IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 May 2023, accepted 24 May 2023, date of publication 31 May 2023, date of current version 19 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3281737

RESEARCH ARTICLE

Regularizing Binary Neural Networks via Ensembling for Efficient Person Re-Identification

AYSE SERBETCI[1], (Member, IEEE), AND YUSUF SINAN AKGUL[2]
[1]Department of Computer Engineering, Istanbul Commerce University, 34445 Istanbul, Turkey
[2]Department of Computer Engineering, Gebze Technical University, 41400 Kocaeli, Turkey

Corresponding author: Ayse Serbetci (asturan@ticaret.edu.tr)

ABSTRACT This study aims to leverage Binary Neural Networks (BNN) to learn binary hash codes for efficient person re-identification (ReID). BNNs, which use binary weights and activations, show promise in speeding up the inference time in deep models. However, BNNs typically suffer from performance degradation mainly due to the discontinuity of the binarization operation. Proxy functions have been proposed to calculate the gradients in the backward propagation, but they lead to the gradient mismatch problem. In this study, we propose to address the gradient mismatch problem by designing a multi-branch ensemble model consisting of many weak hash code learners. Specifically, our design aggregates the gradients from multiple branches, which allows a better approximation of the gradients and regularizes the network. Our model adds little computational cost to the baseline BNN since a vast amount of network parameters are shared between the weak learners. Combining the efficiency of the BNNs and hash code learning, we obtain an effective ensemble model which is efficient both in feature extraction and ranking phases. Our experiments demonstrate that the proposed model outperforms a single BNN by more than %20 using nearly the same amount of floating point operations. Moreover, the proposed model outperforms a conventional ensemble of BNN by more than %7 while being nearly 10x and 2x more efficient in terms of CPU consumption and memory footprint, respectively. We explore the performance of BNNs for efficient person ReID as one of the first systems available in the literature. Moreover, we adopt the proposed ensemble model for further validation of the image classification task and observe that our method effectively regularizes BNNs, providing robustness to hyperparameter selection and producing more consistent results under different settings.

INDEX TERMS Binary neural network, network regularization, hash retrieval, ensemble learning, person re-identification.

I. INTRODUCTION

Given a watch-list of people, a person ReID model is supposed to search and identify the same people through extensive video surveillance data [1], which is a resource-intensive task. Therefore, developing efficient ReID models is of crucial importance. Nevertheless, the state-of-the-art approaches mostly owe their remarkable accuracy to heavy models and high-precision similarity calculations in large dimensions [2], [3]. One of the most promising and widely studied directions for improving the computational efficiency of person retrieval is binary hash coding [4], [5], [6], [7], significantly reducing the memory requirements and enabling fast similarity approximations in Hamming space. Nevertheless, these models still necessitate a large number of real-valued matrix operations for query feature extraction.

Developing efficient neural architectures has become an active research field in recent years [8], [9]. One of the most promising models proposed for energy efficiency is Binary Neural Networks (BNNs) [10], [11]. BNNs perform bitwise operations in the forward pass using binarized weights and activations, leading to significant speed up the inference time. Hence, BNNs are a promising option for deploying deep models in tasks that require scalability, such as person ReID. However, training BNNs is challenging mainly due to the discontinuity of the binarization, which is incompatible with the

The associate editor coordinating the review of this manuscript and approving it for publication was Vivek Kumar Sehgal.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

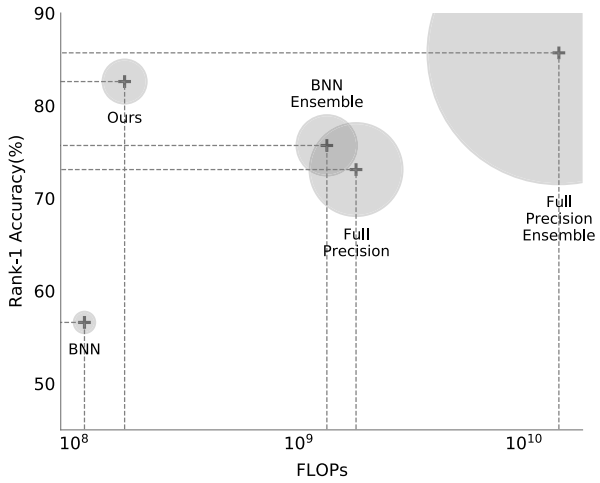


FIGURE 1. Accuracy vs. efficiency of the ReID models. Bubble size is proportional to the memory usage.

backpropagation algorithm [12]. The derivative of the `sign` function used for binarization is zero almost everywhere, intercepting the gradient flow. To overcome this problem, proxy functions are designed to approximate the gradients during the backward propagation [13], [14], [15]. Nevertheless, the gradient mismatch between the real and proxy functions inevitably degrades the performance.

This paper aims to exploit the efficiency of BNNs for person ReID by addressing the gradient mismatch problem and regularizing the network. To this end, we propose a multi-branch ensemble architecture where each branch corresponds to a weak learner projecting the input image into compact binary hash codes. In the proposed model a significant number of network parameters are shared, leading to two benefits: Firstly, the gradients from different branches are combined in the shared layers, which improves the gradient approximation and drives the weights towards binary values. Secondly, a powerful ensemble model is obtained without adding much computational cost to the BNN. The proposed model offers an ideal combination of the efficiency of BNNs, hash code learning, and the accuracy of ensemble models. Figure 1 presents a comparison between our model, a standard BNN, a full-precision counterpart, and their conventional ensemble models. We highlight that our proposed model not only outperforms the standard BNN, an ensemble of BNN, and the full-precision counterpart in terms of accuracy, but also demonstrates superior efficiency by using significantly fewer Floating Point Operations (FLOPs) and less memory. Specifically, it has a similar number of FLOPs to a standard BNN and is theoretically 10 times faster than the full-precision counterpart and a standard BNN ensemble. In contrast, its accuracy is comparable to a full-precision network, which serves as an upper bound. These results demonstrate the effectiveness of our proposed model in achieving a balance between accuracy and efficiency.

Our main contributions include

- We suggest a regularization method for BNNs to handle the gradient mismatch problem effectively.
- Our work is one of the first to investigate the performance of BNNs for efficient person ReID, and we propose an efficient ensembling model consisting of multiple hash code learners.
- Our experiments show that our proposed ensemble model outperforms conventional ensembles of BNNs significantly in terms of both accuracy and computational requirements.
- The proposed model performs comparably with the recent hashing methods while being several times more efficient.

The rest of this paper is organized as follows: Section II overviews the previous work while Section III explains the details of the proposed BNN ensemble model. In Section IV we present the experimental results on two benchmark datasets and further demonstrate the results on the image classification task. Finally, Section V includes concluding remarks.

II. RELATED WORK

A. BNNs

The deployment of deep neural networks on low-end devices is challenging due to their requirement for high-precision operations on GPU-based machines with large memory capacity [11], [16]. To address this issue, network quantization has emerged as an active area of research aimed at using low-bit values for the weights and activations to reduce computational requirements [17]. BNNs [10] representan extreme case of quantization using 1-bit representation, thus replacing the floating-point operations by XNOR-Bitcount operations [12]. However, the discreteness of the binarization brings difficulty in using standard backpropagation algorithm. To overcome this inconvenience, straight-through estimator is proposed as a workaround [18], which approximates the derivative of the `sign` function using `clip` or `identity` functions. However, the difference between the real and the proxy functions leads to gradient mismatch problem, making optimization tricky.

Recent work on BNNs have proposed many approaches for ease the training and reduce the gradient error. Reference [19] adopts knowledge-distillation from full-precision networks to guide the training process. Reference [20] introduces real-valued shortcut connections to enhance the gradient flow. Reference [21] improves the optimization function by adopting explicit regularization terms. References [15] and [22] refine the quantization function gradually during training. Reference [23] proposes to rotate the weight matrices to reduce the angular bias between the full-precision weight vector and its binarized version. To avoid the dependence on the sign function, [24] restricts the binarization to $\{0, +1\}$, resulting an efficient analytical solution for angle alignment. Reference [25] uses a series of sine functions in frequency

domain to better approximate the gradient. Reference [26] points out that the weights located ar the tails of the weight distribution are barely updated, resulting in slow convergence. They propose REctified Clap Unit to move these weights towards the distribution peak, thereby increasing the probability of updating them. Finally, [27] observes that the binary kernels learned in convolutional layers are clustered, enabling further compression and acceleration of BNNs.

B. FAST-ReID

Person ReID is a task where efficiency has crucial importance. However, only a small number of recent studies explicitly aim to tackle this problem. To speed up the feature extraction process, some methods design lightweight models for ReID. Reference [28] design a network for multi-granularity attention selection. Reference [29] proposed a compact model that aggregates multi-scale features. Reference [30] calculate the similarity of the input images in multiple levels. Reference [31] emphasized the use of small-sized network usage accelerates the exploration of architecture and training modifications. Reference [32] proposed a method based on Neural Architecture Search to automatically search for an effective CNN specialized for ReID, which alleviates the tedious task of manually designing compact networks.

To speed up the process of finding the nearest neighbors, prior work has utilized binary hash coding. Hashing methods create compressed binary codes to represent data, while still preserving the similarity between data items [33]. To leverage the powerful representational capability of deep CNNs, learning the image representation and hash coding in an end-to-end manner has become a promising approach. One proposed method is the part-based deep hashing approach, which optimizes the triplet loss on horizontal splits of the person image [5]. The binarization operation in hashing methods causes loss of information, leading to a decrease in accuracy. To improve the effectiveness of these models, [7] enforces robustness and topology consistency between hash code extraction and high-dimensional feature learning.

C. ENSEMBLE LEARNING

Ensemble learning is a technique that combines multiple weak learners to produce a strong learner. The accuracy of the ensemble models depends on the accuracy of the weak learners and the diversity of their decisions. DNN's are favorable base learners due to their performance on various tasks and their ability to become divergent with simple hyperparameter changes [34]. However, training and deployment of multiple DNNs can be time-consuming and computationally unaffordable. Various strategies have been developed to simplify ensemble models and reduce their costs, taking into account the trade-off between their accuracy and computational overhead. To reduce the training time, snapshot ensembles [35] utilize cyclical learning rate and take snapshots of the weights in different local minima during training, which produces an ensemble for the inference time. Similarly, [36] obtain an ensemble at the cost of training a single model by analyzing the loss surface and finding out the high-accuracy pathways. In [37], an efficient adaboost algorithm for CNN is proposed to improve the accuracy on imbalanced data. The parameters of the trained networks are transferred to the subsequent networks, which prevents overfitting on the data with limited sample size. Reference [38] reveals that sharing the first layers of the weak networks improves the classical ensemble in the image classification task. Reference [39] propose training a mother network and ''hatching'' many ensemble networks by expanding the mother network through function-preserving transformations. References [40] and [41] design a multi-branch architecture to effectively ensemble feature extractors for image representation. Reference [42] develop a classical ensemble of BNNs using bagging and boosting and show that ensembling increases the robustness and training stability of BNNs. However, this improvement comes at the cost of considerable increase in computational requirements. Previous works have mostly focused on developing efficient approaches to train traditional ensemble models using CNN as a weak learner. However, in this research, our focus is on designing an end-to-end ensemble model with the primary objective of regularizing the BNNs. To be more specific, we combine the gradient information from various branches to tackle the gradient mismatch problem. To our knowledge, our study is the first to suggest using end-to-end ensembling as an effective regularization method for BNNs.

III. OUR METHOD

The proposed multi-branch model consists of many lightweight binary hash code learners. To be consistent with the ensemble learning literature, we name these learners as weak learners. In our proposed model, these weak learners share a large number of network parameters in our proposed model. Conventional ensembling is a well-known way of regularization that combines the decisions of independently trained models. On the contrary, the weak learners in our model are trained jointly, which enables better approximations the gradients from the proxy function by aggregating gradients from multiple branches. This, in turn, effectively regularizes the BNN backbone. Our method leverages the inherent efficiency of BNNs and introduces two novel improvements: Firstly, weight-sharing between weak learners reduces the memory and computational costs in comparison with the conventional ensemble methods, which is critical for deployment on low-end devices. Secondly, binary hash code learning enables high-speed ranking in large-scale ReID datasets. We begin by introducing the overall architecture and then provide a detailed description of the model components and optimization.

A. BNN ENSEMBLE NETWORK

The proposed model includes three main components: A shared backbone network, multi-branch modules, and

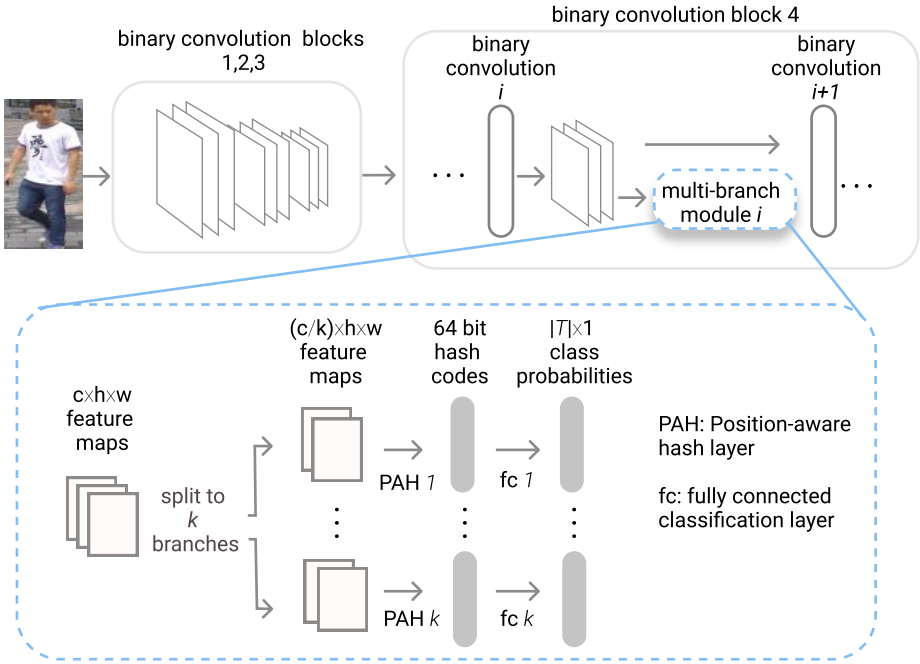


FIGURE 2. The overall view of the proposed model (upper) and the structure of the multi-branch module (lower).

position-aware hash code learners. In the following sections, we provide a detailed description of these components.

1) BACKBONE NETWORK

The upper part of Figure 2 illustrates the overall architecture, which only modifies the last block of the backbone network. The first layer is a standard convolutional layer (not shown in the figure for simplicity), followed by three binary convolutional blocks composed of stacked binary convolutional layers and batch normalization layers. In the last block, multi-branch modules are integrated after the binary convolution layers. This means that the output feature maps are fed into both the succeeding layer and the multi-branch module allowing the weak learners share most of the network parameters.

The number of multi-branch modules depends on the number of convolutional layers in the last block, which is 4 in our model. Note that the proposed architecture can be easily adapted to any architecture.

2) MULTI-BRANCH ARCHITECTURE

In Figure 2, the lower part shows the structure of the multi-branch modules. The feature maps obtained from the final convolutional layers are fed into a multi-branch module. To promote the diversity among the weak learners, the multi-branch module performs a channel-wise split operation on the input feature maps and sends them to different branches, depending on the number of weak learners to be included. In our case, we use $k = 8$ learners after each block resulting in a total of $8 \times 4 = 32$ weak learners. Each branch contains a position-aware hash code layer which ensures diversity among the weak learners. The position-aware hash layer transforms the input feature maps into binary hash codes and is followed by a classification layer that discriminates the input image into $|T|$ person identities in the training set. Note that the classification layers operate in high-precision numbers following the previous work [20]. However, they are ignored during the inference time, and the network is used as a hash function extractor, which is explained later. Thus their computational overhead of the high-precision classification layers can be ignored for the inference.

3) POSITION-AWARE BINARY HASH LAYERS

The diversity among weak learners in ensemble models has been shown to have a positive correlation with overall performance. [43]. To promote the diversity of the weak learners even further, we avoid the global average pooling operation before the hash layer and allow full connection with the preceding layer. The structure of the position-aware hash layers can be seen in Figure 3. We employ a binary convolutional layer to efficiently transform the input feature maps into compact binary codes. We call this layer position-aware because it preserves the spatial information of the input feature maps. Position-awareness prevents separate hash code learners from focusing only on the most discriminative features and boosts their diversity [41]. While this approach does increase model complexity, the binarization of weights and activations ensures that the additional overhead is negligible during inference time. In Section IV-E, we experimentally verify that the position-aware hash layers perform superior compared to their average pooled counterparts.

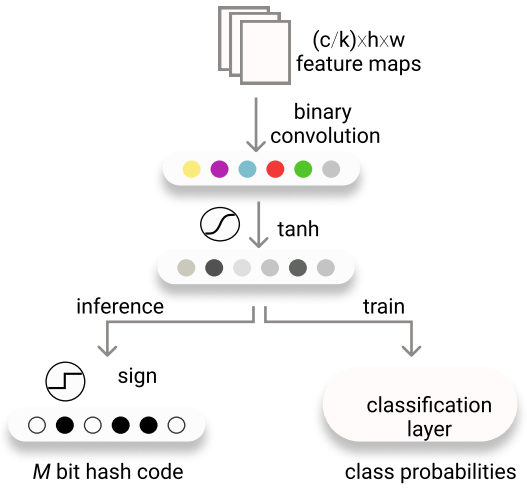


FIGURE 3. Position-aware binary hash code learning with binary convolutional layers.

Previous works on learning hash codes through deep networks typically involve a compact layer with binary activations before the classification layer. However, since the `sign` function used for binarization is not differentiable, most methods relax the binary constraint and use the hyperbolic tangent (*tanh*) function during the training. We use the same approach in this paper and employ hyperbolic tangent `tanh` on the outputs of the extracted features. As shown in Figure 3, in the training phase, the classification layer discriminates the person identities based on the embeddings using high-precision weights and activations. In the inference time, the extracted features are further binarized to be used in ranking in Hamming space.

B. TRAINING AND INFERENCE

CNNs that are used for person ReID are generally optimized with metric learning objective via triplet loss or classification objective using cross-entropy loss [44]. Triplet loss requires similarity calculation within each mini-batch to select the most informative triplets, which is not feasible in our case as there are many weak learners, each having their own embedding space. Therefore, we employ cross-entropy loss in our model. During training, a mini-batch passes through the shared blocks and is projected to different embedding spaces by the hash code layers of different weak learners. Let the training images be represented by X, and the corresponding identities are represented by Y. The categorical cross-entropy loss of a single weak learner i is denoted as $\mathcal{L}_i(X, Y)$ while the ensemble loss $\mathcal{L}_E(X, Y)$ is the sum of cross-entropy loss obtained from each learner i.

$$\mathcal{L}_E(X, Y) = \sum_i \mathcal{L}_i(X, Y). \tag{1}$$

During inference, the classification layers are not used, and the input image is transformed to compact binary codes by each weak learner. The resulting binary hash codes from different branches are concatenated to form the ensemble representation, which is utilized for the similarity computation in Hamming space. It is important to note that our model is scalable in terms of hash code length, meaning that the number of weak learners used can be adjusted to obtain a desired code length.

IV. EXPERIMENTAL RESULTS

In this section, we report the results of our experiments to evaluate the effectiveness of our model on two large-scale person ReID datasets: Market-1501 [45] and DukeMTMC-reid [46]. To this end, we define two baseline models : a standard BNN, which is a binarized ResNet18 architecture, and a classical ResNet18 as the full-precision counterpart. We also train a conventional ensemble of these baseline models. We first analyze the accuracy and efficiency of the proposed model and then compare it with recent hashing methods. Finally, we present the results of experiments that investigate the effect of hyperparameters and position-awareness.

To further investigate our method's regularization effect, we adapt our model to the image classification task. We design an experiment where we train our model multiple times with varying combinations of hyperparameters and compare our model with the baseline BNN model. The results show that our approach is also effective in enhancing the accuracy of BNNs in image classification, serving as a strong regularizer that provides robustness to the selection of hyperparameters.

A. EXPERIMENTAL SETUP

Before reporting the results, we will describe the implementation details of the two baseline models and our model, the datasets used for evaluation and the evaluation protocols.

1) IMPLEMENTATION DETAILS

We use Bi-RealNet-18 [20] as the BNN backbone in all experiments. We follow a representation learning objective and train all models with categorical cross-entropy loss. We resize the input images to 384×128. The data augmentation includes random cropping, horizontal flopping, and random erasing [47]. We train the networks with Adam optimizer for 200 epochs with a batch size of 128 and a learning rate of 0.01. Our method introduces two hyperparameters: the number of weak learners and the length of binary hash code. We set the total number of weak learners as $K = 32$ and the hash code length as $H = 64$. We analyze the impact of these hyperparameters in Section IV-D. In the BNN and full-precision baseline models, the backbone network is followed by a global average pooling layer, a binary hash code layer, and a classification layer. We have selected a binary hash code length of 128 to establish a strong baseline for our experiments.

2) DATASETS

Market-1501 dataset consists of 32,668 images of 1501 identities collected from 6 cameras. 750 identities are reserved for evaluation and the rest is used for training. We follow the same evaluation protocol as in [45]. DukeMTMC-reid dataset is a multi-target multi-camera pedestrian tracking dataset and consists of 85-minute videos from 8 cameras. A subset of this dataset has been used for image-based person ReID in [46] with a training and evaluation protocol similar to that of Market-1501. Training set includes 17,661 images of 702 individuals, while the evaluation set includes 2228 query images and 16,522 gallery images from the same 702 individual.

3) EVALUATION PROTOCOL

We adopt mean Average Precision (mAP) and Cumulative Matching Characteristics (CMC) as the evaluation metrics in our experiments. For efficiency analysis, we calculate memory consumption and the required number of FLoating Point Operations (FLOPs), following the previous work [11].

B. ACCURACY AND EFFICIENCY

Table 1 presents a comparison of the proposed model with the baselines and their ensemble models in terms of accuracy, memory usage, and CPU consumption. Figure 1 visualizes the same information as a bubble graph. The full-precision (FP) ensemble model outperforms other models, as expected. We observe that the classical ensemble method of BNN improves the baseline BNN model significantly but still cannot perform comparably with the FP baseline. Moreover, the computational cost and the memory requirement increase proportionally to the ensemble size and become closer to the full-precision baseline. On the other hand, our ensemble model outperforms all models and performs comparably with the full-precision (FP) ensemble, which serves as an upper bound. The results also demonstrate that our single weak learner outperforms the baseline BNN model despite using a smaller hash code length. It is worth noting that our proposed model adds only a negligible amount of FLOPs to a single weak learner, only about 10% of the network.

The objective of the BNN training is to obtain weights that cause minimal information loss when they are binarized. Therefore the weights are expected to be around to -1 and $+1$ at the end of the training. Our proposed ensemble model aggregates the gradients from multiple branches, which enables averaging out the gradient error and allows a better approximation. In Figure 4, we provide the cumulative distribution histogram of the weights in different training stages for both the baseline BNN and our model and report their Rank-1 accuracy. Our model pushes the weights towards the binary values starting from the early stages and, compared to the baseline, results in a more binary-like network with significantly fewer parameters far from the binary values. To provide a quantitative result, in Figure 5 we use cross-entropy as a measure of quantization error and calculate

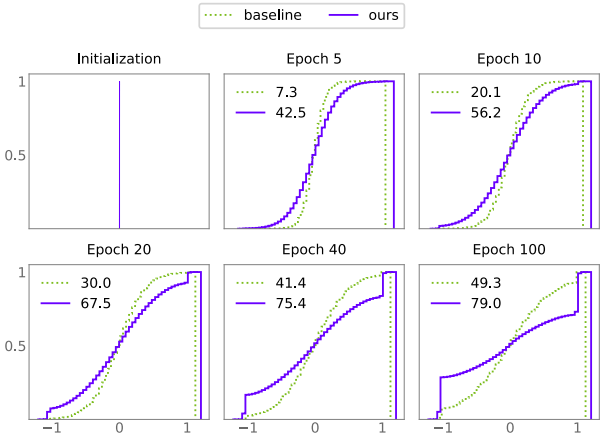


FIGURE 4. Comparison of the cumulative distribution of the network weights and their Rank-1 accuracy in different training stages. Dashed: Baseline(Standard BNN), Solid: Our model. The cdf of the initial weights seems as a single line due to uniform sampling from a small range [0,0.001].

the cross-entropy between the real weights of a BNN and its binarized version. As shown in the figure, starting from the early stages, our model has lower cross-entropy and higher Rank-1 accuracy than the baseline model. Both Figure 4 and 5 validate that our model effectively reduces the gradient error and allows for faster and accurate optimization.

Although we select Bi-RealNet-18 as a strong baseline, we validate our approach in a standard BNN (with ResNet-18 architecture) where (*tanh*) function is used for forward binarization and straight-through-estimator is used to calculate the gradients. We perform experiments on Market-1501 dataset and observe that standard BNN is not capable of learning discriminative feature vectors. More specifically, baseline model results in 10% top-1 and 5% mAP scores, respectively. By utilizing our ensemble model on the standard BNN, we observed a substantial enhancement in accuracy, although it remained inferior to the state-of-the-art (30% and 13.4% top-1 and mAP scores, respectively). This experiment confirms the applicability of our approach to different network architectures.

C. COMPARISON WITH HASH-BASED METHODS

Since our model is the first to utilize Binary Neural Networks (BNNs) for efficient ReID, it is not possible to make a fair comparison with existing works that use real-valued CNN models. Therefore, we compare our model with the most relevant direction of works, which are hash-based models. Please note that hash-based models define binary constraints only on the output of the last layer while the rest of the model is trained using real-values, which makes the comparison biased towards the hash-based models. The methods that we provide comparison with include ABC [6], PDH [5], Structured [48] and Self-guided [22], Deep Multi-Index Hashing [49] and Consistency-Preserving Deep Hashing [50].

The results are shown in Table 2. Our method outperforms most of the hash-based methods, while DMIH and CPDH

TABLE 1. Comparison with the standard BNN, full-precision counterpart and conventional ensemble models of these baselines.

Method	Bits	Market-1501		DukeMTMC		Memory	FLOPs
		R-1	mAP	R-1	mAP	(Mbits)	
FP	128	73.1	51.6	61.4	41.5	379.2	17.80×10^8
FP Ensemble	1024	85.7	67.9	74.7	57.7	3033	94.40×10^8
BNN (avg)	128	56.6	35.1	49.0	30.6	20.2	1.65×10^8
BNN Ensemble	128x8	75.7	53.9	65.1	47.2	161.6	13.2×10^8
Our weak (avg)	64	61.1	37.6	50.3	32.4	31.1	1.52×10^8
Our Ensemble	64x16	81.1	61.3	70.5	52.8	85.8	1.66×10^8
Our Ensemble	64x32	82.6	62.1	72.2	54.0	85.8	1.66×10^8

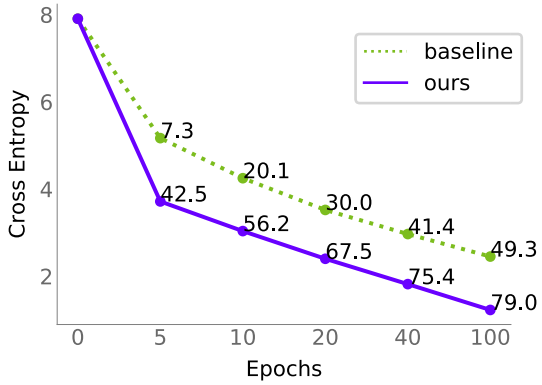


FIGURE 5. Cross entropy of the real weights and their binarized version both for baseline and our model and their corresponding Rank-1 accuracy.

TABLE 2. Comparison with hashing methods on Market-1501.

	128-bit		512-bit	
	R-1	mAP	R-1	mAP
ABC [6]	44.5	-	59.3	43.8
PDH [5]	36.3	19.6	44.6	24.3
Structured [48]	-	48.2	-	-
Self-Guided [22]	66.9	45.1	78.7	58.1
DMIH [49]	-	62.2	-	-
CPDH [50]	83.1	67.2	88.4	74.9
Ours-ResNet18	72.2	49.4	80.5	59.5

perform better than our model. Although we do not have the exact FLOPs and memory requirements for these methods, they are expected to have orders of magnitude higher than our proposed method because of their high-precision operations in the feature extraction phase.

D. PARAMETER ANALYSIS

We conduct experiments to analyze the impact of the two hyperparameters introduced by our model: binary hash code length of each weak learner and the ensemble size.In our proposed method, we divide the last convolutional layers into multiple branches, each functioning as an independent binary feature extractor during test time. The ensemble size (K), i.e., the number of weak learners, specifies the total number of branches while the code length H represents the binary hash code length used in each weak learner.

TABLE 3. Performance change with respect to ensemble size on Market-1501.

Bits (H)	Ensemble Size (K)	R-1	mAP
32	16	78.0	57.4
	32	80.8	59.7
	64	79.1	58.3
64	16	81.6	60.4
	32	82.6	62.1
	64	81.7	61.3

In order to investigate how the hash code length affects the performance of the proposed model, we trained ensemble models with $K = 32$ weak learners while varying the code length from $H = 16$ to $H = 128$. To evaluate the diversity between the weak learners with different code lengths, we calculate Rank-1 and mAP scores by combining the hash codes of different weak learners and obtain a cumulative measurement. The resulting performance curves in Figure 6 show a monotonic increase in accuracy as more weak learners are added to the ensemble, indicating that the weak learners are diverse and contribute positively to the overall accuracy. The code length has negligible effect (about 3%) on the number of FLOPs in our model, since most of the calculations are performed in the early convolutional layers. On the other hand, it should be noted that increasing the code length from $H = 64$ to $H = 128$ results in an increase of approximately 50% in the required memory size.

To analyze the effect of ensemble size, we train models with different numbers of weak learners. Specifically, we trained models with $K = 16$, 32, and 64 weak learners using code lengths H=32 and H=64. Table 3 shows the performance changes corresponding to different ensemble size. Our proposed model exhibits significantly better performance than the baseline BNN model (Table 1). Although the ensemble model with K=16 weak learners performed slightly worse than the other two models in either setting, our proposed model shows robustness to varying ensemble sizes

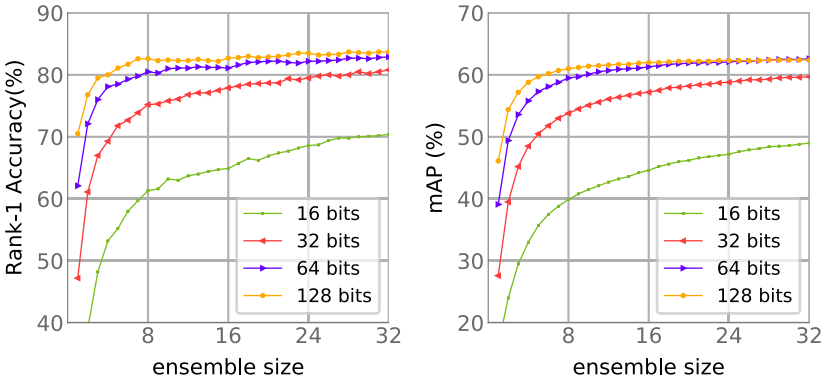


FIGURE 6. Cumulative ensemble performance on Market-1501 when using different code lengths.

as it performs similarly under different ensemble sizes when using code lengths $H = 32$ and $H = 64$.

E. ABLATION STUDY

In this section, we perform an ablation study to analyze the impact of the two main components of the proposed model: position-aware weak learners and end-to-end ensemble training. Specifically, we train our weak learners both with and without position-awareness as well as under single training (i.e., training only one weak learner) and ensemble training (i.e., joint training of weak learners) settings. To remove the position-awareness, we replace binary convolutional layers with a regular global average pooling layer before the hash code layer (Figure 3). Additionally, to examine the individual impact of ensemble training, we train our ensemble model with a single weak learner. We also investigate the effect of position-awareness on the baseline BNN.

The results are shown in Table 4, where we report Rank-1 and mAP scores under different settings. The upper part of the table presents that position-awareness increases the accuracy of the baseline model (position-awareness in the baseline model requires 5x more memory space). In the lower part, it can be observed that both position-awareness and ensemble training significantly increase the Rank-1 accuracy of the average weak learner when employed individually. The combination of both techniques, however, results in a much more substantial improvement. This demonstrates that ensemble learning reduces the effects of gradient mismatch problem and dramatically increases the accuracy of weak learners in either case. Additionally, we infer that global average pooling causes severe information loss and may not be ideal for hash-based retrieval tasks.

F. EVALUATION ON IMAGE CLASSIFICATION

Binarized neural networks (BNNs) are known to be more sensitive to hyperparameters compared to their floating point counterparts, making them challenging to optimize [51]. Consequently, effective regularization techniques are necessary to improve their training. To further verify the

TABLE 4. The effect of position-awareness on Market-1501.

Method	Bits	Pos-aw	Ensemble training	R-1	mAP
BNN (baseline)	128	-	-	56.6	35.1
BNN	128	✓	-	69.8	44.1
Our weak	64	-	-	22.1	11.5
Our weak	64	-	✓	38.4	22.5
Our weak	64	✓	-	42.4	28.6
Our weak	64	✓	✓	61.1	37.6
Our Ensemble	64x32	-	✓	68.8	46.7
Our Ensemble	64x32	✓	✓	82.6	62.1

effectiveness of our proposed model, we conducted an experiment for analysing its sensitivity to hyperparameters. We perform experiments on the image classification task using CIFAR10 [52] dataset training both the baseline model and the proposed model with different hyperparameters and monitor the validation accuracy during training. We use Bi-RealNet-18 as the backbone network with position-aware weak learners and we train the networks 350 epochs using SGD with momentum. The learning rate is decayed by 0.1 after epochs 150 and 250 and the classification scores of the weak learners were averaged to obtain the ensemble decision. We employ label smoothing regularization for training both models. We varied the epsilon hyperparameter in the label smoothing regularization and learning rate to investigate the behavior of the two models.

The validation accuracy with respect to the hyperparameters is given in Figure 7. We observe that the proposed model is more robust to hyperparameter selection and produces significantly more stable results. Notably, when both the learning rate and epsilon are set to 0.1 (the leftmost figure in the bottom row), the baseline model fails to converge and is not shown in the plot. On the other hand, our method can be trained appropriately and achieves satisfactory accuracy. The proposed model achieves better validation accuracy; the difference is particularly noticeable in some cases. These results indicate that our multi-branch BNN ensemble model is an effective regularization technique that can improve stability and accuracy.

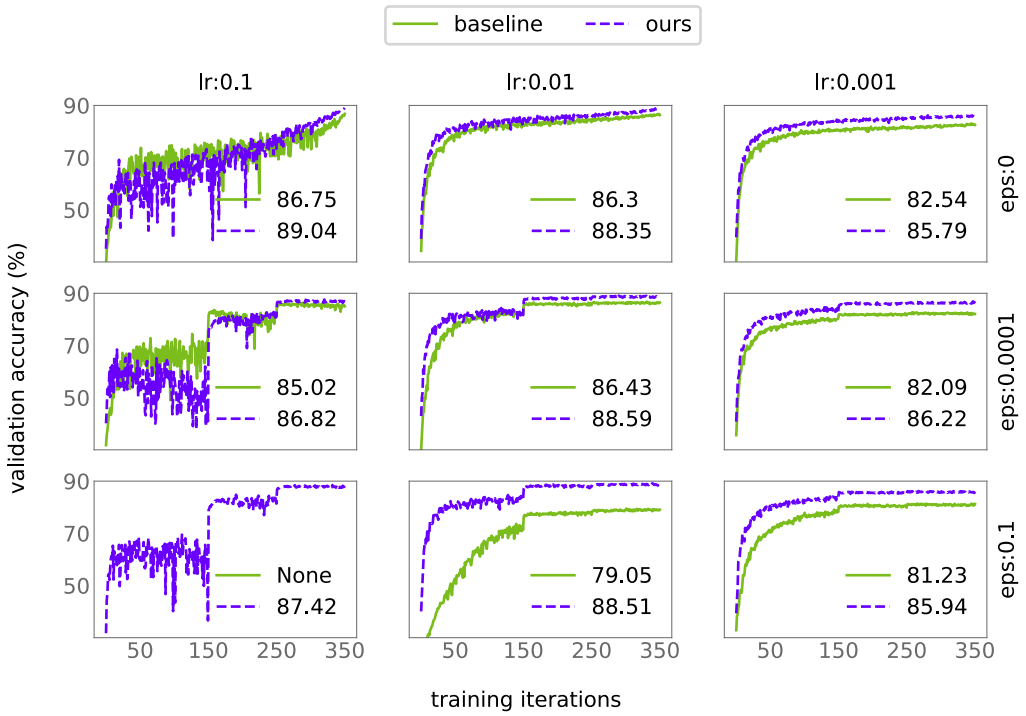


FIGURE 7. Performance comparison with the baseline model during the training under varying learning rate and epsilon hyperparameter settings.

V. CONCLUSION

This paper presented an efficient model for person ReID by regularizing BNNs with ensemble learning. Our design consists of a single backbone network with multiple branches as weak learners, which constitute the ensemble members. By training multiple weak learners jointly, we address the gradient mismatch problem by aggregating the gradients from multiple branches. As a result, we obtain a network whose real-valued weights are much closer to the binary values than that of the baseline BNN model. Our model outperforms the baseline BNN and a classical ensemble of BNN and the full-precision counterpart. Our model performs comparably with most state-of-the-art models despite being much more efficient in the feature extraction phase.

To our knowledge, this study is the first to investigate the performance of BNNs for image retrieval. In addition to image retrieval, we also evaluated our model on the image classification task under varying hyperparameter settings. The experiments confirm the regularization effect of the proposed model on BNNs, demonstrating more accurate results and robustness to hyperparameter selection.

REFERENCES

[1] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. H. Hoi, "Deep learning for person re-identification: A survey and outlook," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 2872–2893, Jun. 2022.

[2] C. Tay, S. Roy, and K. Yap, "AANet: Attribute attention network for person re-identifications," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7127–7136.

[3] T. Chen, S. Ding, J. Xie, Y. Yuan, W. Chen, Y. Yang, Z. Ren, and Z. Wang, "ABD-Net: Attentive but diverse person re-identification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8350–8360.

[4] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, "Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4766–4779, Dec. 2015.

[5] F. Zhu, X. Kong, L. Zheng, H. Fu, and Q. Tian, "Part-based deep hashing for large-scale person re-identification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4806–4817, Oct. 2017.

[6] Z. Liu, J. Qin, A. Li, Y. Wang, and L. Van Gool, "Adversarial binary coding for efficient person re-identification," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 700–705.

[7] G. Wang, S. Gong, J. Cheng, and Z. Hou, "Faster person re-identification," 2020, *arXiv:2008.06826*.

[8] C. Zhang, D. Wu, J. Sun, G. Sun, G. Luo, and J. Cong, "Energy-efficient CNN implementation on a deeply pipelined FPGA cluster," in *Proc. Int. Symp. Low Power Electron. Design*, Aug. 2016, pp. 326–331.

[9] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[10] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4107–4115.

[11] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 525–542.

[12] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, and N. Sebe, "Binary neural networks: A survey," *Pattern Recognit.*, vol. 105, Sep. 2020, Art. no. 107281.

[13] R. Gong, X. Liu, S. Jiang, T. Li, P. Hu, J. Lin, F. Yu, and J. Yan, "Differentiable soft quantization: Bridging full-precision and low-bit neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4851–4860.

[14] F. Lahoud, R. Achanta, P. Márquez-Neila, and S. Susstrunk, "Self-binarizing networks," 2019, *arXiv:1902.00730*.

[15] H. Qin, R. Gong, X. Liu, M. Shen, Z. Wei, F. Yu, and J. Song, "Forward and backward information retention for accurate binary neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2247–2256.

[16] Y. Guo, A. Yao, H. Zhao, and Y. Chen, "Network sketching: Exploiting binary structure in deep CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4040–4048.

[17] F. Tung and G. Mori, "Deep neural network compression by in-parallel pruning-quantization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 568–579, Mar. 2020.

[18] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013, *arXiv:1308.3432*.

[19] A. Mishra and D. Marr, "Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy," 2017, *arXiv:1711.05852*.

[20] Z. Liu, B. Wu, W. Luo, X. Yang, W. Liu, and K.-T. Cheng, "Bi-real Net: Enhancing the performance of 1-bit CNNs with improved representational capability and advanced training algorithm," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 722–737.

[21] R. Ding, T.-W. Chin, Z. Liu, and D. Marculescu, "Regularizing activation distribution for training binarized deep netwo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 11408–11417.

[22] X. Liu, S. Zhang, and M. Yang, "Self-guided hash coding for large-scale person re-identification," in *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, Mar. 2019, pp. 246–251.

[23] M. Lin, R. Ji, Z. Xu, B. Zhang, Y. Wang, Y. Wu, F. Huang, and C.-W. Lin, "Rotated binary neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 7474–7485.

[24] M. Lin, R. Ji, Z. Xu, B. Zhang, F. Chao, C.-W. Lin, and L. Shao, "SiMaN: Sign-to-magnitude network binarization," 2021, *arXiv:2102.07981*.

[25] Y. Xu, K. Han, C. Xu, Y. Tang, C. Xu, and Y. Wang, "Learning frequency domain approximation for binary neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 25553–25565.

[26] Z. Xu, M. Lin, J. Liu, J. Chen, L. Shao, Y. Gao, Y. Tian, and R. Ji, "ReCU: Reviving the dead weights in binary neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5178–5188.

[27] Y. Wang, Y. Yang, F. Sun, and A. Yao, "Sub-bit neural networks: Learning to compress and accelerate binary neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5340–5349.

[28] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2285–2294.

[29] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3701–3711.

[30] Y. Guo and N. Cheung, "Efficient and deep person re-identification using multi-level similarity," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2335–2344.

[31] H. Lawen, A. Ben-Cohen, M. Protter, I. Friedman, and L. Zelnik-Manor, "Compact network training for person ReID," in *Proc. Int. Conf. Multimedia Retr.*, Jun. 2020, pp. 164–171.

[32] R. Quan, X. Dong, Y. Wu, L. Zhu, and Y. Yang, "Auto-ReID: Searching for a part-aware ConvNet for person re-identification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3749–3758.

[33] Z. Cao, M. Long, J. Wang, and P. S. Yu, "HashNet: Deep learning to hash by continuation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5609–5618.

[34] X. Zhang and D. Wang, "A deep ensemble learning method for monaural speech separation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 5, pp. 967–977, May 2016.

[35] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get M for free," 2017, *arXiv:1704.00109*.

[36] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, "Loss surfaces, mode connectivity, and fast ensembling of DNNs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8789–8798.

[37] A. Taherkhani, G. Cosma, and T. M. McGinnity, "AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning," *Neurocomputing*, vol. 404, pp. 351–366, Sep. 2020.

[38] S. Lee, S. Purushwalkam, M. Cogswell, D. Crandall, and D. Batra, "Why M heads are better than one: Training a diverse ensemble of deep networks," 2015, *arXiv:1511.06314*.

[39] A. Wasay, B. Hentschel, Y. Liao, S. Chen, and S. Idreos, "MotherNets: Rapid deep ensemble learning," in *Proc. Mach. Learn. Syst.*, 2020, pp. 199–215.

[40] M. Opitz, G. Waltner, H. Possegger, and H. Bischof, "BIER—Boosting independent embeddings robustly," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5199–5208.

[41] A. Serbetci and Y. S. Akgul, "End-to-end training of CNN ensembles for person re-identification," *Pattern Recognit.*, vol. 104, Aug. 2020, Art. no. 107319.

[42] S. Zhu, X. Dong, and H. Su, "Binary ensemble neural network: More bits per network or more networks per bit?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4918–4927.

[43] L. Liu, W. Wei, K. Chow, M. Loper, E. Gursoy, S. Truex, and Y. Wu, "Deep neural network ensembles against deception: Ensemble diversity, accuracy and robustness," in *Proc. IEEE 16th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Nov. 2019, pp. 274–282.

[44] D. Wu, S.-J. Zheng, W.-Z. Bao, X.-P. Zhang, C.-A. Yuan, and D.-S. Huang, "A novel deep model with multi-loss and efficient training for person re-identification," *Neurocomputing*, vol. 324, pp. 69–75, Jan. 2019.

[45] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1116–1124.

[46] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by GAN improve the person re-identification baseline in vitro," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3774–3782.

[47] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," 2017, *arXiv:1708.04896*.

[48] L. Wu, Y. Wang, Z. Ge, Q. Hu, and X. Li, "Structured deep hashing with convolutional neural networks for fast person re-identification," *Comput. Vis. Image Understand.*, vol. 167, pp. 63–73, Feb. 2018.

[49] M.-W. Li, Q.-Y. Jiang, and W.-J. Li, "Deep multi-index hashing for person re-identification," 2019, *arXiv:1905.10980*.

[50] D. Li, Y. Gong, D. Cheng, W. Shi, X. Tao, and X. Chang, "Consistency-preserving deep hashing for fast person re-identification," *Pattern Recognit.*, vol. 94, pp. 207–217, Oct. 2019.

[51] S. Leroux, B. Vankeirsbilck, T. Verbelen, P. Simoens, and B. Dhoedt, "Training binary neural networks with knowledge transfer," *Neurocomputing*, vol. 396, pp. 534–541, Jul. 2020.

[52] A. Krizhevsky, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.



AYSE SERBETCI (Member, IEEE) received the B.S. degree in computer engineering from Hacettepe University, Ankara, in 2009, and the Ph.D. degree in computer engineering from Gebze Technical University, Kocaeli, in 2020. From 2015 to 2019, she was a Research Assistant with the Computer Engineering Department, Gebze Technical University. Since 2022, she has been an Assistant Professor with the Computer Engineering Department, Istanbul Commerce University, Turkey.



YUSUF SINAN AKGUL received the B.S. degree in computer engineering from Middle East Technical University, Ankara, Turkey, in 1992, and the M.S. and Ph.D. degrees in computer science from the University of Delaware, in 1995 and 2000, respectively. He was a Senior Vision Engineer with Cognex Corporation, Natick, MA, USA, from 2000 to 2005. He is currently with the Department of Computer Engineering, Gebze Technical University, Turkey. His research interests include the application of computer vision in medical image analysis, video analysis, object recognition, 3D analysis, and industrial inspection.