ± 0.027	1.00 ± 0.0003	6.35 ± 0.027
Dense SIFT	61.59 ± 0.11	2.3 ± 0.0006	63.86 ± 0.11

Acknowledgments

This research was supported by the ARIS Research Programme, Slovenia P2-0250 "Metrology and Biometric Systems", the ARIS Research Programme, Slovenia P2-0214 "Computer Vision", and NVIDIA, US's "Academic Hardware Grants" Programme, which provided the RTX A5000 GPU used in our research.

Appendix. Time complexity in practice

The analysis of the number of operations performed by each feature extractor in Section 3 is a good theoretical foundation for the comparison of the methods. However, in practice, the FLOP counts of different approaches often do not correspond exactly to the differences in execution times on real-life hardware. As such, we additionally measure the real time required by each of these methods to process a single image in Table A.7. Note, however, that such comparison is heavily reliant on the specific hardware, software framework choice, and implementation details. In our case specifically, our focus during implementation was on the accuracy of the results, not the optimisation of execution times. Thus, while all our approaches were implemented in Python,

- ScleraNET and GazeNet both run on the GPU, but were implemented in Keras[3] (with the Tensorflow backend) and PyTorch,[4] respectively;
- SIFT, SURF, and ORB were implemented using the Python wrappers of the OpenCV[5] library and all run on the CPU;
- Dense SIFT was also implemented with Python OpenCV and runs on the CPU, however, due to the specifics of the SIFT implementation in the library, the grid traversal had to be implemented in pure Python (rather than delegating to highly optimised C/C++ implementations, as the keypoint-based descriptor methods above do), slowing the execution down considerably.

While the computation capability of the GPU and CPU differs too much for accurate comparison, we can still compare the methods within those groups. We see that the methods' execution times quite closely follow the differences in FLOP counts outlined in Section 3, i.e., the GazeNet 1.1 network finishes a forward pass in roughly $\frac{1}{3}$ of the time of GazeNet 1.0, which in turn takes roughly $\frac{1}{4}$ of the time of ScleraNET. Likewise, SURF and especially ORB execute faster than SIFT, in line with their lower FLOP counts. Dense SIFT is the only outlier, taking a long time to finish a forward pass despite its low FLOP count, due to the implementation difficulties noted above.

[3] https://keras.io/
[4] https://pytorch.org/
[5] https://opencv.org/

References

[1] Y. Kortli, M. Jridi, A. Al Falou, M. Atri, Face recognition systems: A survey, Sensors 20 (2) (2020) 342.

[2] L. Premk, Ž. Emeršič, T. Oblak, Automatic latent fingerprint segmentation using convolutional neural networks, in: 2021 44th International Convention on Information, Communication and Electronic Technology, MIPRO, 2021, pp. 1010–1014, http://dx.doi.org/10.23919/MIPRO52101.2021.9597100.

[3] R.R. Garafutdinov, A.R. Sakhabutdinova, P.A. Slominsky, F.G. Aminev, A.V. Chemeris, A new digital approach to SNP encoding for DNA identification, Forensic Sci. Int. 317 (2020) 110520.

[4] R.M. Devi, P. Keerthika, P. Suresh, P.P. Sarangi, M. Sangeetha, C. Sagana, K. Devendran, Retina biometrics for personal authentication, in: Machine Learning for Biometrics, Elsevier, 2022, pp. 87–104.

[5] J. Lozej, D. Štepec, V. Štruc, P. Peer, Influence of segmentation on deep iris recognition performance, in: 2019 IEEE International Work Conference on Bioinspired Intelligence, IWOBI, 2019, pp. 1–6.

[6] P. Rot, M. Vitek, B. Meden, Ž. Emeršič, P. Peer, Deep periocular recognition: A case study, in: IEEE International Work Conference on Bioinspired Intelligence, IWOBI, 2019, pp. 21–26, http://dx.doi.org/10.1109/IWOBI47054.2019.9114509.

[7] Ž. Emeršič, A.K. SV, B. Harish, W. Gutfeter, J. Khiarak, A. Pacut, E. Hansley, M.P. Segundo, S. Sarkar, H. Park, et al., The unconstrained ear recognition challenge 2019, in: International Conference on Biometrics, ICB, IEEE, 2019, pp. 1–15.

[8] Ž. Emeršič, T. Ohki, M. Akasaka, T. Arakawa, S. Maeda, M. Okano, Y. Sato, A. George, S. Marcel, I.I. Ganapathi, S.S. Ali, S. Javed, N. Werghi, S.G. Işık, E. Sarıtaş, H.K. Ekenel, V. Hudovernik, J.N. Kolf, F. Boutros, N. Damer, G. Sharma, A. Kamboj, A. Nigam, D.K. Jain, G. Cámara-Chávez, P. Peer, V. Štruc, The unconstrained ear recognition challenge 2023: Maximizing performance and minimizing bias*, in: IEEE International Joint Conference on Biometrics, IJCB, 2023, pp. 1–10, http://dx.doi.org/10.1109/IJCB57857.2023.10449062.

[9] M. Vitek, P. Rot, V. Štruc, P. Peer, A comprehensive investigation into sclera biometrics: A novel dataset and performance study, Neural Comput. Appl. (NCAA) 32 (2020) 17941–17955, http://dx.doi.org/10.1007/s00521-020-04782-1.

[10] P. Rot, M. Vitek, K. Grm, Ž. Emeršič, P. Peer, V. Štruc, Deep sclera segmentation and recognition, in: A. Uhl, C. Busch, S. Marcel, R.N.J. Veldhuis (Eds.), Handbook of Vascular Biometrics, HVB, Springer, 2020, pp. 395–432, http://dx.doi.org/10.1007/978-3-030-27731-4_13.

[11] M. Vitek, A. Das, Y. Pourcenoux, A. Missler, C. Paumier, S. Das, I. De Ghosh, D.R. Lucio, L.A. Zanlorensi Jr., D. Menotti, F. Boutros, N. Damer, J.H. Grebe, A. Kuijper, J. Hu, Y. He, C. Wang, H. Liu, Y. Wang, Z. Sun, D. Osorio-Roig, C. Rathgeb, C. Busch, J. Tapia Farias, A. Valenzuela, G. Zampoukis, L. Tsochatzidis, I. Pratikakis, S. Nathan, R. Suganya, V. Mehta, A. Dhall, K. Raja, G. Gupta, J.N. Khiarak, M. Akbari-Shahper, F. Jaryani, M. Asgari-Chenaghlu, R. Vyas, S. Dakshit, S. Dakshit, P. Peer, U. Pal, V. Štruc, SSBC 2020: Sclera segmentation benchmarking competition in the mobile environment, in: IEEE International Joint Conference on Biometrics, IJCB, 2020, pp. 1–10, http://dx.doi.org/10.1109/IJCB48548.2020.9304881.

[12] A.K. Chaudhary, R. Kothari, M. Acharya, S. Dangi, N. Nair, R. Bailey, C. Kanan, G. Diaz, J.B. Pelz, RITnet: Real-time semantic segmentation of the eye for gaze tracking, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop, ICCVW, IEEE, 2019, pp. 3698–3702.

[13] S.J. Garbin, Y. Shen, I. Schuetz, R. Cavin, G. Hughes, S.S. Talathi, OpenEDS: Open eye dataset, 2019, arXiv Preprint URL https://arxiv.org/abs/1905.03702.

[14] M. Vitek, M. Bizjak, P. Peer, V. Štruc, IPAD: Iterative pruning with activation deviation for sclera biometrics, J. King Saud Univ.- Comput. Inf. Sci. 35 (8) (2023) 101630, http://dx.doi.org/10.1016/J.JKSUCI.2023.101630.

[15] K. Wu, J. Zhang, H. Peng, M. Liu, B. Xiao, J. Fu, L. Yuan, Tinyvit: Fast pretraining distillation for small vision transformers, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXI, Springer, 2022, pp. 68–85.

[16] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, S. Yan, Metaformer is actually what you need for vision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10819–10829.

[17] C. Yang, Y. Wang, J. Zhang, H. Zhang, Z. Wei, Z. Lin, A. Yuille, Lite vision transformer with enhanced self-attention, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11998–12008.

[18] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size, 2016, arXiv preprint URL https://arxiv.org/abs/1602.07360.

[19] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.

[20] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017, arXiv preprint URL https://arxiv.org/abs/1704.04861.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.

[22] I. Radosavovic, R.P. Kosaraju, R. Girshick, K. He, P. Dollár, Designing network design spaces, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10428–10436.

[23] W. Li, H. Luo, Z. Lin, C. Zhang, Z. Lu, D. Ye, A survey on transformers in reinforcement learning, 2023, arXiv preprint URL https://arxiv.org/abs/2301.03044.

[24] Y. Liu, E. Sangineto, W. Bi, N. Sebe, B. Lepri, M. Nadai, Efficient training of visual transformers with small datasets, Adv. Neural Inf. Process. Syst. 34 (2021) 23818–23830.

[25] A. Das, U. Pal, M.A. Ferrer, M. Blumenstein, SSBC 2015: Sclera segmentation benchmarking competition, in: Conference on Biometrics: Theory, Applications, and Systems, BTAS, 2015, pp. 742–747.

[26] A. Das, U. Pal, M.A. Ferrer-Ballester, M. Blumenstein, SSRBC 2016: Sclera segmentation and recognition benchmarking competition, in: International Conference on Biometrics, ICB, 2016, pp. 1–6.

[27] A. Das, U. Pal, M.A. Ferrer, M. Blumenstein, D. Štepec, P. Rot, Z. Emeršič, P. Peer, V. Štruc, S. Kumar, SSERBC 2017: Sclera segmentation and eye recognition benchmarking competition, in: International Joint Conference on Biometrics, IJCB, 2017, pp. 742–747.

[28] A. Das, U. Pal, M.A. Ferrer, M. Blumenstein, D. Štepec, P. Rot, P. Peer, V. Štruc, SSBC 2018: Sclera Segmentation Benchmarking Competition, in: International Conference on Biometrics, ICB, 2018, pp. 303–308.

[29] A. Das, U. Pal, M. Blumenstein, C. Wang, Y. He, Y. Zhu, Z. Sun, Sclera segmentation benchmarking competition in cross-resolution environment, in: IAPR International Conference on Biometrics, IEEE, 2019, pp. 1–7.

[30] S. Das, I. De Ghosh, A. Chattopadhyay, An efficient deep sclera recognition framework with novel sclera segmentation, vessel extraction and gaze detection, Signal Process., Image Commun. 97 (2021) 116349.

[31] Y. LeCun, J.S. Denker, S.A. Solla, Optimal brain damage, in: Advances in Neural Information Processing Systems, 1990, pp. 598–605.

[32] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, in: Neural Information Processing Systems (NeurIPS) Deep Learning Workshop, 2014, pp. 1–9.

[33] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, J. Wang, Structured knowledge distillation for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2604–2613.

[34] Y. Hu, T. Huang, R. Run, L. Yin, G. Li, X. Xie, PPBAM: A preprocessing-based power-efficient approximate multiplier design for CNN, in: 2022 IEEE International Conference on Integrated Circuits, Technologies and Applications, ICTA, IEEE, 2022, pp. 166–167.

[35] S. Mei, X. Chen, Y. Zhang, J. Li, A. Plaza, Accelerating convolutional neural network-based hyperspectral image classification by step activation quantization, IEEE Trans. Geosci. Remote Sens. 60 (2021) 1–12.

[36] E. Dupuis, D. Novo, I. O'Connor, A. Bosio, A heuristic exploration of retraining-free weight-sharing for cnn compression, in: 2022 27th Asia and South Pacific Design Automation Conference, ASP-DAC, IEEE, 2022, pp. 134–139.

[37] N. Kozyrskiy, A.-H. Phan, CNN acceleration by low-rank approximation with quantized factors, 2020, arXiv preprint URL https://arxiv.org/abs/2006.08878.

[38] R. Wu, F. Zhang, J. Guan, Z. Zheng, X. Du, X. Shen, Drew: Efficient winograd cnn inference with deep reuse, in: Proceedings of the ACM Web Conference 2022, 2022, pp. 1807–1816.

[39] M. Zhu, S. Gupta, To prune, or not to prune: exploring the efficacy of pruning for model compression, 2017, arXiv preprint URL https://arxiv.org/abs/1710.01878.

[40] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6848–6856.

[41] N. Ma, X. Zhang, H.-T. Zheng, J. Sun, Shufflenet v2: Practical guidelines for efficient cnn architecture design, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 116–131.

[42] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint URL https://arxiv.org/abs/2010.11929.

[43] IEEE standard for floating-point arithmetic, 2008, http://dx.doi.org/10.1109/IEEESTD.2008.4610935, IEEE Std 754-2008.

[44] D. Lowe, Object recognition from local scale-invariant features, 2, IEEE, 1999, pp. 1150–1157,

[45] A. Das, U. Pal, M.A.F. Ballester, M. Blumenstein, Sclera recognition using dense-SIFT, in: 2013 13th International Conference on Intellient Systems Design and Applications, IEEE, 2013, pp. 74–79.

[46] H. Bay, T. Tuytelaars, L. Van Gool, SURF: Speeded up robust features, in: European Conference on Computer Vision, Springer, 2006, pp. 404–417.

[47] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: An efficient alternative to SIFT or SURF, in: 2011 International Conference on Computer Vision, IEEE, 2011, pp. 2564–2571.

[48] P. Vinukonda, A Study of the Scale-Invariant Feature Transform on a Parallel Pipeline, Louisiana State University and Agricultural & Mechanical College, 2011.

[49] M. Vitek, A. Das, D.R. Lucio, L.A. Zanlorensi, D. Menotti, J.N. Khiarak, M.A. Shahpar, M. Asgari-Chenaghlu, F. Jaryani, J.E. Tapia, A. Valenzuela, C. Wang, Y. Wang, Z. He, Z. Sun, F. Boutros, N. Damer, J.H. Grebe, A. Kuijper, K. Raja, G. Gupta, G. Zampoukis, L. Tsochatzidis, I. Pratikakis, S. Aruna Kumar, B. Harish, U. Pal, P. Peer, V. Štruc, Exploring bias in sclera segmentation models: A group evaluation approach, IEEE Trans. Inf. Forensics Secur. (TIFS) 18 (2023) 190–205, http://dx.doi.org/10.1109/TIFS.2022.3216468.

[50] ISO/IEC information technology – vocabulary – part 37: Biometrics, 2022, ISO/IEC 2382-37:2022.

[51] O. Golob, P. Peer, M. Vitek, Semi-automated correction of MOBIUS eye region annotations, in: IEEE International Electrotechnical and Computer Science Conference, ERK, 2020, pp. 344–347.