electronics

MDPI

Article

Weight Quantization Retraining for Sparse and Compressed Spatial Domain Correlation Filters

Dilshad Sabir [1,*], Muhammmad Abdullah Hanif [2], Ali Hassan [1], Saad Rehman [3] and Muhammad Shafique [4]

[1] Department of Computer and Software Engineering, College of Electrical and Mechanical Engineering (E&ME), National University of Sciences and Technology, Islamabad 44000, Pakistan; alihassan@ce.ceme.edu.pk
[2] Institute of Computer Engineering, Technische Universität Wien (TU Wien), 1040 Vienna, Austria; muhammad.hanif@tuwien.ac.at
[3] Faculty of Computer Engineering, HITEC University, Taxila 47080, Pakistan; saad.rehman@hitecuni.edu.pk
[4] Division of Engineering, New York University Abu Dhabi (NYU AD), Abu Dhabi 00000, United Arab Emirates; muhammad.shafique@nyu.edu
* Correspondence: dsabir.ce14ceme@student.nust.edu.pk

Abstract: Using Spatial Domain Correlation Pattern Recognition (CPR) in Internet-of-Things (IoT)-based applications often faces constraints, like inadequate computational resources and limited memory. To reduce the computation workload of inference due to large spatial-domain CPR filters and convert filter weights into hardware-friendly data-types, this paper introduces the power-of-two (Po2) and dynamic-fixed-point (DFP) quantization techniques for weight compression and the sparsity induction in filters. Weight quantization re-training (WQR), the log-polar, and the inverse log-polar geometric transformations are introduced to reduce quantization error. WQR is a method of retraining the CPR filter, which is presented to recover the accuracy loss. It forces the given quantization scheme by adding the quantization error in the training sample and then re-quantizes the filter to the desired quantization levels which reduce quantization noise. Further, Particle Swarm Optimization (PSO) is used to fine-tune parameters during WQR. Both geometric transforms are applied as pre-processing steps. The Po2 quantization scheme showed better performance close to the performance of full precision, while the DFP quantization showed further closeness to the Receiver Operator Characteristic of full precision for the same bit-length. Overall, spatial-trained filters showed a better compression ratio for Po2 quantization after retraining of the CPR filter. The direct quantization approach achieved a compression ratio of 8 at 4.37× speedup with no accuracy degradation. In contrast, quantization with a log-polar transform is accomplished at a compression ratio of 4 at 1.12× speedup, but, in this case, 16% accuracy of degradation is noticed. Inverse log-polar transform showed a compression ratio of 16 at 8.90× speedup and 6% accuracy degradation. All the mentioned accuracies are reported for a common database.

Keywords: correlation pattern recognition; sparsity; weight precision reduction; geometric pre-processing; dynamic-fixed-point quantization; transform; particle swarm optimization; memory minimization

check for updates

Citation: Sabir, D.; Hanif, M.A.; Hassan, A.; Rehman, S.; Shafique, M. Weight Quantization Retraining for Sparse and Compressed Spatial Domain Correlation Filters. *Electronics* **2021**, *10*, 351. https://doi.org/10.3390/electronics10030351

Academic Editor: Gwanggil Jeon
Received: 13 November 2020
Accepted: 4 January 2021
Published: 2 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The computer vision system competence has faced many challenges during their early development phases. These challenges impeded the target detection performance of artificial vision systems. The human vision can easily distinguish the object despite occlusion, clutter, rotation, lighting conditions, scale, or noise; however, the camera sensors encountered difficulty while resolving the mentioned challenges. In order to mitigate these hindrances, multiple efforts are made in the Correlation Pattern Recognition (CPR) literature. Usually, these challenges can be solved by applying an appropriate form of rota-

tion. Besides scale invariance, improving the statistical approach to training the matching filter, extraction, and exploiting the scale-invariant features for the detection or geometrical transform are other useful steps. Affixing pre-processing steps before the training and inference phase incurs an extra computation cost.

Traditionally, CPR filters are trained and tested in the frequency domain. Contrary to that, spatial domain CPR filters are trained in the frequency domain, and they are later converted back to the space domain for inference. The current paper refers to this methodology as frequency-trained (FT). In addition to this approach, this paper also considers complete training and inference in the space domain known as spatially-trained (ST). Inference in the spatial domain is computationally expensive as compared to the frequency domain. It involves cross-correlation between the test image and the reference template. Inference can be performed on various devices, like CPU, Internet-of-Things (IoT) devices, GPU, or ASICs.

1.1. Motivation and Research Challenges

Computation Cost Associated with Spatial Domain Correlation Filters: In order to handle the false detection under non-uniform lighting conditions, the state-of-the-art CPR [1,2] employed spatial filters instead of a typical approach of training and testing filters in the frequency domain. However, real-time implementation of the spatial filters demands more computational resources than frequency domain filters.

Hardware Implementation Constraints: Intrinsically, embedded systems have limited resources. So, synthesizing the state-of-the-art CPR inference on embedded systems poses many challenges. Hardware is either constrained by the number of operations that can be executed in parallel or by the memory interface transmission rate [3].

Associated Research Challenges: The problem of computation complexity and hardware constraints poses the following challenges.

- Efficiency and Computational Complexity of Inference due to Number and Large Sizes of Spatial Domain Correlation Filters: The large size of CPR-trained templates and the number of filters required for each target, especially for out-of-plan training, make the inference phase computationally complex. This complexity increases because of certain limitations and critical requirements, such as limited available power, high throughput demand, and hard real-time processing requirements; so, sparsity can reduce workload and increase inference efficiency.
- Memory Requirement of CPR filter Weights: Full-precision filter weights have higher memory requirements, which increases with the size and number of spatial filters. In that case, memory minimization is possible through filter-weight compression.

Consequently, both the above-mentioned challenges increase the number and complexity of operations required to detect the target. To address such challenges, this paper mainly focuses on compression and retraining CPR approaches. Subsequently, the following research gaps should be explored:

- We need to explore compression techniques for CPR filters and improve the inference computation efficiency; however, reducing the weight precision results in the emergence of quantization error, which degrades the classification accuracy. The real challenge is to maintain the classification accuracy for assuring the maximum possible compression ratio
- To minimize the computation workload for inference without degradation in classification accuracy.

Recent researches on CPR apply some pre-processing steps before training filters focusing on accuracy or in-variance. These steps are used to achieve zoom, rotation, or translation in-variance. Gardezi et al. [1] use Affine Scale Shift Invariant Feature Transform (ASIFT) along with a spatial correlation filter that enables the fully-invariant filter. Similarly, Awan et al. [4] devise an auto-contour-based technique to reduce the side lobes. This method assures higher accuracy through prior object segregation before correlation

with the reference template; however, the mentioned techniques do not target the filters' compression, efficiency, or memory requirements.

The flow diagram of the proposed techniques is illustrated in Figure 1. Database of training, validation, and testing samples are read to pre-process through geometric transformation (step 1 in Figure 1), which is log-polar or inverse log-polar transform. For direct quantization, training samples are passed through spatial-training (ST) (step 2 in Figure 1) or frequency-training (FT) (step 3 in Figure 1) before applying quantization schemes. Weight quantization retraining (WQR) (step 5 in Figure 1) is applied after ST. Then, the outcome is quantized (step 4 in Figure 1) for compressed filters. These quantization techniques from all methods are cross-correlated with a testing image to produce a correlation plane. This plane is used to generate the detection score. Particle Swarm Optimization (PSO) (step 6 in Figure 1) is employed to find and fine-tune the γ and β parameters. Further, Table 1 represents the details of variable and its description used in this paper.

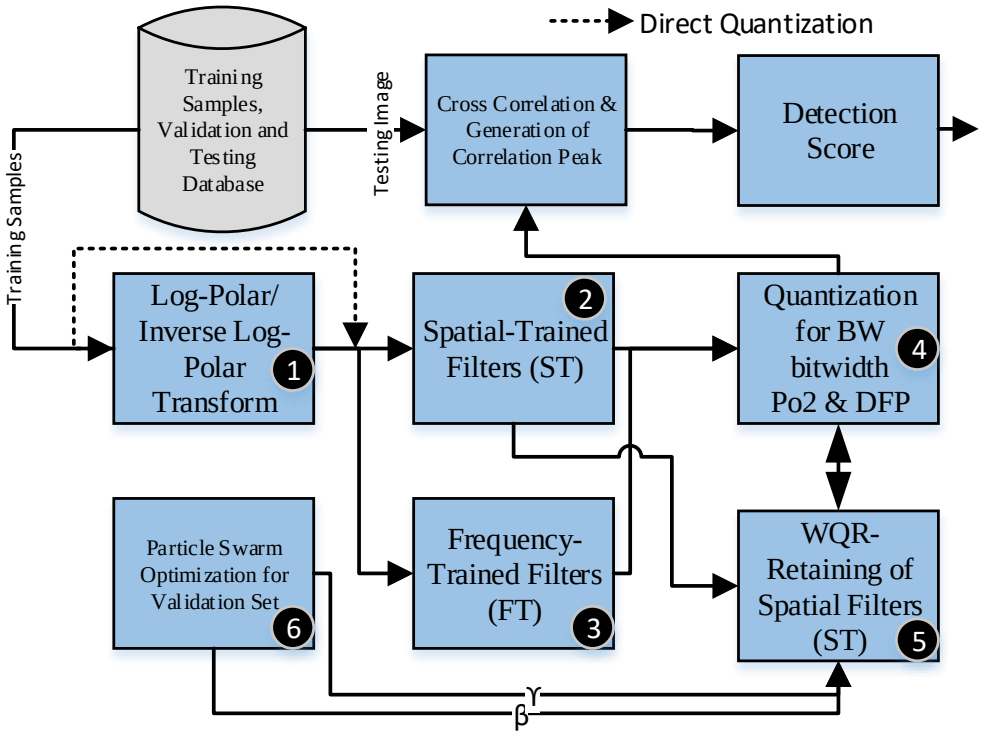


Figure 1. Optimization flow of our Correlation Pattern Recognition (*CPR*) quantization technique.

1.2. Contributions

This paper makes the following contributions:

- A Weight Quantization Retraining (WQR) (step 5 in Figure 1) method is proposed in this paper to retrain low-precision quantization weights of the *CPR* filter for dynamic fixed point and power-of-two (step 4 in Figure 1) quantization schemes. Further, the PSO (step 6 in Figure 1) technique is applied to optimize β and γ.
- Log-polar and inverse log-polar transforms (step 1 in Figure 1) are introduced as the pre-processing strategies to support the low-precision CPR filter quantization.
- An analysis is performed to compare the advantages of ST filters (step 2 in Figure 1) and FT filters (step 3 in Figure 1). This analysis is further extended to each domain, either spatially-trained or frequency-trained, to investigate the comparative benefits of power-of-two (Po2) and dynamic-fixed-point (DFP) quantization schemes.
- The overall analysis compares the advantages of direct, log-polar, inverse log-polar, and WQR, which provides a better perspective.

Table 1. Variables used in this work.

Variables	Comments
f_w	Spatial filter weights in floating-point precision
L_{pow2}	Quantized weights for Power-of-Two (Po2) scheme
L_{dfp}	Quantized weights for Dynamic-Fixed-Point (DFP) scheme
BW	Bit-width for precision reduction
m_1	Exponential power of two used for the upper-bound
m_2	Exponential power of two used for the lower-bound
v	Maximum absolute value of f_w
U_{dfp}	Quantized weights for the Dynamic-Fixed-Point scheme
$mC^x_{\beta,\gamma}$	Modified Average Image Correlation Height
$mS^x_{\beta,\gamma}$	Modified Average Image Similarity
γ	Parameter of the contribution of quantization error
β	Parameter of the contribution of average
ξ	Co-efficient of quantization error
cp_j	Raw correlation plane for test image j
ncp	Normalized correlation plane
$COPI$	Correlation output peak intensity
τ	Threshold for object detection
$\Delta\%$	Percentage difference between threshold τ and $COPI$

2. Mathematical Background and Related Work

CPR is a match-filtering [5–7] technique. Reference [8] is one of the correlation-based pattern recognition approaches. During the last three decades, CPR progressively improves the designs of statistical methods for training the reference templates. Typically, the CPR training phase involves different sample images and their processing through a statistical training method to prepare a reference template for the target/object. Target localization in the testing image uses cross-correlation with a stride of 1, which means convolution after 180-degree rotation for searching the target/object at each location, and that gives the output correlation plane. For spatial domain filtering, each output in the correlation plane has a float-point operational cost, which is equal to the product of height and width of the reference template. The presence of the target is identified by the height of the peak in the correlation plane. The relative class resemblance of the target/object is proportional to the peak height in the output correlation plane. The larger peak in the output correlation plane corresponds to a stronger probability of the target, whereas the absence of the target results in a broader peak in the correlation output plane. Each reference template design must be designed keeping in view the fact that there is a trade-off between optimal correlation peak, distortion invariance, and clutter suppression [9,10]. Generally, these traits are regulated by optimized parameters. Preliminary steps to resolve the target detection problem [5] are limited to optimal optical correlators [6,7]; however, these basic template designs do not solve the issues, such as distortion and clutter rejection. Synthetic Discriminant Filters (SDFs) [11] are used as the first decent strategy to deal with the challenges. It is the earliest effort in the overall CPR domain and besides, it provides the foundation for further advancements in this field. Although, the later generalization of SDFs involves addressing the in-variance issue in the filter design [12,13]. This approach partially handles the mentioned challenges. Besides, for achieving the optimality and invariance [13], the SDF design emphasizes on enhances the signal-to-noise ratio in the proximity of the target but it allows the side-lobes to function in the proximity of the correlation peak, which complicates the estimation method, and makes it difficult to obtain the optimal threshold value. Casasent et al. [14] generate a bigger dataset for training, which is obtained by rotating and shifting each image. After that, the Minimum Average Correlation Energy (MACE) [15,16] filter, which is a hybrid form of MACE, and a filter with the minimum variance SDF (MVSDF) [17,18] are proposed. The proposed approaches produce sharper peaks in the correlation output plan as compared to their predecessors. Both these filters (MACE, MVSDF) give excellent responses against noise; however, their

performances are inadequate against distortion. A trade-off must be kept between the parameter to control the object/target detection despite clutter and the distortion of an object/target. To answer the possible challenges, a breakthrough in the CPR field is made in the mid-90s. First, the Maximum Average Correlation Height (MACH) and then the Optimal Trade-Off MACH filter are introduced. These statistical models optimize the filter response between noise, distortion, and clutter rejection but these filters excessively depend on the mean of all the samples. This obstacle impedes the classifier's performance and results in false positives. Eigen Maximum Average Correlation Height (EMACH) [19] mitigates the dependence on the sample average, which relatively improves the classifier accuracy. Further improvement in the classification accuracy is possible in the Enhanced Eigen Maximum Average Correlation Height Filter (EEMACH) [20] along with a tradeoff. WMACH [21–23] enhances the performance of the reference template, and the Gaussian wavelet is applied before training as a pre-processing step. Target search in the input scene enables the CPR techniques to accurately localize the target. MMC filter [24] exploits this CPR feature by integrating the Support Vector Machine (SVM) with CPR to pinpoint the target's location within the input scene. The CPR localizes the target when the SVM allowed generalization in the input. Further enhancement in the CPR performance is introduced through partial-aliasing correlation filters [25]. These filtering techniques ensure sharper peaks in the presence of a target/object. Performance improvement originates from the aliasing effect that takes place because of the circular correlation as compared to linear convolution, which impedes the CPR performance. Human action recognition [26–29] intercedes the CPR filters to detect human actions.

Achuthanunni et al. [30] and Banerjee et al. [31] propose the band-pass pre-processing of Laplacian of Gaussian (LoG) of unconstrained correlation filter for facial recognition. The band-pass filtering achieves a trade-off between suppressing irrelevant details and enhancing the edges for feature representation. The proposed technique applies PSO to find the optimum scale. The filter successfully handles the challenges, like illumination and noise, during face recognition and outperforms other correlation filters. However, it lacks detection in the presence of out-of-plane and in-plane rotation and scale. Akbar et al. [32] also employ the rotational invariant correlation filter for moving human detection. The proposed methodology pre-processes the color conversion approach and background elimination to enhance correlation filters' speed and accuracy. Akbar et al. [33] propose the hardware implementation of correlation filters on FPGA, which reduces the processing time with negligible performance loss. Hardware design is implemented in LabView, which later may be used later in real-time security applications. Haris et al. [34] apply the MACH filter to localize the target in videos. The target is tracked using a particle filter, while motion is approximated using the Markov model. An approximate proximal gradient algorithm is applied to limit the object tracking to target templates. Haris et al. [35] implement the fast-tracking and recognition using Proximal Gradient Filter (PG) and modified MACH. The proposed tracking approach resolves the challenge of target detection and changing the coordinates of the target. In another research [28], a blended approach is proposed to simultaneously handle noise, clutter, and occlusion. Logarithm transform and DoG are applied as a pre-processing step to achieve this. Further, to produce sharp peaks, a minimum average correlation energy filter is adopted to recognize the target. The results show a remarkable performance of the mentioned approach as compared to other correlation filters.

Reducing the data precision is a straightforward approximation technique to reduce memory and stringent requirements of energy. It also brings some accuracy degradation; however, finding the application's resilience against error introduces due to bit-width reduction is vital for approximation. This approximation is feasible both at software and architecture levels. Venkataramani et al. [36] propose an approximation approach to mitigate the energy requirements of Neural Networks (NNs). An approximation framework is presented which employed the back-propagation to convert the standard trained NN to AxNN which is an approximated and energy-efficient version with almost the same

accuracy. This method locates the neuron, which has the least effect on accuracy then replaces that neuron with its approximated equivalent neuron. Different approximation versions with the energy-accuracy trade-off for original NN are produced by adjusting the input precision and neuron weights. Retraining is used to recover the accuracy loss generated due to approximations. Authors also proposed customized hardware that enables the flexibility of weights, topologies, and tunable approximation called neuromorphic processing engine. This engine exploits the computation and activations units to implement the AxNN and achieve precision -energy trade-off during execution.

Rubio-Gonzalez et al. [37] propose a Precimonious framework for approximating floating-point precision reduction. This approach finds low precision floating-point data type for the program's variables, which depends on given accuracy constraints. For hardware applications, FPGA implementation requires the code change, while software application only requires to use a dedicated library or modification in data type. The framework evaluates different test programs which include numerical analysis applications, Scientific Library, and Numerical Aerodynamic Simulation (NAS) parallel computing. The results demonstrate a 41% improvement in the performance for a precision reduction in data types. Pandey et al. [38] propose the fixed-point logarithm function approximation, which is implemented using FPGA. The proposed approach approximates the mathematical function by presenting a binary logarithm unit. The proposed hardware combines the fixed-point data path with a combinational logical circuit, enabling low area utilization. This approach is verified using a Xilinx Virtex-5 device. The hardware can approximate integer, a mix of integer and fraction, and fractional-only inputs. Moreover, Table 2 summarizes the comparison between significant works in CPR literature.

2.1. Optimal Trade-Off Maximum Average Height Correlation (OT-MACH) Filter

Maximum Average Height Correlation filter [39,40] is designed with a prime objective of target/object recognition, which, unlike previous methods, simultaneously handles maximum possible distortion tolerance, ability to discriminate objects, and capability of dealing with noise in the test image. The MACH filter mainly comprises of the criteria known as Average Correlation Height (ACH), Average Correlation Energy (ACE), and Output Noise Variance (ONV). For deriving the MACH, the following energy expression is used:

$$E(h) = \alpha(ONV) + \beta(ACE) + \gamma(ASM) - \delta(ACH), \tag{1}$$

$$E(h) = \alpha h^{+} C h + \beta h^{+} D_x h + \gamma h^{+} S_x h - \delta |h^{T} m_x|, \tag{2}$$

$$h = \frac{m_x^*}{\alpha C + \beta D_x + \gamma S_x}, \tag{3}$$

where α, β and γ are non-negative filter-tuning parameters, m_x is the average of the training image vector $x_1, x_2, x_3 \ldots, x_n$, and C is a diagonal-power spectral density matrix of additive input noise.

$$D_x = \frac{1}{N} \sum_{i=1}^{N} \left[X_i * X_i^{+} \right], \tag{4}$$

where X_i is a diagonal matrix of the i^{th} training image. S_x denotes the similarity matrix of the training images.

$$S_x = \frac{1}{N} \sum_{i=1}^{N} \left[(X_i - M_x) * (X_i - M_x)^{+} \right], \tag{5}$$

where M_x is mean of vectors X_i. Different values of α, β and γ are optimized to get the required response under different test-image scenarios.

Table 2. Comparative analysis of related work.

Approaches	Cross-Correlation Domain	Emphasis	Methodologies and Strengths	Limitations
MACH, OT-MACH [39,40]	Frequency domain	Automatic Target Recognition	The generalization of the minimum average correlation energy presented, which improved the target recognition in the presence of additive noise and distortions	Results in false-positives obtained because of excessive dependence on mean image and low discrimination ability
EMACH [19]	Frequency domain	Automatic Target Recognition	Two new metrics, all image correlation height and the modified average similarity measure introduced to improve false-positives and increase the discrimination ability	Poor generalization capability
EEMACH [20]	Frequency domain	Automatic Target Recognition	Based on Eigen analysis of EMACH filter which resulted in better generalization capability than the EMACH filter	Required extensive Eigen analysis is computationally expansive
Fully invariant quaternion based filter [41]	Frequency domain	Automatic Target Recognition, to achieve invariance in terms of color, scale, and orientation.	For color target recognition, logarithm mapping and EMACH combined in quaternion domain which successfully solved color, rotation, and scale distortions	Incurs an extra computational cost due to pre-processing involved
Space variant maximum average correlation height (MACH) filter [42]	Spatial domain	Automatic Target Recognition	Enables detection in an unpredictable environment which is resilient against background heat signature variance and scale changes	Incurs the additional computation cost due to spatial domain filters
Pre-processing using low-pass filtering of space-variant correlation filter [43]	Spatial domain, frequency domain pre-processing	Automatic Target Recognition reduces computation workload for target search	A low-pass filter is employed to reduces the search space for target detection	Incurs the computation complexity due to spatial domain filters and pre-processing steps
Combination of spatial correlation filters and affine scale-invariant feature transform [1]	Spatial domain, spatial domain pre-processing	Automatic Target Recognition, to achieve invariance to color, scale, and orientation	Used Affine Scale Invariant Feature Transform (ASIFT) for pre-processing to achieve translation, zoom, rotation, and two camera axis orientation invariance	Increases performance and increases the computation complexity due to ASIFT
Composite filtering [44]	Frequency domain	Automatic Target Recognition, to achieve full invariance	Resilience against distortion, for example, in-plane and out-of-plane rotation, illumination, and scale alterations which obtained after the pre-processing of the difference of Gaussian (DoG) and logarithmic on EEMACH	Requirement of computational resources increase extensively
Ours	Spatial domain	Automatic Target Recognition, to achieve sparse and compressed correlation filter representation.	Automatic Target Recognition to achieve sparse and compressed correlation filter representation.	

2.2. Eigen Maximum Average Correlation Height (EMACH) Filter

EMACH [19] is designed to improve the false-positive generated due to over-emphasis on the average training image. The improved statistical method introduced β to control the average training image's contribution in the filter design. EMACH filter is defined by criteria C_x^β and S_x^β.

$$C_x^\beta = \frac{1}{N}\sum_{i=1}^{N}(x_i - \beta m)(x_i - \beta m)^+, \tag{6}$$

$$S_x^\beta = \frac{1}{N}\sum_{i=1}^{N}[X_i - (1-\beta)M][X_i - (1-\beta)M]^+, \tag{7}$$

$$J_x^\beta = \frac{h^+ C_x^\beta h}{h^+(1+S_x^\beta)h}, \tag{8}$$

$$(1+S_x^\beta)^{-1}C_x^\beta h = \lambda h. \tag{9}$$

Eigen value λ and Eigen vector $(1+S_x^\beta)C_x^\beta$ define the filter.

2.3. Log-Polar Transform

The mammalian retina is analog to the log-polar transform which converts the standard Cartesian coordinate (x, y) into log-polar coordinates θ and ρ. The log-polar transform is used for object rotation and scale-invariance [41], where scalability and rotation translate into a peak position in the output correlation plane. Log-polar transform of the color image in RGB format is shown in Figure 2. Note that transform is applied to each channel separately.

$$z = re^{\theta+\rho}. \tag{10}$$

In log-polar domain, the same is represented as:

$$w = logr + \theta i + \rho i = u + iv + \rho i. \tag{11}$$

In the log-polar domain, θ corresponds to the rotation. In fact, rescaling an object results in a horizontal shift in the mapping, and the re-scaling effect is grasped through the following set of equations (see Figure 2):

$$z = r\gamma e^{\theta} \tag{12}$$

$$w = logr + \theta i + log\gamma = u + iv + log\gamma \tag{13}$$

where $\log\gamma$ corresponds to the horizontal shift.

This paper is organized into different sections. Each section describes a significant part of our approach in detail. Section 3 demonstrates the methodology and subdivides it into further sections. Section 3.2 describes the details of each quantization scheme. Section 3.3 explains the CPR filters' re-training using the quantization error. Details of log-polar, inverse log-polar transformation and its support for quantization are mentioned in Section 3.4. The quantization configuration settings are given in Section 3.5. Section 4 provides the details of the experimental analysis, which is further divided into experimental setup, parameter optimization, and performance analysis. In the end, Section 5 concludes the paper.

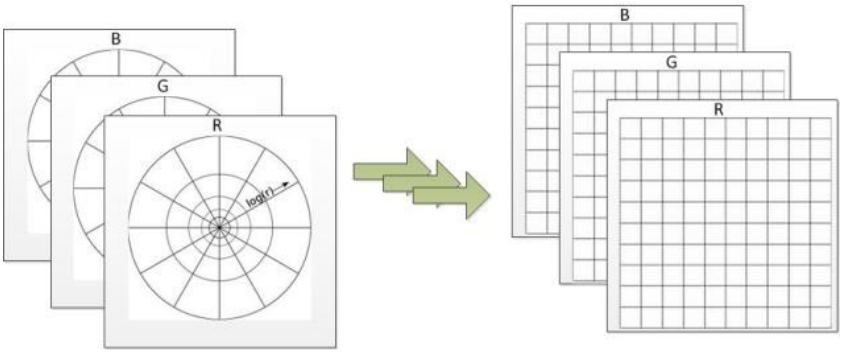


Figure 2. Log-polar illustration from certain coordinates to log-polar coordinates.

3. Methodology

3.1. Overview

The block diagram presents an integrated framework of the proposed approach in Figure 3. The framework accepts a number of training instances $x_1, x_2, x_3, ...x_N$ and a testing sample y_i as inputs. Instances carry through a log-map (step 1 in Figure 3), and inverse log-map transforms (step 2 in Figure 3). The details of these respective transforms are provided in Section 3.4. Direct quantization is applied (step 3 in Figure 3) without the transform. From this point onwards, each of these cases either branches out into FT, ST, or spatially-retrained categories. For a FT filter, frequency transform (step 4 in Figure 3) is used.

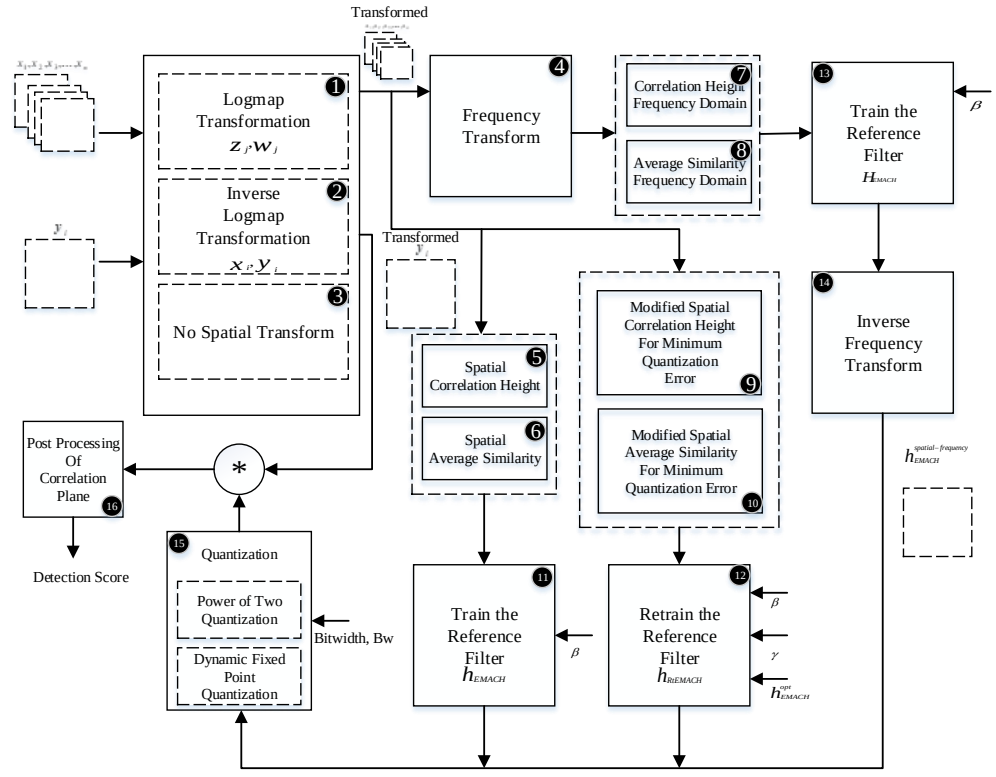


Figure 3. Complete block diagram representation of step-by-step implementation of quantization schemes for *CPR*.

After obtaining the Correlation Height (step 7 in Figure 3) and Average Similarity (step 8 in Figure 3) in the frequency domain, the regular training (step 13 in Figure 3) is performed to get the H_{EEMACH}. Spatial frequency response $h_{EEMACH}^{spatial-frequency}$ is obtained after the Inverse Fourier Transform (step 14 in Figure 3) of H_{EEMACH}. Further, the training process

is repeated in the spatial domain where Spatial Correlation Height (step 5 in Figure 3) and Spatial Average Similarity (step 6 in Figure 3) are calculated, while the training is conducted in the spatial domain. Similarly, Weighted Quantization Retraining Modified Spatial Correlation Height (step 9 in Figure 3) and Modified Average Similarity (step 10 in Figure 3) are computed for retraining (step 12 in Figure 3) of reference filter $h_{RtEEMACH}$. Retraining requires β, γ and the already optimized filter h^{opt}_{EEMACH} with floating-point precision. The details of retraining are available in Section 3.3. Consequently, the trained templates from all the quantized approaches (step 15 in Figure 3) use power-of-two (Po2) and dynamic-fixed-point (DFP) schemes. The details of these schemes are in Section 3.2. Subsequently, the correlation is calculated in a spatial domain, like a window operation, which generates the correlation output plane. Lastly, the detection score evaluation (step 16 in Figure 3) is performed by post-processing of the output correlation plane. The details of post-processing are available in Section 4.

3.2. Quantization Schemes

Quantization schemes convert the pre-trained floating-point precision weights into quantized weights with minimum distance from the original filter; however, the magnitude of distance depends on the type of quantization scheme.

Evaluation: For evaluating the quantization mechanism, two quantization schemes are chosen and evaluated for filter compression. These schemes are power-of-two (Po2) and dynamic-fixed-point (DFP) quantization. The resulting properties of these quantization schemes are studied in conjunction with direct, log-polar, inverse log-polar, and filter retraining.

Power-of-Two Quantization: Po2 is the state-of-the-art quantization technique used for data compression. Zhou et al. [45] implement the Po2 quantization for quantization of deep networks. This technique is employed due to its hardware-friendly nature, which means that multiplication can be performed using a shift operation. This property gives it an advantage over other quantization schemes during spatial cross-correlation. Po2 quantization can be defined using the following mathematical framework:

$$L_{pow2} = [\pm 2^{m_1}, ..., \pm 2^{m_2}, 0]. \tag{14}$$

m_1 and m_2 are integer numbers with

$$m_1 = floor(\log_2 1.33v), \tag{15}$$

$$v = \max(abs(f_w)). \tag{16}$$

For a given bit-width (BW), m_2 can be mathematically represented as follows:

$$m_2 = m_1 - (2^{BW-1} - 1). \tag{17}$$

Equation (14) presents the proposed quantization levels for different values of m_1 and m_2; however, as mentioned in Equations (15) and (16), these values further depend on the absolute maximum value of weights in the filter f_w. Overall, the quantization levels depend on the distribution and maximum absolute value of weights in the filter. Since the quantization scheme's nature is symmetric, we utilize only $2^{BW} - 1$ of 2^{BW} quantization levels. Additionally, we have added an extra quantization level at zero value, as shown in Equation (14). Therefore, the filter may show high accuracy. CPR filters employ the training samples containing black background, adding the quantization level at zero increases the sparsity significantly in trained filters.

Dynamic-fixed-point Quantization: Since there are plenty of cases in which this scheme is successfully implemented to achieve relatively better compression versus accuracy trade-off [46,47], however, unlike Po2 quantization, DFP quantization has a better peak-signal-to-noise ratio (PSNR). This property provides it an edge over Po2. Overall, this scheme assures less noise because it produces equidistant points for quantization levels

as compared to the previous quantization method. Equations (18) and (19) represent the quantization scheme. For bit-width (BW), Equation (18) maintains quantization levels at equal distance from each other.

$$L_{dfp} = [\pm 2^{BW-1} - 1, \pm 2^{BW-1} - 2, ..., 0], \tag{18}$$

$$U_{dfp} = \frac{L_{dfp}}{2^{BW-1}} \times 2^{m_1}. \tag{19}$$

Like Po2 quantization, this approach is symmetric, while Equation (18) provides the normalizing and scaling functions to the already-bounded quantization levels in Equation (19).

The weight distribution of both quantization schemes before and after quantization is presented in Figure 4. By comparing Figure 4a,b, the Po2 has non-uniform quantization levels as compared to DFP, whereas more weights are mainly quantized around zeroth quantization level, which preserves the low-value weights that help to improve the accuracy. Meanwhile, the DFP induces more sparsity as compared to Po2 quantization. This sparsity increases with a rise in compression ratio as more zeroth levels are added with an increase in compression ratio.

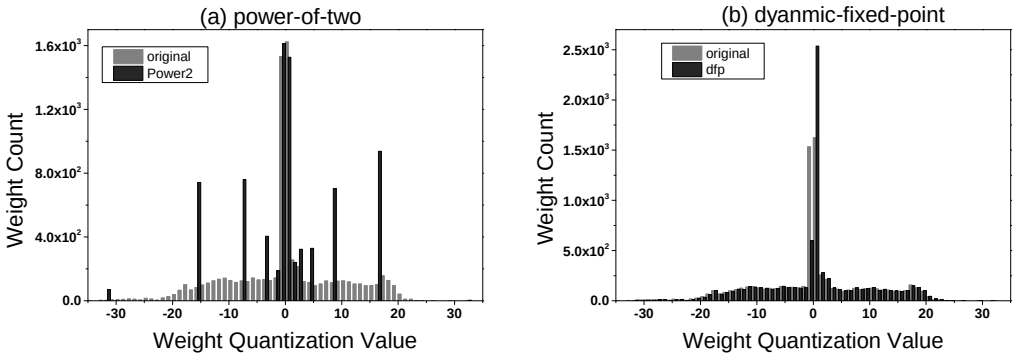


Figure 4. Distribution of filter weights prior and after direct quantization: (**a**) power-of-two (Po2); (**b**) dynamic-fixed-point (DFP).

To establish a proper connection between the quantization schemes and the resulting quantization noise, Figures 5 and 6 are provided for analysis. For DFP quantization, Figure 5 illustrates the relationship between peak signal-to-quantization noise ratio (PSNR) and compression ratio. A sample is taken from the Fashion MNIST dataset, and we approximate it using DFP quantization. By applying direct DFP to the filter, the peak signal-to-noise ratio remains constant up to 6-bit compression, but, after this, peak-signal-to-noise-ratio starts to decrease because the quantization interval doubles with the reduction of a bit, which doubles the compression ratio. Similarly, DFP quantization has better PSNR after log-polar transform, while both quantization schemes have the same PSNR values for 2-bit (CR = 8) to 1-bit compression (CR = 16). Likewise, in Figure 6, applying Po2 quantization on a sample and monitoring PSNR values for each compression level yields a constant PSNR value, and this goes on up to 3-bit compression. From that point, as the compression ratio increases, the new quantization levels are added near the zeroth level. This causes more quantization noise in a compressed version. For direct Po2 quantization, it gradually falls to 2-bit (CR = 8) and 1-bit (CR = 16) compression. Overall, a better PSNR value is achieved through this method than the direct Po2 quantization, and besides, that behavior reverses after 3-bit compression. PSNR for 2-bit (CR = 8) and 1-bit (CR = 16) compression falls more rapidly than the direct Po2 quantization. By comparing the two different quantization approaches, it is evident that DFP quantization has achieved better PSNR values as compared to Po2 quantization. Contrary to that, PSNR values of Po2 remain insensitive to most compression values. Moreover, in the case of DFP quantization,

PSNR value for 1-bit compression (CR = 16) drops to -40; however, in the case of Po2 quantization, PSNR drops to -32 in case of 1-bit (CR = 16) compression; therefore, it is quite clear from both Figure 5 and 6 that Po2 quantization drops less in case of PSNR for lower bit compression as compared to DFP quantization.

These observations hint at the superior accuracy of DFP quantization as compared to Po2 up to 5-bit compression. After that, accuracy of both should be equal for 4-bit compression, while, for 3-bit and more, Po2 quantization should have greater accuracy.

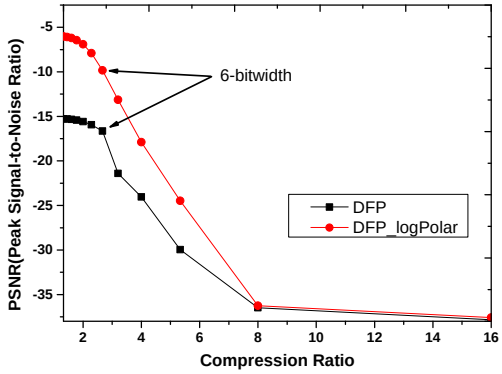


Figure 5. Peak signal-to-noise ratio (PSNR) of baseline DFP quantization; DFP quantization after log-polar transform for different compression ratios (CR).

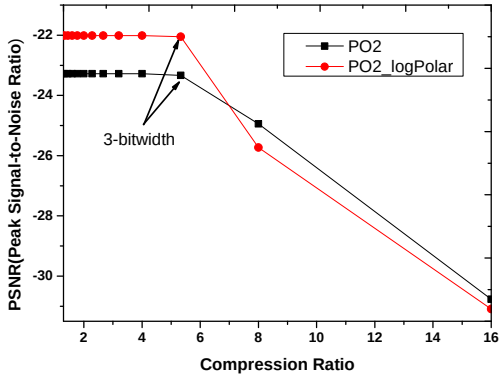


Figure 6. Peak signal-to-noise ratio (PSNR) of baseline Po2 quantization; Po2 quantization after log-polar transform for different compression ratios (CR).

3.3. Retraining the CPR Filter

Quantization error causes inaccurately trained filters. The fine-tuning of the quantized trained filter is an approach to get more accurate filters, which can be possible through a retraining filter. A mathematical framework is proposed for retraining filter to add the quantization error term in an already defined statistical training method. Equation (20) adds the given quantization error for trained filter h_{eq} with each sample x_i. In Section 3.2, we have already seen the PSNR degradation with the increase in the compression ratio. This PSNR degradation is different for the Po2 and DFP quantization schemes due to quantization noise. In order to ensure the compensation because of introducing these quantization approaches, quantization error co-efficient ξ is used to control the contribution of quantization error (h_{eq}) in filter design, where ($0 < \xi < 1$). Modified Average Image Correlation Height (mAICH) criteria has an additional term h_{eq} for each sample x_i because it adds to each sample, as well as in the average of samples, m. Equation (22) presents the m$AICH$ after substitution of v_i and m_h.

$$mAICH = \frac{1}{N}\sum_{i=1}^{N}(h^{+}(x_i + \xi h_{eq}))^2 - \alpha(h^{+}(m + \xi h_{eq}))^2, \tag{20}$$

where h^{+} is the complex conjugate transpose of h.

$$v_i = x_i + \xi h_{eq}, m_h = m + \xi h_{eq}, \tag{21}$$

$$mAICH = \frac{1}{N}\sum_{i=1}^{N}(h^{+}v_i)^2 - \alpha(h^{+}m_h)^2, \tag{22}$$

$$mC_{\beta}^{x} = \frac{1}{N}\sum_{i=1}^{N}[v_i - \beta m_h]^{+}[v_i - \beta m_h], \tag{23}$$

whereas β denotes the contribution of m_h in Equation (23). In Equation (23), mC_{β}^{x} is the average of correlation peak intensities $(v_i - \beta m_h)$ samples. Ideally, all training images should follow this convention in which v_i is subtracted from a partial average of training samples. To achieve this, every sample v_i should have an identical output correlation plane, like the ideal output correlation plane f. To find out the f that suits all samples' correlation output planes, the minimum deviation is required between its correlation planes. Equation (24) describes this deviation as the average square error (ASE).

The average square error between f and g_i is given in Equation (24):

$$ASE = \frac{1}{N}\sum_{i=1}^{N}(g_i - f)^{+}(g_i - f), \tag{24}$$

where

$$g = (1 - \beta)M_h h^{*}, \tag{25}$$

where h^{*} is the complex conjugate of h. To achieve the maximum peak, partial derivative with respect to f should be equal to zero, as given in Equation (26):

$$\frac{\partial_{ASE}}{\partial f} = 0 \tag{26}$$

In this equation, f_{opt} is the optimized filter, and, after solving Equation (26) and substituting g_i in Equation (27), we get the following:

$$f_{opt} = \frac{1}{N}\sum_{i=1}^{N} g_i = (1 - \beta)M_h h^{*} = (1 - \beta)(M + \xi H_{eq})h^{*}, \tag{27}$$

where $M_h = M + \xi H_{eq}$. H_{eq} is a diagonal matrix having h_{eq} along its main diagonal, and M is a diagonal matrix having m along its diagonal. Substituting Equation (21) into Equation (23), we get:

$$mC_{\beta,\gamma}^{x} = \frac{1}{N}\sum_{i=1}^{N}[x_i - \beta m + \gamma h_{eq}]^{+}[x_i - \beta m + \gamma h_{eq}], \tag{28}$$

$$\gamma = \xi(1 - \beta). \tag{29}$$

The next step is to change the Average Similarity Measure (ASM), which defines the dissimilarity of training samples to $(1 - \beta)M_h h^{*}$, the measure referred as modified ASM, or mASM.

$$
\begin{aligned}
mASM &= \frac{1}{N}\sum_{i=1}^{N}[V_i h^* - (1-\beta)M_h h^*]^{+}[V_i h^* - (1-\beta)M_h h^*] \\
&= h^T[\frac{1}{N}\sum_{i=1}^{N}[X_i - (1-\beta)M + \beta\xi H_{eq}]^*[X_i - (1-\beta)M + \beta\xi H_{eq}]]h^{*'} \\
&= h^T mS^x_{\beta,\gamma} h^* = h^{+} mS^x_{\beta,\gamma} h
\end{aligned}
$$

where $V_i = X_i + \xi H_{eq}$. X_i is a diagonal matrix having vector x_i along its main diagonal.

Similarly,

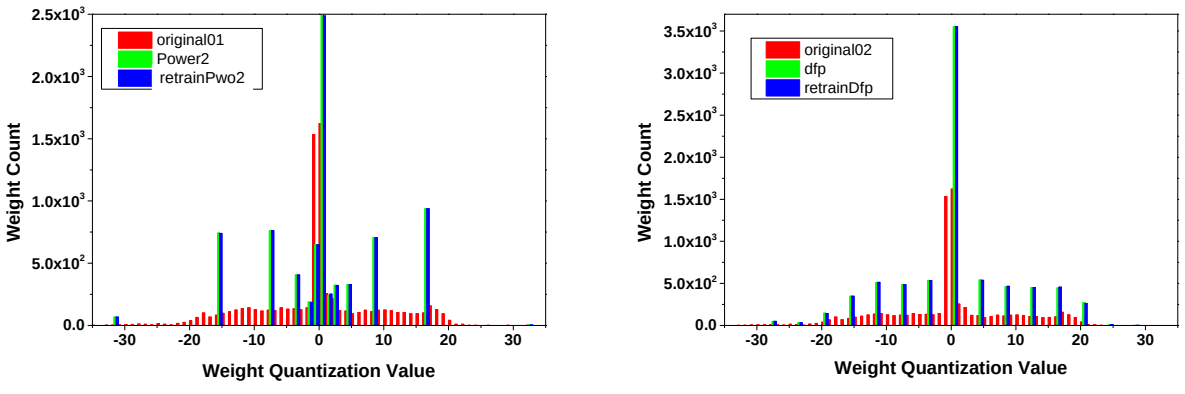
$$
mS^x_{\beta,\gamma} = \frac{1}{N}\sum_{i=1}^{N}[X_i - (1-\beta)M + \beta\xi H_{eq}]^*[X_i - (1-\beta)M + \beta\xi H_{eq}], \tag{30}
$$

where $mS^x_{\beta,\gamma}$ is a diagonal matrix.

$$
mJ^{\beta}_{x} = \frac{h^{+} mC^{\beta}_{x} h}{h^{+}(1 + mS^{\beta}_{x})h}, \tag{31}
$$

$$
(1 + mS^{\beta}_{x})^{-1} mC^{\beta}_{x} h = \lambda h. \tag{32}
$$

Eigen value λ and Eigen vector $(1 + mS^{\beta}_{x})mC^{\beta}_{x}$ define the filter. In Equations (28) and (30), $mS^x_{\beta,\gamma}$ and $mC^x_{\beta,\gamma}$ are the modified forms of $S^x_{\beta,\gamma}$ and $C^x_{\beta,\gamma}$. Figure 7 shows the floating-point filter's histogram, quantized version, and the retrained quantize version. The illustration clearly demonstrates the displacement of weight values of the retrained filter. Noticeably, the intensity values of the filter shift to adjust to new values. The retraining process changes the value of retraining intensities. The complete retraining process for a 3-by-3 snip of the filter is shown in Figure 8a. The floating-point precision filter, h_f transforms into a quantized version, h_q. Further, the quantization error, h_{eq} is calculated to support the retraining process. This process yields h_{rt} filter using Equations (31) and (32), which is a retrained version of the filter in floating-point precision. Finally, its quantized version, h_{rtq} has a reduced quantization error h_{eq}. Figure 8a demonstrates the function of retraining approach as quantization error h_{eq} in case of retaining method (see the first row in Figure 8a) is less than the direct quantization (see the second row in Figure 8a). Note that the retrained filter in floating-point precision, h_{rt} alters its weights to reduce the quantization error, h_{eq}. Figure 8b illustrates a part of the filter before and after the retraining process. All the weights of the filter do not change the value because the alteration is only limited to certain intensities. The above observations confirm that WQR reduces the quantization error h_{eq}. We expect that WQR will reduce the accuracy degradation in trained CPR filters due to the quantization process.



(**a**) (**b**)

Figure 7. Comparison of weight distribution of filter between full precision, direct, and retrain quantization. (**a**) The comparison case for Po2. (**b**) The comparison case for DFP.

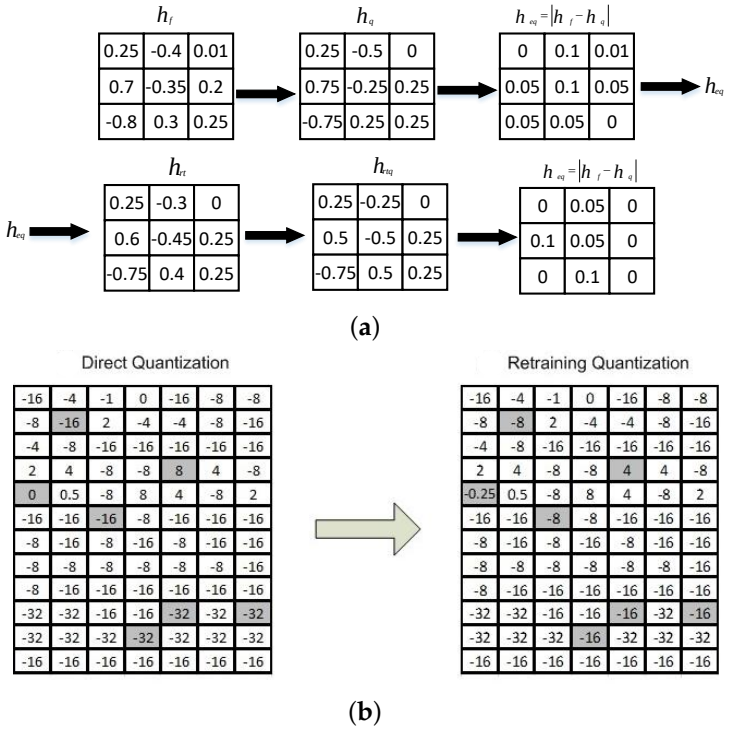


Figure 8. Retraining of a spatially-trained CPR filter. (**a**) Retraining process converts the floating-point precision weights filter, h_f into a retrain quantized version h_{rtq}. The first row describes the direct quantization and quantization error (h_{eq} calculation). The second row represents the retraining filter h_{rt} and retrained quantized version h_{rtq} when $\xi = 1$. (**b**) Snips of the filter after direct and retraining quantization. Highlighted weights are change before and after retraining.

3.4. Geometric Transform

Quantizing the magnitude of 2-dimensional filters introduces the quantization error, which degrades the quality and causes accuracy loss during the inference process. PSNR measures the ratio between the maximum possible signal power and noise power. This ratio estimates the quality after quantization. Equation (33) represents the *PSNR*, while the power of noise in the denominator is defined by the Mean Square Error *(MSE)*, which is the average of the square of the pixel-by-pixel difference between the original image and the approximated version of the image; however, the *MSE* also depends on the variance of the original and quantized signals. Equation (34) establishes a relationship between *MSE* and variance of the signal. For a higher number of pixels, the variance of both original and estimated images has more contribution than the equation's last three terms. The equation implies that there is a higher variance value of the original signal and its quantized version results in more error. It is obvious in mathematical proof presented in the Appendix A.

$$PSNR = 20 \log_{10} \frac{MAX_f}{\sqrt{MSE}}, \tag{33}$$

whereas MAX_f is the maximum possible value for a given bit-width.

$$MSE = \sigma^2_{Y_i} + \sigma^2_{\hat{Y}_i} - \frac{2}{N} \sum_i^N Y_i \hat{Y}_i + \frac{1}{N^2} (\sum_i^N Y_i)^2 + \frac{1}{N^2} (\sum_i^N \hat{Y}_i)^2, \tag{34}$$

whereas Y_i and $\hat{Y}_i$ denote the original image and the estimated image, respectively. $\sigma^2_{Y_i}$ and $\sigma^2_{\hat{Y}_i}$ are the variances of the original image and the estimated image, respectively. N denotes the total number of pixels in the original image.

To enhance PSNR for a given compression ratio, minimizing the variance of the signal and its approximated version require some transformation of the original signal. Here,

we have introduced two types of geometric transforms to reduce the variance in the next two subsections.

3.4.1. Reducing the Standard Deviation Using Log-Polar Transform

Sabir et al. [41] have already demonstrated in a previous study that applying the log-polar transform has negligible influence on the classification accuracy degradation. This paper demonstrates the additional property of log-polar besides achieving the scale and rotation invariance. This transform alters the distribution of intensity levels, resulting in reduced standard deviation in the transformed image compared to the original image. This property may be useful for the quantization of the filter weights. A sample of a shirt is selected in grayscale format to understand the effect of log-polar transform outcomes on the intensity value distribution. Figure 9a is the picture of a shirt with strips of various intensity levels. Figure 9c is a histogram of the image, which shows that a large portion of the image has zero intensity level, while 100th intensity level has the second-largest occurrence. By analyzing the overall distribution, the estimated standard deviation is 63.91. Figure 9b represents the picture's log-polar transform, which is a distorted form of an image; however, it changes the intensity distribution of the image. By observing the histogram in Figure 9d, it is evident that it reduces the frequency of black from 1200 to just 200, while the occurrence of 100th intensity level varies from less than 200 to ∼280, it alters the histogram distribution of the image of the shirt. When the log-polar transform is used, the standard deviation of the image falls from 63.91 to 45.62, which shows that now, the frequency of intensity level is in a more compact form than before; therefore, it became more efficient and convenient to apply any quantization scheme to represent the intensity levels because it will reduce the *PSNR* value. For log-polar quantization, higher PSNR values as compared to direct quantization confirm the better resilience of this method as shown in Figures 5 and 6.

3.4.2. Reducing the Standard Deviation Using Inverse Log-Polar Transform

One of the many properties of log-polar is its reversibility, which means that it is possible to convert an image back to its original for using a 2-dimensional inverse log-polar transform. Figure 10b shows a transformed image, but, unlike the previous transform, the resulting transformed object in the image reduced in size and quality because many horizontal features of the image are almost curbed. The inverse log-polar transform demonstrates in Figure 10d, and the standard deviation of the image is further reduced to 42.48, but, in this process, zero intensity increased to 2300, which is almost double as compared to the intensity of the log-polar transform.

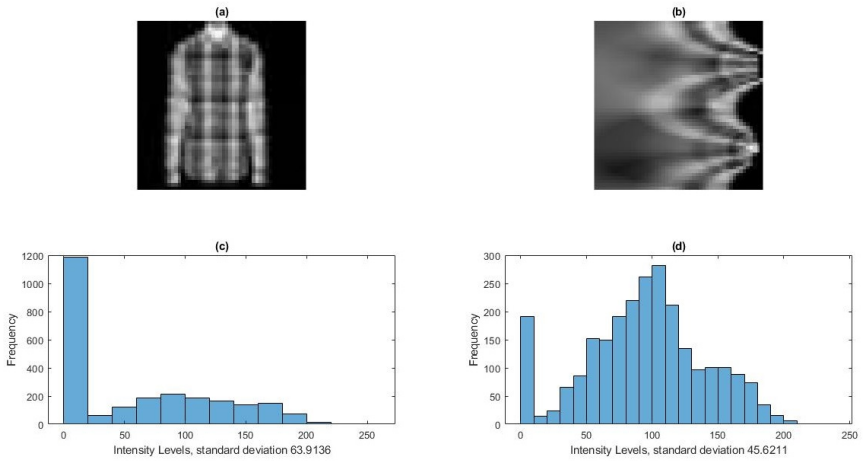


Figure 9. (**a**) Represents the picture of full shirt sample from the Fashion MNIST dataset. (**b**) Two-dimension log-polar transform of an image sample. (**c**) Histogram of image. (**d**) Histogram of log-polar transform of the image.

Equations (35) and (36) show the conversion of θ into x and y Cartesian coordinates; however, when θ varies across its range (0 to 2π), the Cartesian coordinates x and y range is 0–r. Figure 10b represents the evidence that the frequency of most intensity levels beyond zero is modified and reduced to the minimum level.

$$x = e^{\rho} \cos\theta, \tag{35}$$

$$y = e^{\rho} \sin\theta, \tag{36}$$

whereas ρ denotes the logarithm of the distance between the given point and the origin, and θ denotes the angle between the x-axis and the line through the origin and the given point. Based on the mentioned observations, we can expect that applying the log-map and inverse log-map pre-processing will reduce the quantization noise, which indirectly increases the compression ratio of spatial CPR filters.

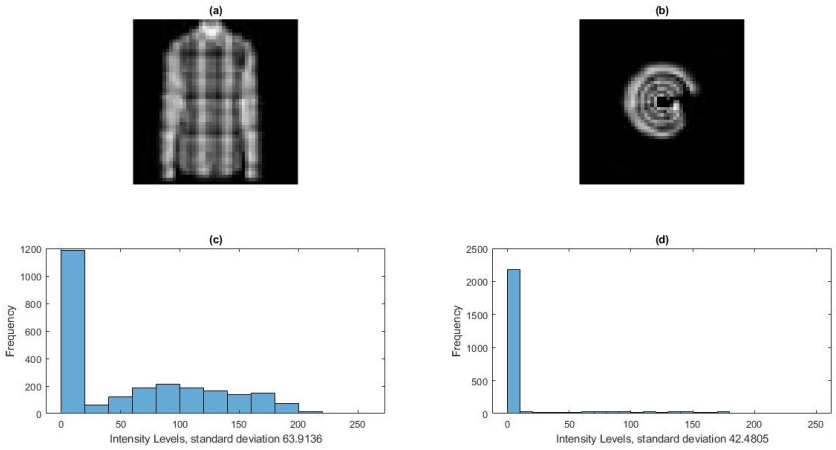


Figure 10. (**a**) Represents the picture of full shirt sample from Fashion MNIST dataset. (**b**) Two-dimensional inverse log-polar transform of an image sample. (**c**) Histogram of image. (**d**) Histogram of inverse log-polar transform of image.

3.5. Configurations for Weight Quantization

To understand the quantization and re-quantization, it is necessary to first understand different quantization methods and their configurations with or without the transform. Figure 11a illustrates the direct quantization method, through which regular training of the filter h is followed by quantization, which implies that either DFP or Po2 is performed for a given bit-width. Figure 11b represents the retraining method, through which intensity levels are reinforced using the retraining process. First, like direct quantization, h_q filter is obtained after regular filter training. Then, using h_q, a separate retraining process for each quantization approach (DFP and Po2) is employed. Finally, after the re-quantization process, h_{qrt} is achieved, which is a quantized form of the retrained filter. In Figure 11c, transforms are applied to support the quantization process. The resulting filter h is transformed using a log-polar or an inverse log-polar transform. For a given bit-width, each quantization technique is performed to obtain a different filter of h_q^l.

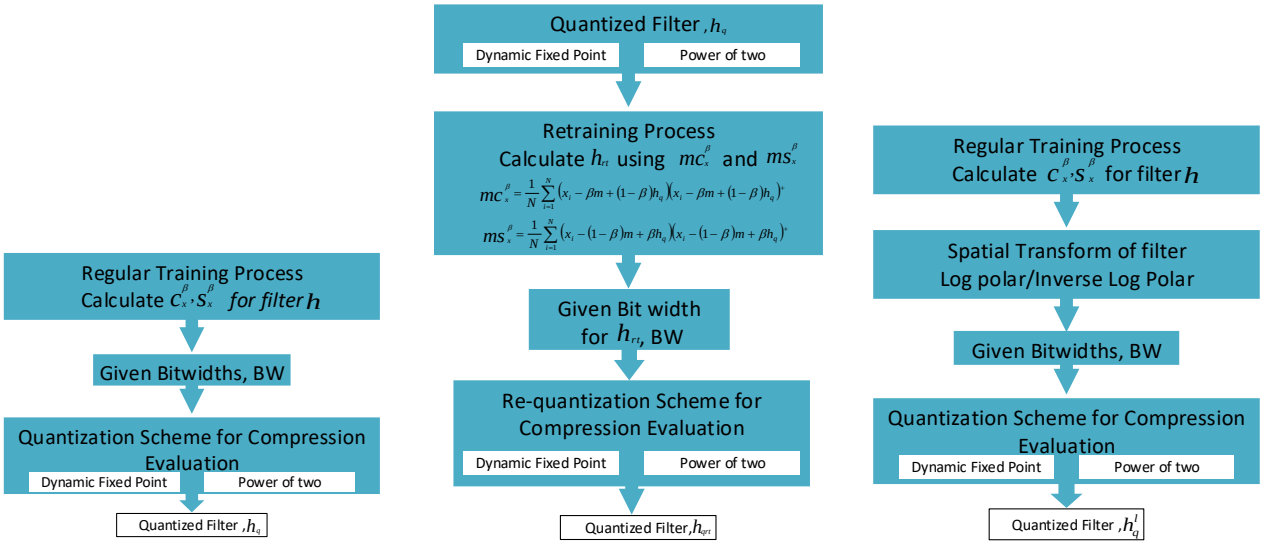


Figure 11. Flow diagram of (**a**) Direct Quantization using DFP and Po2 quantization schemes. Filters have regular training before quantization for a given bit-width. (**b**) Retraining Quantization using DFP and Po2 quantization schemes. Quantized version of regularly trained filters retrain for the corresponding bit-width, then re-quantize to obtain Weight quantization re-training (WQR) filters. (**c**) Pre-processing Quantization using DFP and Po2 quantization schemes. After regular training, a spatial transform is applied. Then, quantization is performed.

4. Experimental Analysis

4.1. Experimental Setup

Figure 12 illustrates the overall experimental setup consisting of different components.

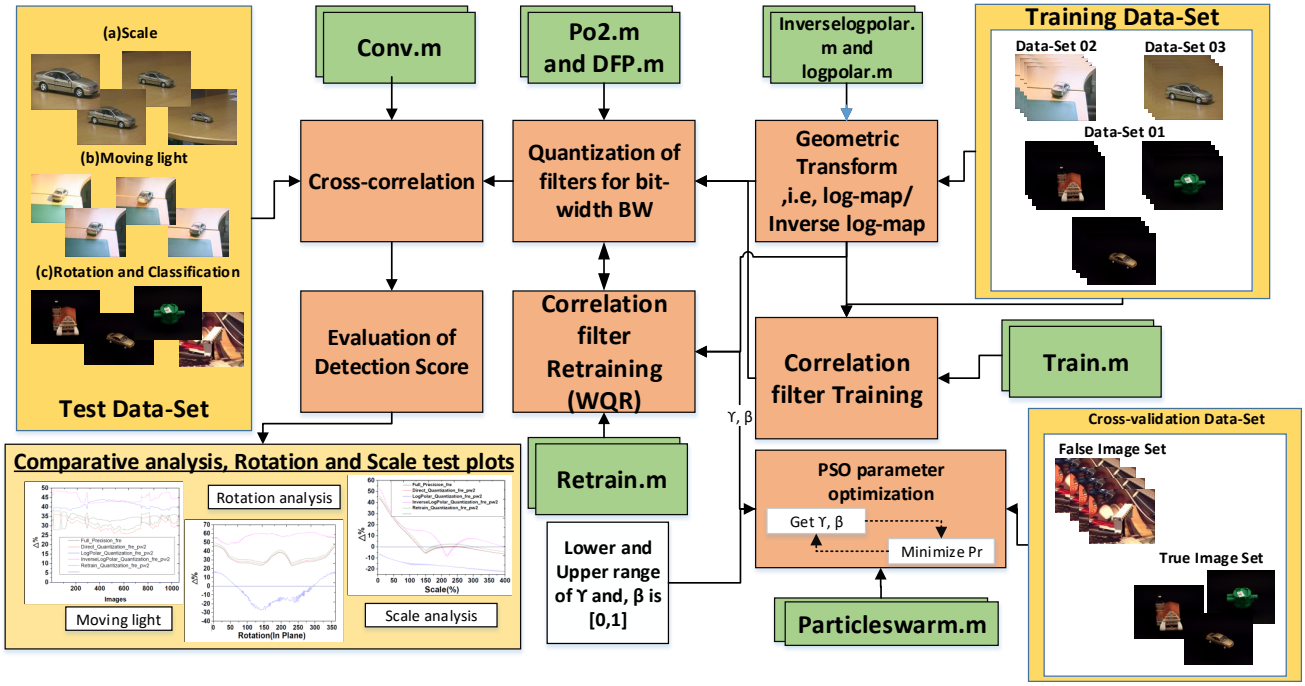


Figure 12. Experimental setup showing different software components.

4.1.1. CPR Filter Implementations and Setting

EEMACH [20] and its derivatives [44] showed a remarkable performance as compared to other CPR filters. Literature shows their superior clutter-rejection capability as compared to other methods, and experiments, which have shown better results, are conducted taking $N_v = 1$ and not at other value of N_v. The same setting of N_v. is applied to our experiments.

We applied a couple of mathematical quantization techniques to filters to reiterate the proposed approaches, as demonstrated in block diagrams Figure 11a–c. The experimental setup in Figure 12 uses in-house MATLAB scripts and functions for training (Train.m), cross-correlation (conv.m), quantizaion (Po2.m and DFT.m), WQR (Retraining.m) and geometric transform (logpolar.m and inverselogpolar.m). The experiment setup trains the filters on dataset 01, 02, and 03, with each dataset has unique purposes. For validation purposes, the parameters β and γ are obtained using PSO optimization. The in-built MATLAB function (Particleswarm.m) is employed for the PSO optimization of parameters where cross-validation dataset is used explicitly for said purpose. Initially, the filters train (Train.m) with a geometric transform (logpolar.m or inverselogpolar.m) or directly. Optimal values of β and γ parameters are generated by the PSO optimization which are employed for retraining (Retrain.m) the CPR filter. Either of these trained filters is quantized for DFP (DFT.m) or Po2 (Po2) scheme. Then, either scale, moving lighting, rotation, or classification test is performed using cross-correlation (conv.m). After this, evaluation and detection score produces the analysis graphs separately for each test.

4.1.2. Database

For evaluation, the experimental work is carried out on publicly available datasets [48]. These datasets contain test images with or without a (black) background in different poses, which vary from 0 to 180 degrees out of a plane at different elevation angles. For the training phase, we use images without a background. These training snips are centered in the middle of the test image, which makes them ideal for recognizing correlation patterns. In order to analyze the responses of precision reduction in filters, dataset 01 is specifically used to evaluate the ROC's comparison of different techniques and methods adopted in this paper. Similarly, to study the precision reduction responses against the scale enhancement and lighting alterations, dataset 02 and dataset 03 are employed, respectively.

4.1.3. Evaluation Framework

To understand the efficiency of the proposed techniques, we initially outlined an appropriate framework for experimental evaluation. After choosing the database, the next step is to define the performance evaluation framework. Instead of performing a lexicographical scan, equal window size for both the filter and the full-test image is considered. The block diagram of this framework is shown in Figure 3. Three different objects are demonstrated in Figure 13. At a 30-degree elevation angle, the images of each object are divided into six sections. Each section has six object images with six consecutive out-of-plane angles. These image sections have successive intervals 0–30, 35–55, 60–90, 95–125, 130–160, and 165–180 degrees with 5-degree incremental gap in each section.

(**a**) Volvo C70 Gold (**b**) House (**c**) Cable Box Green

Figure 13. Different training objects are illustrated. These objects have different object complexities.

For testing, we use an image at a 50-degree elevation angle. An example is demonstrated in Figure 14. Thus, a total of 18 filters are formed, while there are six filters for each object. To assess the clutter rejection capability of each filter, we draw 2560 clutter images from the database. The filter response of each filter has a maximum value called correlation-output peak intensity. Instead of directly considering the raw correla-

tion plane *cp* and the peak value for measuring the target object's existence, we consider a mathematically-derived form of correlation plane output intensity, as is presented in Equations (37) and (38). This mathematical transform assures the output quality of the correlation plane. Raw correlation plane response does not provide the quality of correlation. Only considering the maximum value in the output correlation plane provides no information about the suppression of side lobes in a correlation plane; therefore, during the correlation process, there is a high probability that the final response has high-correlation output peak intensity, but, after mathematical processing, correlation peak can reduce. Conversely, after the mathematical process, lesser peak intensity is observed in the correlation plane with shorter side lobes' intensity, which might increase.

$$\vartheta_j = \frac{cp_j - \mu_{cp_j}}{\sqrt{\sum_x \sum_y |cp_j|^2}}, \tag{37}$$

$$ncp_j = \frac{\vartheta_j}{\sigma_{\vartheta_j}}. \tag{38}$$

Here, cp_j is the raw correlation plane of the test image j, σ_{ϑ_j} is the standard deviation of the mean subtracted normalized correlation plane ϑ_j, and ncp is the normalized correlation plane. The correlation output peak intensity of this plane serves as an object-detection score. Table 3 represents sample COPI's and its corresponding detection scores for both quantization schemes.

$$\iota = \max(ncp_j), \tag{39}$$

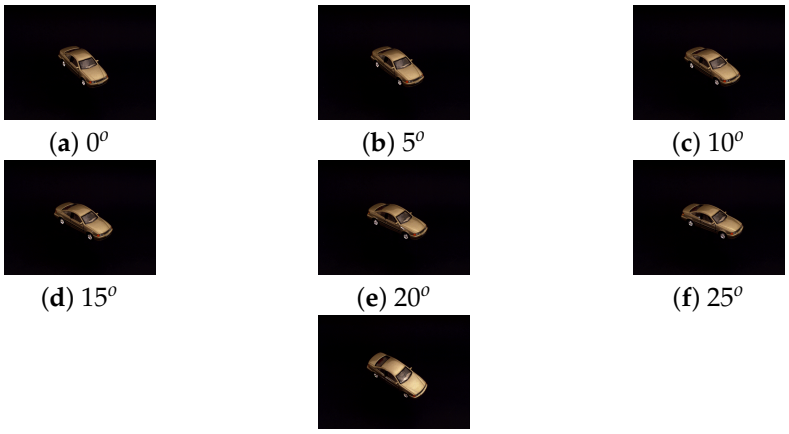
$$\Delta\% = \frac{(\iota - \tau)}{\iota} \times 100, \tag{40}$$

$$\tau = 0.5 \times \frac{1}{N} \sum_{i}^{N} ncp_i. \tag{41}$$

Table 3. The correlation output peak intensity (COPI) of correlated surface and its corresponding detection score of sample compressed correlation filters.

Bit-Width	**Frequency**				**Spatial**			
	Po2		**DFP**		**Po2**		**DFP**	
	COPI	**Score**	**COPI**	**Score**	**COPI**	**Score**	**COPI**	**Score**
1	1.42×10^{11}	3.5128	1.42×10^{11}	3.5128	3.15×10^{17}	3.5548	1.38×10^{15}	3.9381
2	2,439,063	4.3748	2,439,063	4.3748	7.36×10^{15}	3.4448	3.53×10^{15}	3.6044
3	1,557,504	3.8072	1,783,727	4.1245	3.85×10^{10}	3.8715	3.17×10^{10}	3.7645
4	1,532,146	3.7792	1,986,866	3.965	3.69×10^{10}	3.785	7.92×10^{16}	3.4876
5	1,532,708	3.7801	1,901,382	3.9308	3.72×10^{10}	3.7715	3.24×10^{15}	3.4693
6	1,532,707	3.7801	1,895,484	3.9069	3.72×10^{10}	3.7714	2.89×10^{17}	3.4648
7	1,532,707	3.7801	1,898,087	3.9115	3.72×10^{10}	3.7714	2.26×10^{16}	3.4632
8	1,532,707	3.7801	1,903,349	3.9076	3.72×10^{10}	3.7714	2.89×10^{17}	3.4611
9	1,532,707	3.7801	1,903,271	3.9042	3.72×10^{10}	3.7714	7.98×10^{16}	3.461
10	1532707	3.7801	1,903,113	3.9072	3.72×10^{10}	3.7714	7.98×10^{16}	3.4607
11	1,532,707	3.7801	1,902,467	3.9057	3.72×10^{10}	3.7714	7.98×10^{16}	3.461
12	1,532,707	3.7801	1,902,701	3.9063	3.72×10^{10}	3.7714	2.89×10^{17}	3.4609
13	1,532,707	3.7801	1,902,787	3.9058	3.72×10^{10}	3.7714	2.89×10^{17}	3.4609
14	1,532,707	3.7801	1,902,709	3.9058	3.72×10^{10}	3.7714	2.89×10^{17}	3.4609
15	1,532,707	3.7801	1,902,748	3.9058	3.72×10^{10}	3.7714	2.89×10^{17}	3.4609
16	1,532,707	3.7801	1,902,741	3.9059	3.72×10^{10}	3.7714	2.89×10^{17}	3.4609

Note: Detection scores represent very small values as compared to corresponding *COPI*.



(**a**) 0^o (**b**) 5^o (**c**) 10^o

(**d**) 15^o (**e**) 20^o (**f**) 25^o

(**g**) Test image,elevation angle (50^o)

Figure 14. Training image (**a**–**f**). Testing image (**g**).

Here, ncp_j is the correlation response of the test image j. ι is the absolute peak correlation intensity used in Equation (39). In Equation (40), $\Delta\%$ is the percentage difference between the threshold and the $COPI$ of test response j. The average normalized correlation peaks' response τ of training instances $i = 1, 2, 3, ..., N$ multiplied with a factor of 0.5 in Equation (41).

4.2. Parameter Optimization

To maximize the filter response, we should select appropriate parameters for each filter. For this purpose, we establish a framework of cross-validation set for each filter. This validation set has a test image from the training set of the corresponding filter as a true class and 100 clutter images are randomly chosen out of 2560 clutter images as a false class. The cross-validation of 600 clutter images is defined for each object. Previously, to estimate the binary class difference, Peak-to-Side-lobe Ratio and Fisher Ratio were used; however, in this paper, we employ a simple ratio of mean correlation output peak intensity of the false class μ_{ncp_F} to mean correlation output peak intensity of the true class, μ_{ncp_T}. This peak ratio is illustrated in Equation (42).

$$P_r = \frac{\mu_{ncp_F}}{\mu_{ncp_T}}. \tag{42}$$

We select an optimal β value with minimum ratio P_r. We search the optimal β value across the beta range 0–1 using PSO for each compression ratio. As quantization process for each filter results in the quantization error, which is different for direct quantization, log-polar, inverse log-polar, and filter retraining methods; therefore, each compression ratio holds a different set of optimal parameters.

PSO-Based Optimization of γ and β

The classical PSO is a self-organizing approach that holds the powerful property of dynamic non-linearity. Our problem is non-linear, we want to calculate the parameter(s) using the minimum objective function value. This makes PSO an ideal solution for parameter searching, like previous literature [36]. This property ensures the trade-off between the positive and negative response. The positive response supports constructing the swarm structures, while the negative response acts as a counterweight to this construction. Overall, this method provides a stable and complete solution to a non-linear problem. PSO also offers a balance between the exploitation and exploration of a solution. Further, potential solutions are known as particles, which excessively interact with neighboring particles. This interaction spread the updated information through-out the swarm. Filter retraining method search space is not limited to a parameter. We employed the classical PSO technique for finding the optimal values of β and γ, while minimizing the objective function of

the peak ratio P_r, where $v_{i,j} = [v_{1,j}, v_{2,j}, ...v_{20,j}]$ is the velocity vector of twenty particles and $p_i = [\gamma_i, \beta_i]$, $\beta \in [0,1]$, and $\gamma \in [0,1]$ for particle i and dimension j. Velocities and particles are randomly initialized. Each particle is initialized with uniformly random values in a bounded range [0,1]. For each particle, a filter is separately retrained and cross-correlated with false and true images. Subsequently, the P_r ratio is calculated using the average of peak intensity for false and true samples defined in the cross-validation dataset. Pr value is minimized for each filter using the algorithm given in Figure 15. PSO exits on either completion of the maximum number of epochs or getting the same minimum value of Pr for the number of epochs. Equations (43) and (44) are used to update the velocity and position of each particle.

$$v_{i,j}^{itr+1} = wv_{i,j}^{itr} + s_1 r()(pBest_{i,j} - p_{i,j}^{itr}) + s_2 R()(gBest_j - p_{i,j}^{itr}), \quad (43)$$

$$p_{i,j}^{itr+1} = p_{i,j}^{itr} + v_{i,j}^{itr+1}, \quad (44)$$

where itr denotes iteration, $i = 1, 2, ..., 20$ and $j = 1, 2$. s_1 and s_2 are acceleration constants. $r()$ and $R()$ are random functions, while w denotes the influence of motion in previous iteration, $0 \leq w < 1$. $pBest$ is the particle's best position, while $gBest$ is the global best position.

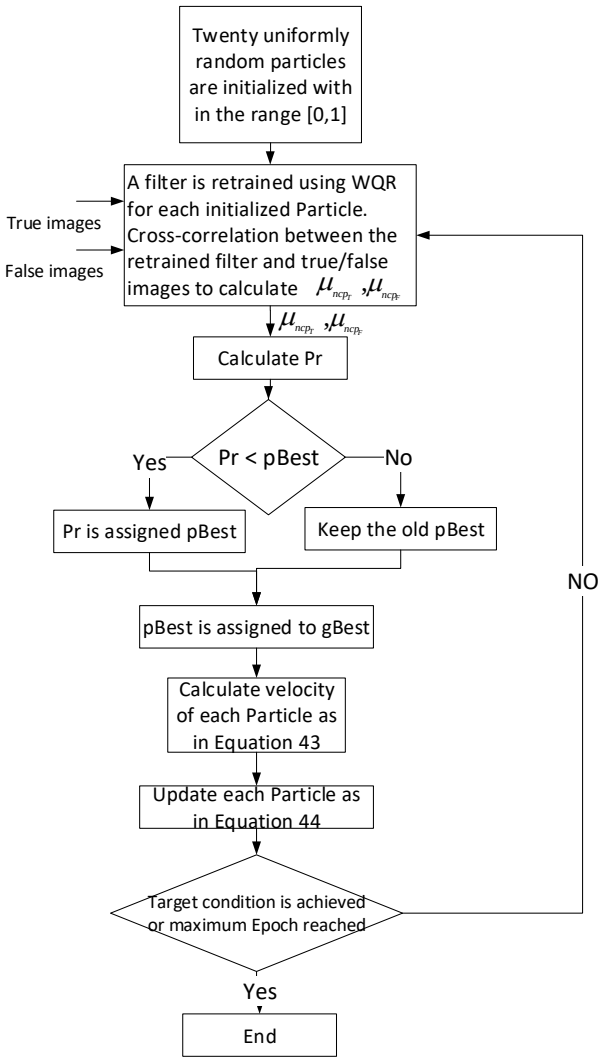


Figure 15. Flow diagram of Particle Swarm Optimization (PSO)-based parameter optimization of β and γ.

4.3. Performance Analysis

4.3.1. Rotational Analysis

For comparison, the full precision responses of 16-bit spatially-trained and frequency-trained EEMACH are considered a baseline for the DFP and Po2 quantized trained filters. For each compression ratio, the ST and FT filters are quantized using both the proposed approaches.

Commencing with a compression ratio of 16, the DFP and Po2 quantization schemes are separately analyzed for spatial and frequency domains. The contemplated strategies show nearly identical performance graphs. Detection responses of all the quantization schemes are slightly above the threshold. For direct quantization and retraining filters, the detection on the average response is ∼23% more than the baseline. In the case of inverse, log-map transform on the average response is ∼30% above the baseline in Figure 16a (box 1). The log-map quantization response has a slight dip around 100 degrees in Figure 16a (dip 1) and 300 degrees in Figure 16a (dip 2). On average, the log-map response is ∼35% below the baseline in Figure 16a (box 2).

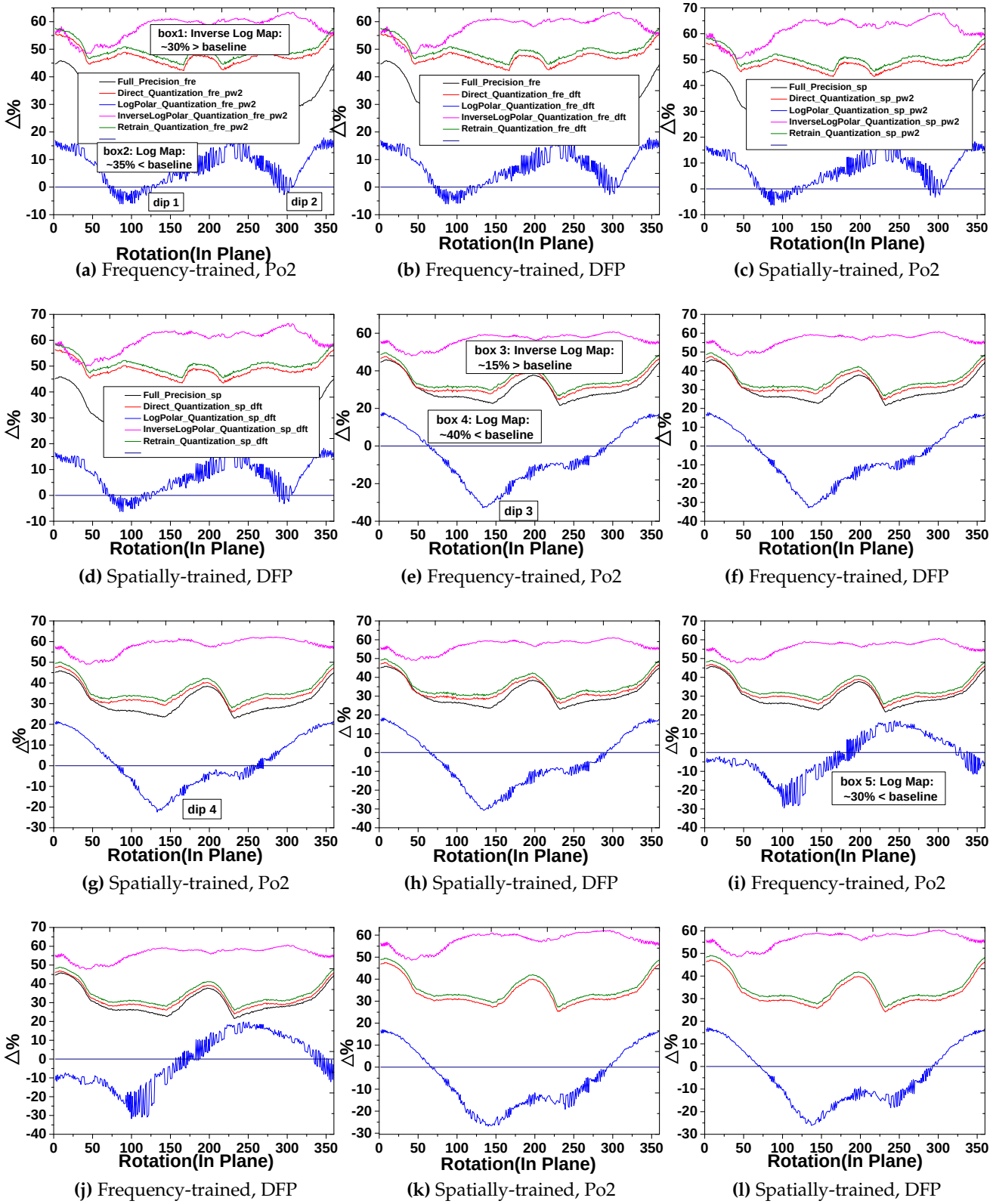
With compression ratio 8 for inverse log-map, the response reduces to ∼15% on average above the baseline in Figure 16e (box 3). Direct and retrained quantization filters show responses, which are almost similar to the reference responses. The log-map pre-processing on the average response is ∼40% below the baseline in Figure 16e (box 4). In the second tetrad (e–h), the log-map response remained lower than the threshold in a 50–300 degree interval in Figure 16e (dip 3) for each quantization type. Contrary to the frequency-trained filter, the response of spatially-trained filters for log-map quantization remains below the threshold within 75–270 degree interval in Figure 16g (dip 4). Subsequently, for all spatial responses below compression ratio 8, the log-map pre-processing response for both quantization types diminish below the threshold from 75 to 270 degrees in Figure 16o (dip 5). No significant change is observed in the rest of the responses as compared to the previous cases. All the responses, except for the log-map quantization, do not significantly vary.

Regarding the compression ratio of 5.33, the average response of the log-map pre-processing remains ∼30% below the baseline; however, for all compression ratios below 5.33, the log-map pre-processing for FT filters with the Po2 quantization shows a response that diminishes below the threshold of 0–190 degree interval in Figure 16m (dip 6). For compression ratio 4 or below, the DFP quantization for FT filter remains below the threshold for a 50–190 degree interval. Consequently, the full precision and direct quantization responses have almost identical curves with a gradual drop in the compression ratio.

4.3.2. Scale and Moving Light Analysis

Dataset 2 includes the image of a car at different scales. Some samples are shown in Figure 17. In order to investigate the resilience of the compressed configurations to handle the target's scalability, filter detection responses are measured on the scale of 0–400% of the original target size, and it is given in Figure 18. Similar to the rotational test, each set of the four graphs obtain for the following corresponding compression ratios: 16, 8, 5.33, 4, 8, and 1.33.

For the 16 compression ratio, the full precision detection response for both ST and FT filter is above the threshold up to 125% scalability. Beyond this scale, this response mainly revolved around the threshold value; however, the detection response of the inverse log-map is well above the threshold with a slight fall around 225% scale in Figure 18a (dip 1). For direct and retrained quantization, the detection score is above the threshold up to 350% in Figure 18a (dip 2), whereas log-map pre-processing is only successful up to 80% scale in Figure 18a (dip 3).



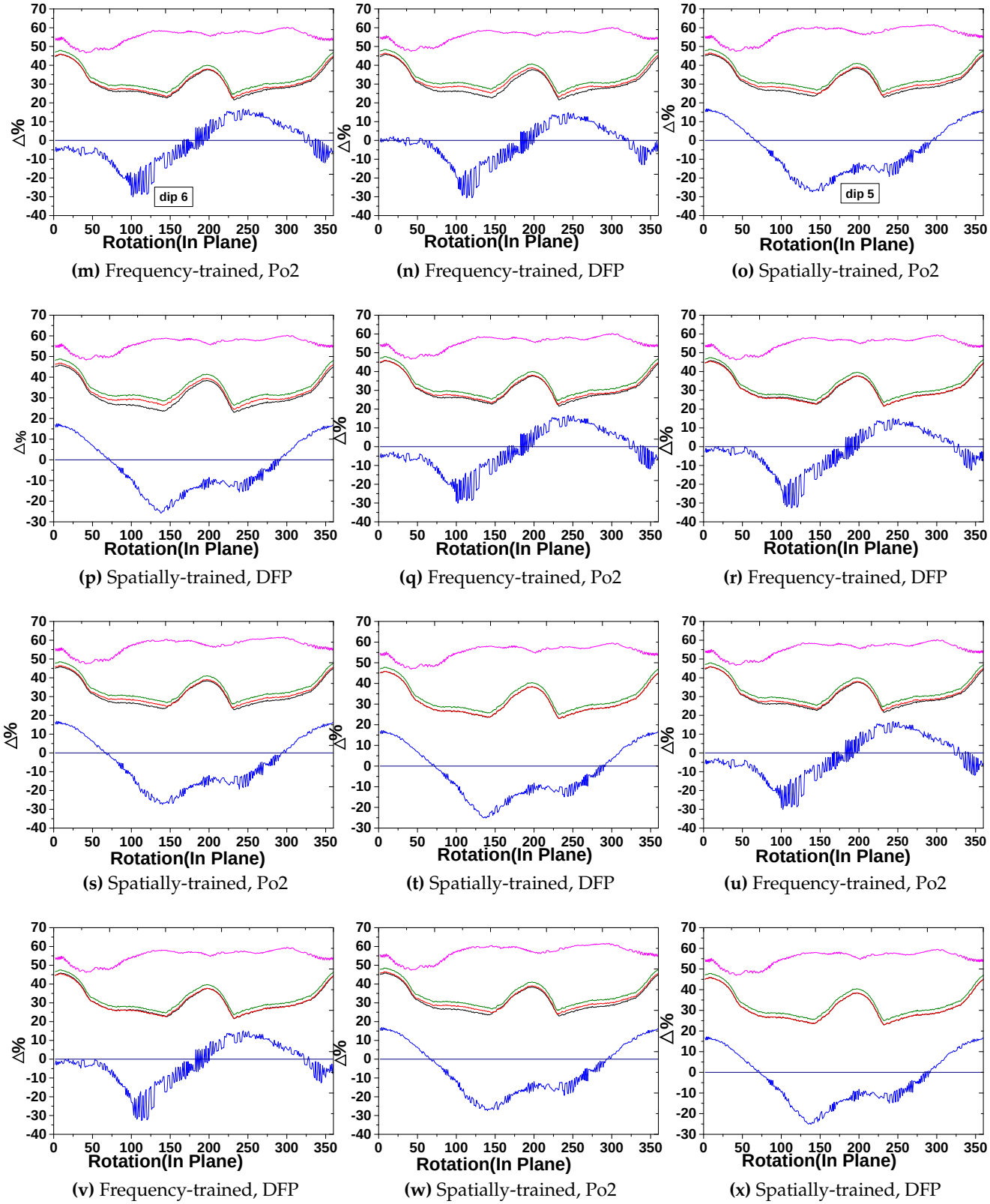
(a) Frequency-trained, Po2 **(b)** Frequency-trained, DFP **(c)** Spatially-trained, Po2

(d) Spatially-trained, DFP **(e)** Frequency-trained, Po2 **(f)** Frequency-trained, DFP

(g) Spatially-trained, Po2 **(h)** Spatially-trained, DFP **(i)** Frequency-trained, Po2

(j) Frequency-trained, DFP **(k)** Spatially-trained, Po2 **(l)** Spatially-trained, DFP

Figure 16. *Cont.*



(m) Frequency-trained, Po2 **(n)** Frequency-trained, DFP **(o)** Spatially-trained, Po2

(p) Spatially-trained, DFP **(q)** Frequency-trained, Po2 **(r)** Frequency-trained, DFP

(s) Spatially-trained, Po2 **(t)** Spatially-trained, DFP **(u)** Frequency-trained, Po2

(v) Frequency-trained, DFP **(w)** Spatially-trained, Po2 **(x)** Spatially-trained, DFP

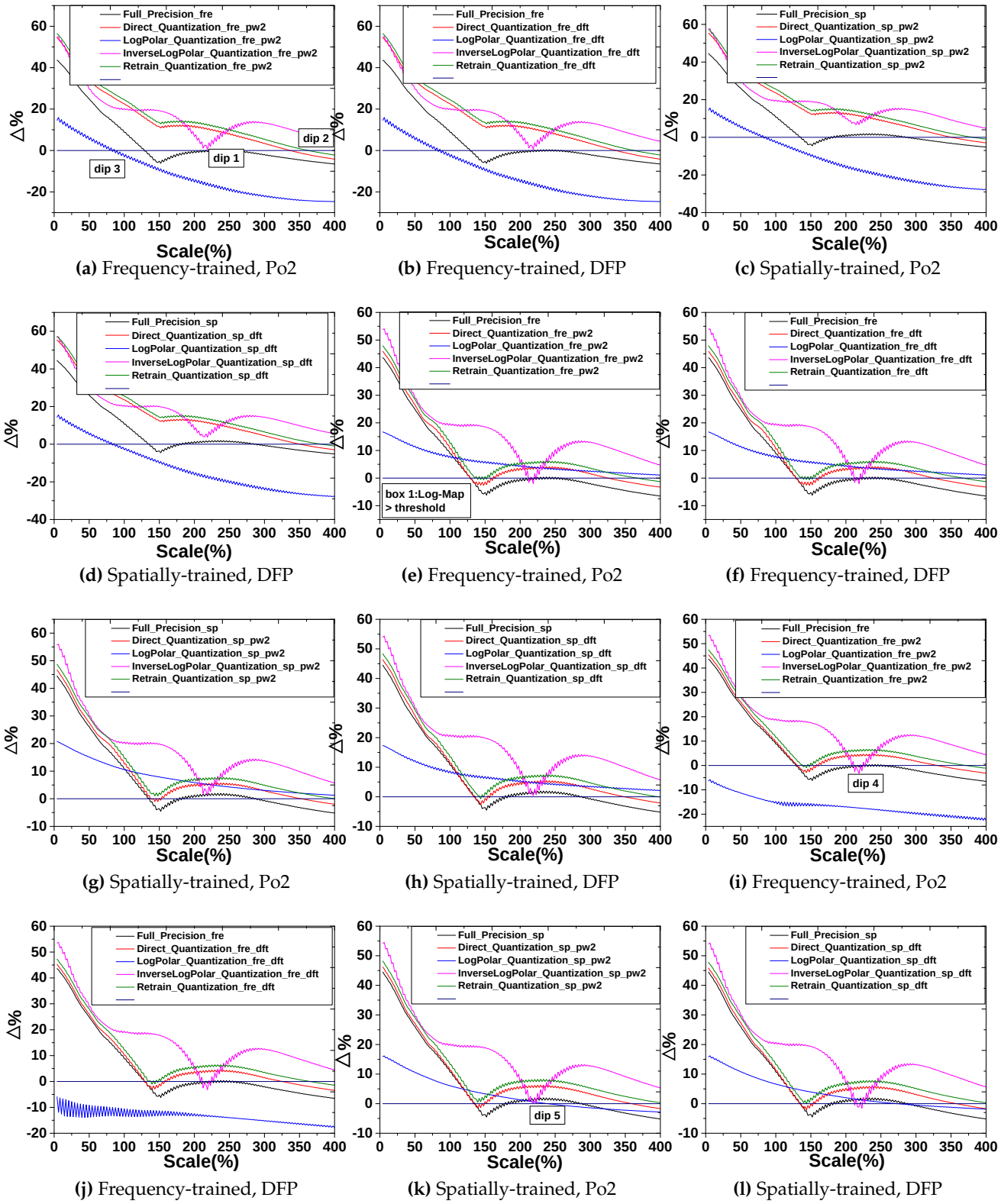
Figure 16. Rotation test for CPR. Each graph represents the responses from 0–360 degree rotation of the testing object. For brevity, every four consecutive graphs, (**a**–**d**), (**e**–**h**), (**i**–**l**), (**m**–**p**), (**q**–**t**), and (**u**–**x**), correspond to compression ratios of 16, 8, 5.33, 4, 2, and 1.33, respectively. Four columns represent Po2 frequency-trained, DFP frequency-trained, Po2 spatially-trained, and DFP spatially-trained filters, respectively.

Figure 17. Random samples of a car from dataset 2 are illustrated at different scales.

When the compression ratio is 8, all the curves' detection responses do not change much except for the log-map pre-processing. In that case, the response increase continued above the threshold on a scale of 0 to 400% in Figure 18e (box 1). Overall, compression ratio 8 is found to be more resilient in terms of scale enhancements for each type of quantization.

For the compression ratio of 5.33, the FT filter's detection score for the inverse log-map pre-processing has a slightly deeper dip with almost 225% scale enhancement in Figure 18i (dip 4). This drop stays above the threshold for the spatially-trained filter in Figure 18k (dip 5). On the other hand, for compression ratio below 5.33, this dip Figure 18m (dip 6) in the detection score goes even deeper for ST and FT filters for both compression schemes but the log-map pre-processing shows a detection score below the threshold for FT-quantized filter versions.

Conversely, the detection score remains well above the threshold for the ST versions of filters. The remaining quantization versions do not considerably alter its detection score by analyzing the curves related to the rest of the compression ratios. Sequel to a comprehensive analysis of graphs in Figure 18, the detection responses of ST quantized filters are more resilient to scale enhancements than the FT quantized filters. Conversely, the detection score remains well above the threshold for the ST filter versions. By analyzing the rest of the compression ratios' curves, the remaining quantization versions do not significantly alter its detection score. The comprehensive analysis is presented in the graphs in Figure 18, which show more resilience in the detection responses of the ST-quantized filters than the FT-quantized filters. Dataset 3 includes more than 1000 car images captured under various lighting conditions. Each image has a background, which shows that it belonged to a specific set of images developed by incremental rotation from 0 to 360 degrees under a particular light setting around the car as shown in Figure 19. Overall, the compressed versions of the filter exhibit excellent responses under different lighting conditions. For brevity, the compression ratios of 8 and 16 are demonstrated in Figures 20 and 21, respectively. Generally, for all compression ratios, including log-map and inverse log-map, pre-processing exhibits superior performance as compared to the baseline. That is equally valid for both ST and FT filters; however, the retrained and direct quantization filter responses are below the baseline. In Figure 20, responses of each graph for all the quantized instances are found identical. In comparison, Figure 21 expresses a better response to the FT filter as compared to the ST filter for both cases of the inverse log-map and the log-map. Beyond a compression ratio of 8, the performance graphs do not change; however, their responses remain well above the threshold. Table 4 explains the legends in Figures 16, 18–21.



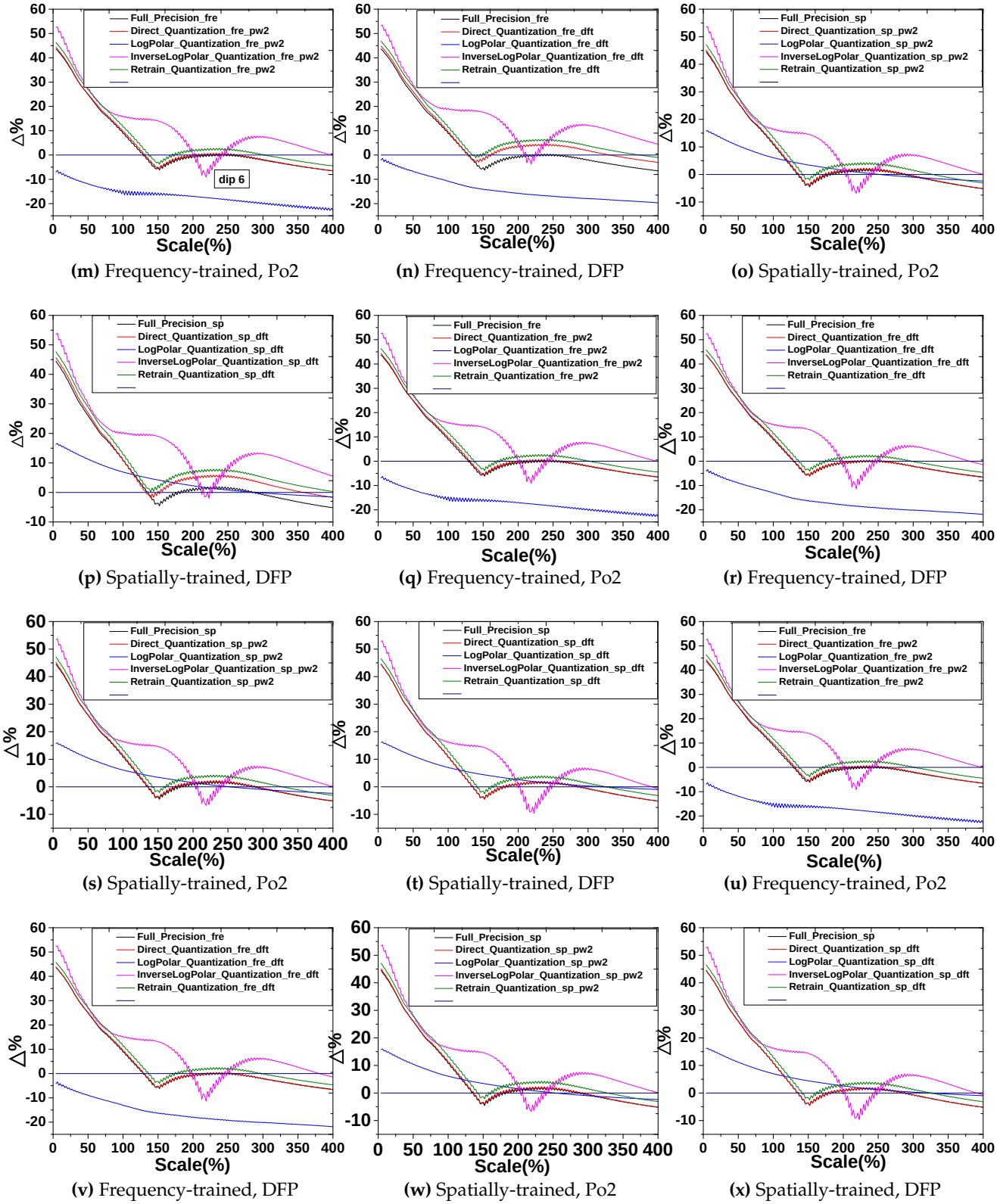
(a) Frequency-trained, Po2 **(b)** Frequency-trained, DFP **(c)** Spatially-trained, Po2

(d) Spatially-trained, DFP **(e)** Frequency-trained, Po2 **(f)** Frequency-trained, DFP

(g) Spatially-trained, Po2 **(h)** Spatially-trained, DFP **(i)** Frequency-trained, Po2

(j) Frequency-trained, DFP **(k)** Spatially-trained, Po2 **(l)** Spatially-trained, DFP

Figure 18. *Cont.*



(m) Frequency-trained, Po2 **(n)** Frequency-trained, DFP **(o)** Spatially-trained, Po2

(p) Spatially-trained, DFP **(q)** Frequency-trained, Po2 **(r)** Frequency-trained, DFP

(s) Spatially-trained, Po2 **(t)** Spatially-trained, DFP **(u)** Frequency-trained, Po2

(v) Frequency-trained, DFP **(w)** Spatially-trained, Po2 **(x)** Spatially-trained, DFP

Figure 18. Scalability test for CPR. Each graph represents the responses up to 400% maximum scale of the testing object. For brevity, every four consecutive graphs, (**a**–**d**), (**e**–**h**), (**i**–**l**), (**m**–**p**), (**q**–**t**), and (**u**–**x**), correspond to compression ratios of 16, 8, 5.33, 4, 2, and 1.33, respectively. Four columns represent Po2 frequency-trained, DFP frequency-trained, Po2 spatially-trained, and DFP spatially-trained filters, respectively.

Figure 19. Samples of a car from dataset 3 are illustrated under different lighting conditions.

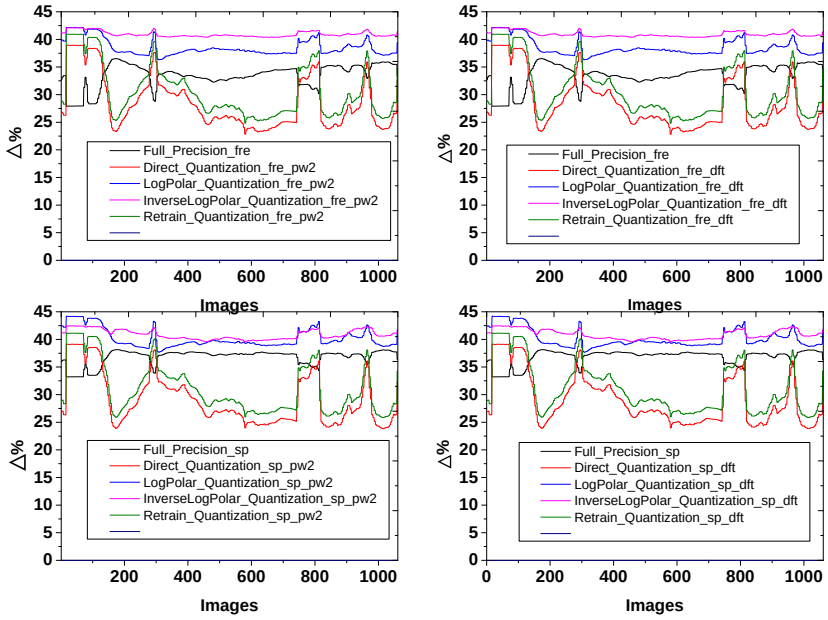


Figure 20. CPR responses for the object under different moving lighting conditions. Graphs with a compression ratio of sixteen contain the first row, which represents Po2 frequency-trained and DFP frequency-trained filter graphs, respectively. The second row represents Po2 spatially-trained, and DFP spatially-trained filter graphs, respectively.

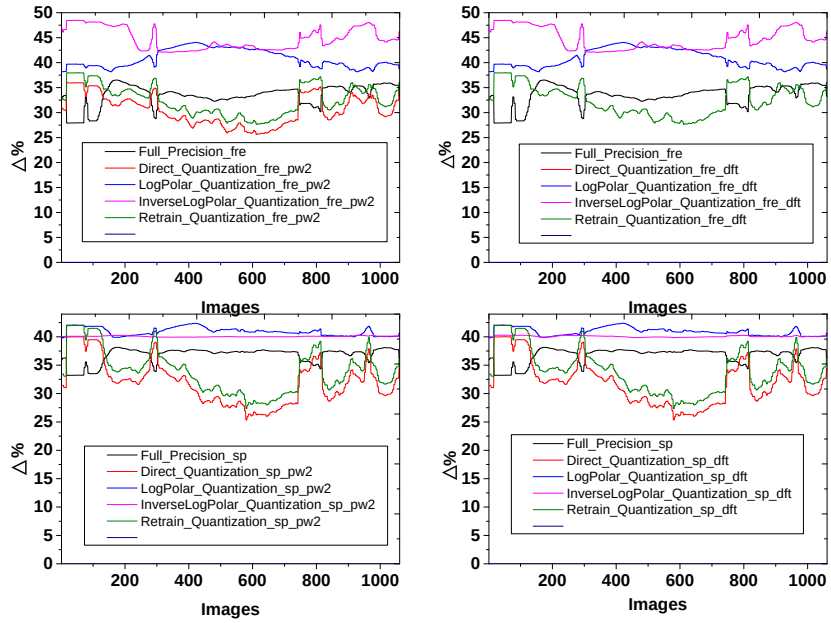


Figure 21. CPR responses for the object under different moving lighting conditions. Graphs with a compression ratio of eight contain the first row which represents Po2 frequency-trained and DFP frequency-trained filter graphs, respectively. The second row represents Po2 spatially-trained, and DFP spatially-trained filter graphs, respectively.

Table 4. Description of legends employed in Figures 16–21.

Legends	Response Description	Legends	Response Description
Full_Precision_sp	Floating-point precision of ST filter	Full_Precision_fre	Floating-point precision of ST filter
Direct_Quantization_fre_pw2	Direct quantization using Po2 of FT filter	InverseLogPolar_Quantization _fre_pw2	Quantization with Inverse log-polar pre-processing using Po2 of FT filter
Direct_Quantization_fre_dft	Direct quantization using DFP of FT filter	InverseLogPolar_Quantization _fre_dft	Quantization with Inverse log-polar pre-processing using DFP of FT filter
Direct_Quantization_sp_pw2	Direct quantization using Po2 of ST filter	InverseLogPolar_Quantization _sp_pw2	Quantization with Inverse log-polar pre-processing using Po2 of ST filter
Direct_Quantization_sp_dft	Direct quantization using DFP of ST filter	InverseLogPolar_Quantization _sp_dft	Quantization with Inverse log-polar pre-processing using DFP of ST filter
LogPolar_Quantization _fre_pw2	Quantization with log-polar pre-processing using Po2 of FT filter	Retrain_Quantization _fre_pw2	Retrain-quantization using Po2 of FT filter
LogPolar_Quantization _fre_dft	Quantization with log-polar pre-processing using DFP of FT filter	Retrain_Quantization _fre_dft	Retrain-quantization using DFP of FT filter
LogPolar_Quantization _sp_pw2	Quantization with log-polar pre-processing using Po2 of ST filter	Retrain_Quantization _sp_pw2	Retrain-quantization using Po2 of ST filter
LogPolar_Quantization _sp_dft	Quantization with log-polar pre-processing using DFP of ST filter	Retrain_Quantization _sp_dft	Retrain-quantization using DFP of ST filter

4.3.3. ROC Comparative Analysis

Typically, the CPR paradigm's evaluation analysis is achieved through conventional Receiver Operator Characteristic (ROC). Previously, the EMACH [19] and EEMACH [20] were analyzed in the available literature using the ROC analysis approach, but the issues with the ROC results made it insignificant for statistical analysis and inconsistent for application. Each compression level has a ROC curve, so each is considered a separate classifier; therefore, there are many ROC curves. The full-precision implementation has a distinct ROC curve for each trained method. These FT or ST methods provide a baseline for comparison to the corresponding compression rates. Since each approach holds 16 curves for each trained method, a total of 32 ROC curves are evaluated for each method. All 32 ROC curves should be compared with the corresponding ROC baseline curve representing full precision to find the compression outcomes for each bit-width. In our experimental analysis, Z, D, E, and their corresponding *p*-values are used to describe the likeness between the baseline ROC and the corresponding compressed versions for each method, for example, direct, log-map, and inverse log-map. E measure may highlight the differences between any two paired ROCs and the E values with their *p*-values are more significant as compared to Z and D; therefore, for brevity, we have only discussed the bit-widths having a minimum E value. See Appendix B for detailed results.

Figure 22a illustrates the E values and *p*-values [49,50] for all above-mentioned methods for ROC comparison. E and *p*-value demonstrate an integrated absolute difference between two ROC curves; so, the smaller E value shows the two ROC curves' closeness. The large value of E and $p < 0.05$ illustrates ROC's degradation due to corresponding bit-width compression. For direct quantization, the FT filter has the least E value 1308 with *p*-value < 0.92 for both Po2 and DFP. This indicates a strong closeness between ROCs,

which implies nearly equal classification performance, like original ROCs. The ST filter has E = 2368, p-value < 0.1525, E = 1106 and p-value < 0.106 for Po2 and DFP, respectively which means relatively less closeness as compared to FT. For log-map transform, the E value of all FT and ST quantization schemes varies between 83,184 and 88,000 with p-value < 2.2×10^{-16}. This shows a classification of performance degradation for all bit-width compression. For inverse log-map transform, the E value of all FT and ST quantization schemes varies between 29,714 and 32,948 with p-value < 2.2×10^{-16}. This demonstrates relatively less classification performance degradation for all bit-width compression as compared to log-map transform. For WQR, the ST filter has E = 3618, p-value < 0.028, E = 2442 and p-value < 0.053 for Po2 and DFP, respectively. These bit-widths have better classification performances than log-map and inverse log-map but less classification performance than the direct quantization.

For comparing ROC curves, the area under the curve (AUC) is assumed as an accuracy measure. In Equation (45), we present parameter Z and p-value to find the difference between the AUC of the two curves.

$$Z = \frac{\theta_2 - \theta_2}{\sigma^2}. \tag{45}$$

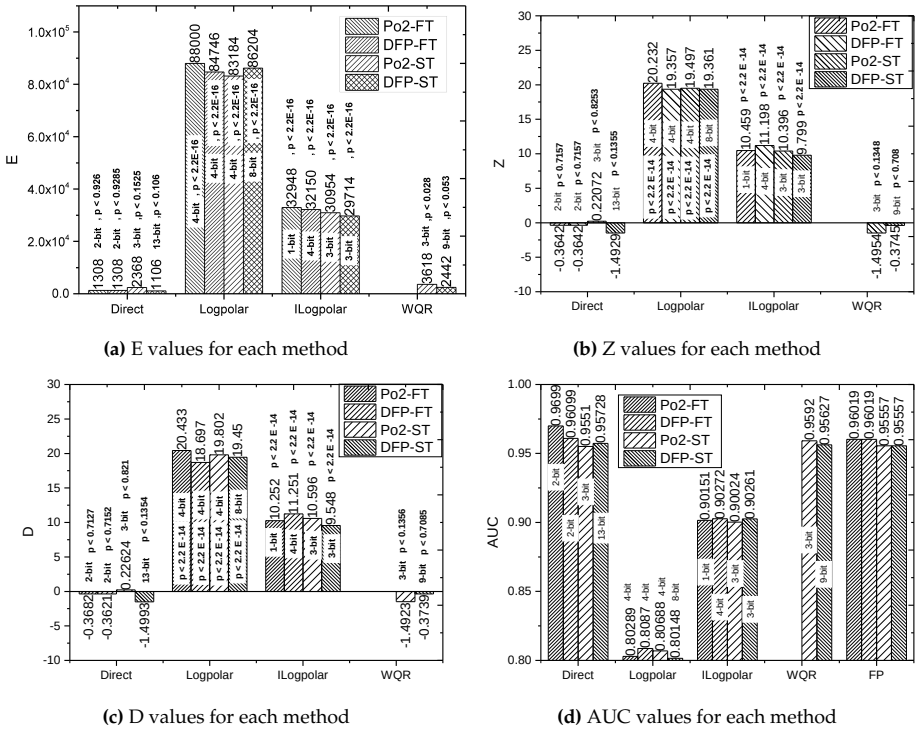
In Equation (45), θ_1 and θ_2 denote the respective AUCs of ROC1 and ROC2, while σ^2 is the standard deviation of the difference between the thetas. Figure 22b illustrates the Z value and p-value [51] for all above-mentioned methods for AUC comparison. The negative and small values indicate no or insignificant drop in AUC, while a large value demonstrates signification fall in AUC. For direct quantization, the Z value varies from -1.4929 to 0.22072, with p-values showing no significant difference between the AUC of quantized versions and the original. For the log-map transform, the Z value for ST and FT varies from 19.357 to 20.232 with p-value 2.2×10^{-16} illustrating a significant AUC loss for all quantization schemes. For the inverse log-map transform, the Z value for ST and FT varies from 9.799 to 11.198, with a p-value 2.2×10^{-16} illustrating a less significant AUC loss than the log-map. For the WQR, the ST filter has the Z value -1.4954 with p-value < 0.1348 for Po2 and Z = -0.3745 with p-value < 0.708 for DFP, which shows better AUC of bit-widths than log-map and inverse log-map transform. Equation (46) is another measure to compare the AUC's of ROCs, whereas D is given as follows:

$$D = \frac{V^r(\theta_r) - V^s(\theta_s)}{\sqrt{\sigma_r^2 + \sigma_s^2}}. \tag{46}$$

In Equation (46), θ_r and θ_s denote the respective AUCs of r and s ROC curves, while σ_r^2 and σ_s^2 are the standard deviations of "r" and "s", respectively. Figure 22c demonstrates the D value and p-value [52] for AUC comparison. The D values further support the results of E and Z. The D value and p-value for direct quantization are showed no significant AUC loss for all quantization methods. The D value varies between -1.4993 to 0.22624 with p-values. For the log-map transform, the D value changes from 18.697 to 20.433 for all quantization. This demonstrates a lot of performance degradation (AUC). For inverse log-map transform, the D value changes from 9.548 to 11.251 for all quantization. This demonstrates a less AUC degradation as compared to the log-map transform. Figure 22d illustrates the AUC values for each quantization method for both ST and FT filters and its comparisons with baseline AUCs.

To signify the benefits of quantization to the CPR filters, the performance parameters are demonstrated by column graphs in Figures 23–25. The compressed versions of these filters are concluded based on the least E value. The selected performance parameters include sparsity, CPU execution time, and memory minimization. Here, the sparsity implies the number of zero weights in the trained filters, which implies a reduction in floating-point operations workload and speed-up in the convolution process during inference. The sparsity of the corresponding quantization schemes is displayed in Figure 23. On average, the direct and inverse log-map quantization schemes have better weight sparsity values as

compared to full-precision versions. The best case is Po2 compression, with a compression ratio 16 but the log-map sparsity is insignificant. For a few instances, their sparsity lagged behind the full precision. The second performance measure is the memory minimization, as shown in Figure 24. Again, the inverse log-map and the direct one have meager memory requirements as compared to the full-precision version.



(a) E values for each method

(b) Z values for each method

(c) D values for each method

(d) AUC values for each method

Figure 22. Performance measures of each method for quantized bit-widths.

On the other hand, log-map memory requirements are modest. This third measure is the CPU execution time, which is given in Figure 25. The CPU is i5-2500k, 3.30 GHz with 3301 MHz 4 Core processor, whereas the system is 64-bit, equipped with 16GB RAM. The CPU time is measured using *tic* and *toc* standard functions available in MATLAB. Overall, the inverse log-map has the least execution time on the CPU, which is followed by the direct quantization scheme. The fastest case is the Po2 quantization with a 16-bit compression, which showed $\sim$8.90$\times$ speed-up capacity during the full-precision implementation.

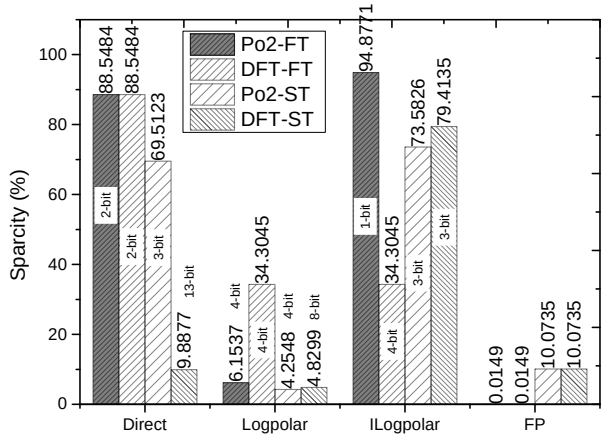


Figure 23. The graph demonstrates a sparsity comparison between direct, log-map, and inverse log-map quantization for filter bank. Note that it is filter sparsity.

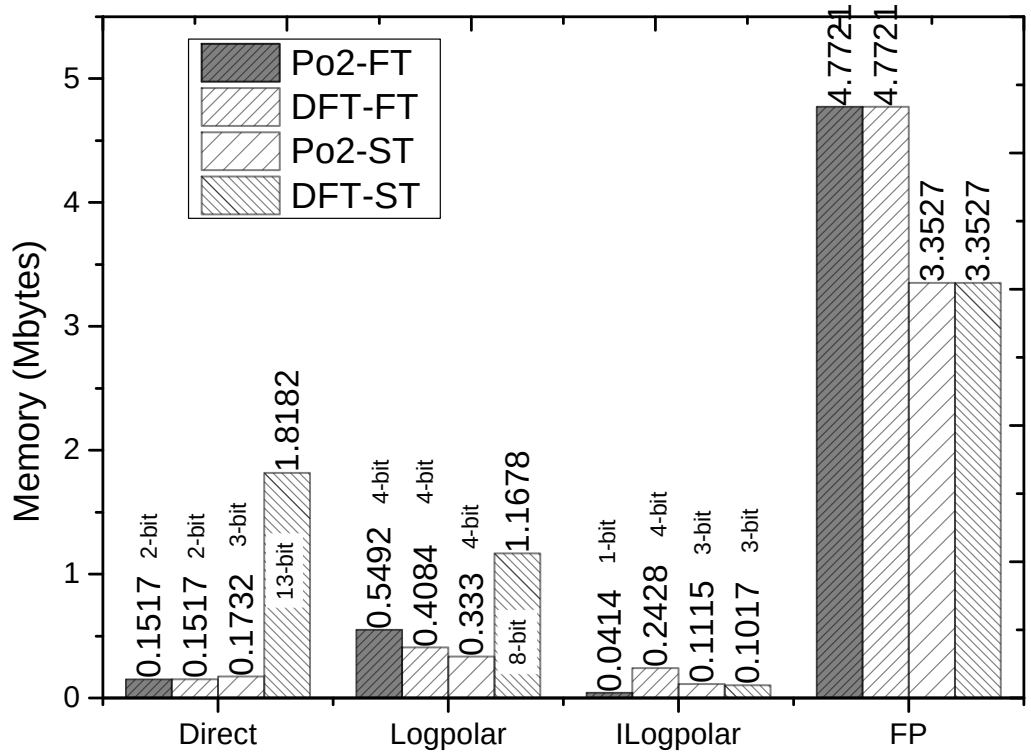


Figure 24. The graph demonstrates a memory comparison between direct, log-map, and inverse log-map quantization for filter bank.

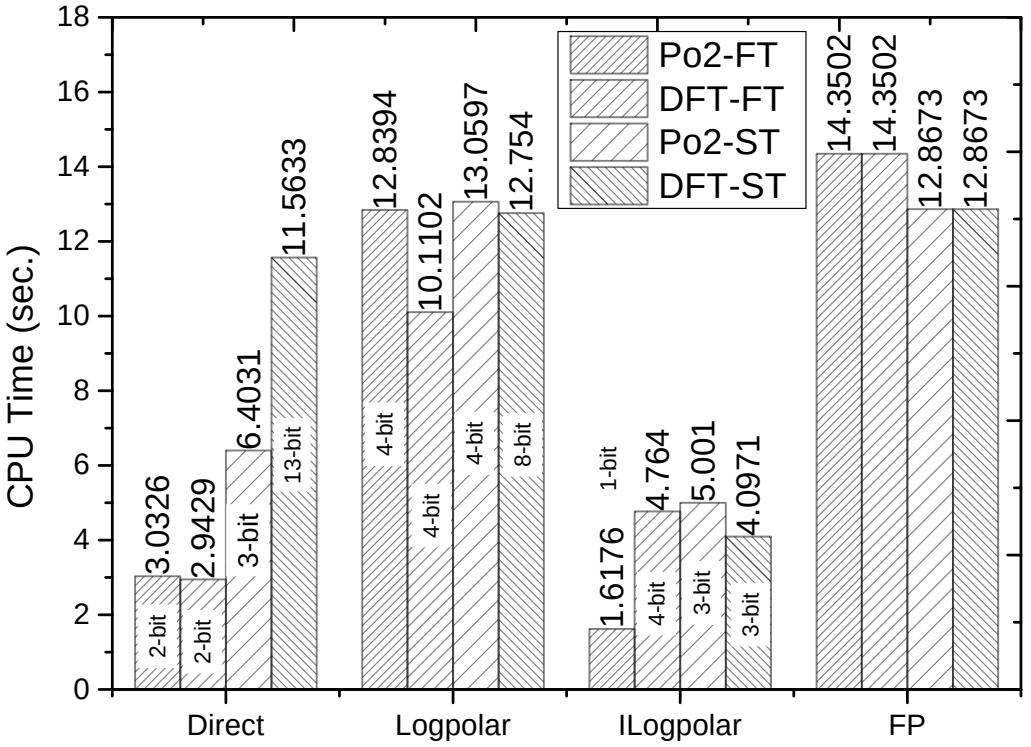


Figure 25. The graph demonstrates a speed-up comparison between direct, log-map, and inverse log-map quantization for filter bank.

5. Conclusions

The spatial-domain CPR filters require substantial computation resources and memory. The proposed weight quantization is imperative to reduce the computation workload,

processing time, and memory minimization. We propose the WQR approach and pre-processing steps, like log-map and inverse log-map, to improve the accuracy degradation through full-precision weight quantization. The WQR regulates the filter retraining process by fine-tuning the weights through any stated quantization scheme. The PSO is used for selecting WQR training parameters. Quantization error causes more accuracy loss to the ST filters than the FT filters, and WQR alleviates this accuracy loss. No accuracy degradation occurs at 9.88–88.54% MAC sparsity, 1.11×–4.73× speedup, and 14 is the maximum compression ratio for direct quantization. The inverse log-map achieves 34.30–94.87% MAC sparsity, 2.57×–8.90× speed-up, and maximum 1-bit compression with 6% accuracy loss, while the log-map achieves 4.25–34.30% MAC sparsity, 0.98×–1.12× speedup, and maximum 4-bit compression with 16% decline in accuracy. To study the quantization for the CPR, Po2 and DFP quantization approaches are applied. The results showed that better ROC closeness of DFP quantization is assured with the floating-point, while Po2 achieved better precision reduction. Based on the results, it is unnecessary to perform the retraining procedure with DFP quantization. On the other hand, retaining with Po2 showed better performance improvement than the DFP quantization. It can easily be concluded that Po2 quantization is a preferable choice for CPR. Moreover, the Po2 implementation for the spatial-domain CPR on hardware is recommended for future work. Further, multiplication in Po2-quantized filter could be performed using the shift operation, which makes it more hardware-friendly; however, it needs optimized hardware. We consider recovering the accuracy loss of geometric pre-processing, Po2 quantization, and retraining for future work.

Author Contributions: Conceptualization, methodology and formal analysis, D.S.; supervision, A.H., S.R. and M.S.; writing—original draft, D.S.; writing—review and editing, M.A.H. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The APC was partially funded by National University of Sciences and Technology, Islamabad.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Relationship between Mean Square Error and Variance of Sample and Its Quantized Version

Whereas Y_i and $\hat{Y}_i$ denote the original image and the estimated image, respectively, N denotes the total number of pixels in the original image. $\triangle Y$ is the difference between an original image and its corresponding quantized version. $\mu_{\triangle Y}$ is the average of differences between original and estimated images.

As we know the equation,

$$\begin{aligned} \textstyle\sum_i^N (\triangle Y - \mu_{\triangle Y})^2 &= \textstyle\sum_i^N (\triangle Y)^2 - \frac{1}{N}(\sum_i^N \triangle Y)^2, \\ \triangle Y &= Y_i - \hat{Y}_i \end{aligned} \tag{A1}$$

$$\begin{aligned} \textstyle\sum_i^N (\triangle Y)^2 &= \textstyle\sum_i^N (Y_i - \hat{Y}_i)^2 \\ &= \textstyle\sum_i^N (Y_i^2 + \hat{Y}_i^2 - 2Y_i\hat{Y}_i) \\ &= \textstyle\sum_i^N Y_i^2 + \sum_i^N \hat{Y}_i^2 - 2\sum_i^N Y_i\hat{Y}_i. \end{aligned} \tag{A2}$$

By multiplying Equation (A2) with $\frac{1}{N}$,

$$\begin{aligned} -\frac{1}{N}(\sum_i^N \triangle Y)^2 &= -\frac{1}{N}(\sum_i^N (Y_i - \hat{Y}_i))^2 \\ &= -\frac{1}{N}(\sum_i^N Y_i - \sum_i^N \hat{Y}_i)^2 \\ &= -\frac{1}{N}((\sum_i^N Y_i)^2 - (\sum_i^N \hat{Y}_i)^2 + 2(\sum_i^N Y_i)(\sum_i^N \hat{Y}_i)) \\ &= -\frac{1}{N}(\sum_i^N Y_i)^2 - \frac{1}{N}(\sum_i^N \hat{Y}_i)^2 + \frac{2}{N}(\sum_i^N Y_i)(\sum_i^N \hat{Y}_i). \end{aligned} \tag{A3}$$

By substituting the Equation (A2) and Equation (A3) into L.H.S. of Equation (A1), we get

$$\sum_i^N (\triangle Y - \mu_{\triangle Y})^2 = \sum_i^N Y_i^2 - \frac{1}{N}(\sum_i^N Y_i)^2 + \sum_i^N \hat{Y}_i^2 - \frac{1}{N}(\sum_i^N \hat{Y}_i)^2 - 2\sum_i^N Y_i\hat{Y}_i + \frac{2}{N}(\sum_i^N Y_i)(\sum_i^N \hat{Y}_i). \tag{A4}$$

Equation (A4) can be simplified by substituting $\mu_{Y_i} = \frac{1}{N}(\sum_i^N Y_i)$ and $\mu_{\hat{Y}_i} = \frac{1}{N}(\sum_i^N \hat{Y}_i)$,

$$\sum_i^N (\triangle Y)^2 - \frac{1}{N}(\sum_i^N \triangle Y)^2 = \sum_i^N (Y_i - \mu_{Y_i})^2 + \sum_i^N (\hat{Y}_i - \mu_{\hat{Y}_i})^2 - 2\sum_i^N Y_i\hat{Y}_i + \frac{2}{N}(\sum_i^N Y_i)(\sum_i^N \hat{Y}_i), \tag{A5}$$

$$\begin{aligned} \sum_i^N (\triangle Y)^2 = \sum_i^N (Y_i - \mu_{Y_i})^2 + \sum_i^N (\hat{Y}_i - \mu_{\hat{Y}_i})^2 - 2\sum_i^N Y_i\hat{Y}_i + \frac{2}{N}(\sum_i^N Y_i)(\sum_i^N \hat{Y}_i) + \frac{1}{N}(\sum_i^N Y_i)^2 \\ + \frac{1}{N}(\sum_i^N \hat{Y}_i)^2 - \frac{2}{N}(\sum_i^N Y_i)(\sum_i^N \hat{Y}_i). \end{aligned} \tag{A6}$$

By simplifying above equations, we get

$$\sum_i^N (\triangle Y)^2 = \sum_i^N (Y_i - \mu_{Y_i})^2 + \sum_i^N (\hat{Y}_i - \mu_{\hat{Y}_i})^2 - 2\sum_i^N Y_i\hat{Y}_i + \frac{1}{N}(\sum_i^N Y_i)^2 + \frac{1}{N}(\sum_i^N \hat{Y}_i)^2. \tag{A7}$$

By multiplying Equation (A7) with $\frac{1}{N}$,

$$\begin{aligned} \frac{1}{N}\sum_i^N (\triangle Y)^2 = \frac{1}{N}\sum_i^N (Y_i - \mu_{Y_i})^2 + \frac{1}{N}\sum_i^N (\hat{Y}_i - \mu_{\hat{Y}_i})^2 \\ - \frac{2}{N}\sum_i^N Y_i\hat{Y}_i + \frac{1}{N^2}(\sum_i^N Y_i)^2 + \frac{1}{N^2}(\sum_i^N \hat{Y}_i)^2. \end{aligned} \tag{A8}$$

$\sigma_{Y_i}^2$ and $\sigma_{\hat{Y}_i}^2$ are the variances of the original image and its corresponding quantized version, respectively.

$$MSE = \sigma_{Y_i}^2 + \sigma_{\hat{Y}_i}^2 - \frac{2}{N}\sum_i^N Y_i\hat{Y}_i + \frac{1}{N^2}(\sum_i^N Y_i)^2 + \frac{1}{N^2}(\sum_i^N \hat{Y}_i)^2, \tag{A9}$$

whereas $MSE = \frac{1}{N}\sum_i^N (\triangle Y)^2$, $\sigma_{Y_i}^2 = \frac{1}{N}\sum_i^N (Y_i - \mu_{Y_i})^2$, and $\sigma_{\hat{Y}_i}^2 = \frac{1}{N}\sum_i^N (\hat{Y}_i - \mu_{\hat{Y}_i})^2$:

$$PNSR = 20\log_{10}\frac{MAX_f}{\sqrt{MSE}}. \tag{A10}$$

Appendix B. Performance Tables

Table A1. Performance measurements of direct Power-of-Two (Po2) quantization scheme.

Bit-Width	**Frequency, AUC = 0.9601852**							**Spatial, AUC = 0.9555699**						
	Z	***p*-Value <**	**D**	***p*-Value <**	**E**	***p*-Value <**	**AUC**	**Z**	***p*-Value <**	**D**	***p*-Value <**	**E**	***p*-Value <**	**AUC**
1	−0.2462	0.8055	−0.2473	0.8047	10,574	2.20×10^{-16}	0.96109	−0.3126	0.7546	−0.3114	0.7555	18,590	2.20×10^{-16}	0.95734
2	−0.3642	0.7157	−0.3682	0.7127	1308	0.926	0.96099	2.3705	0.01776	2.3321	0.0197	7706	2.20×10^{-16}	0.94758
3	3.9235	8.73×10^{-5}	3.9435	8.03×10^{-5}	3960	2.20×10^{-16}	0.95384	0.22072	0.8253	0.22624	0.821	2368	0.1425	0.9551
4	4.1619	3.16×10^{-5}	4.1554	3.25×10^{-5}	2870	2.20×10^{-16}	0.95552	0.70579	0.4803	0.71656	0.4736	2914	0.1435	0.95401
5	4.4951	6.95×10^{-6}	4.5996	4.23×10^{-6}	3114	2.20×10^{-16}	0.9551	1.0599	0.2892	1.0428	0.297	2972	0.115	0.95325
6	4.4951	6.95×10^{-6}	4.507	6.58×10^{-6}	3114	2.20×10^{-16}	0.9551	1.0599	0.2892	1.0751	0.2823	2972	0.117	0.95325
7	4.4951	6.95×10^{-6}	4.398	1.09×10^{-5}	3114	2.20×10^{-16}	0.9551	1.0599	0.2892	1.0528	0.2924	2972	0.103	0.95325
8	4.4951	6.95×10^{-6}	4.4449	8.79×10^{-6}	3114	2.20×10^{-16}	0.9551	1.0599	0.2892	1.0777	0.2812	2972	0.119	0.95325
9	4.4951	6.95×10^{-6}	4.5224	6.11×10^{-6}	3114	2.20×10^{-16}	0.9551	1.0599	0.2892	1.0657	0.2866	2972	0.111	0.95325
10	4.4951	6.95×10^{-6}	4.5241	6.06×10^{-6}	3114	2.20×10^{-16}	0.9551	1.0599	0.2892	1.0655	0.2866	2972	0.1195	0.95325
11	4.4951	6.95×10^{-6}	4.5114	6.44×10^{-6}	3114	2.20×10^{-16}	0.9551	1.0599	0.2892	1.0849	0.278	2972	0.119	0.95325
12	4.4951	6.95×10^{-6}	4.528	5.95×10^{-6}	3114	2.20×10^{-16}	0.9551	1.0599	0.2892	1.0418	0.2975	2972	0.1225	0.95325
13	4.4951	6.95×10^{-6}	4.4079	1.04×10^{-5}	3114	2.20×10^{-16}	0.9551	1.0599	0.2892	1.0544	0.2917	2972	0.1245	0.95325
14	4.4951	6.95×10^{-6}	4.4938	7.00×10^{-6}	3114	2.20×10^{-16}	0.9551	1.0599	0.2892	1.0746	0.2826	2972	0.1355	0.95325
15	4.4951	6.95×10^{-6}	4.4672	7.93×10^{-6}	3114	2.20×10^{-16}	0.9551	1.0599	0.2892	1.0528	0.2924	2972	0.1175	0.95325
16	4.4951	6.95×10^{-6}	4.3845	1.16×10^{-5}	3114	2.20×10^{-16}	0.9551	1.0599	0.2892	1.0692	0.285	2972	0.1275	0.95325
Mean	3.838516	0.0950876	3.829514	0.0948503	3505	0.057875	0.955788	0.981449	0.346773	0.982549	0.345194	4202.625	0.1073125	0.953317

Table A2. Performance measurements of direct Dynamic-Fixed-Point (DFP) quantization scheme.

Bit-Width	**Frequency, AUC = 0.9601852**							**Spatial, AUC = 0.9555699**						
	Z	***p*-Value <**	**D**	***p*-Value <**	**E**	***p*-Value <**	**AUC**	**Z**	***p*-Value <**	**D**	***p*-Value <**	**E**	***p*-Value <**	**AUC**
1	−0.2462	0.8055	−0.2416	0.8057	10,574	2.20×10^{-16}	0.96109	−0.2921	0.7702	−0.2986	0.7652	18,644	2.20×10^{-16}	0.95722
2	−0.3642	0.7157	−0.3621	0.7152	1308	0.9285	0.96099	−2.859	0.00425	−2.859	0.00425	5890	2.20×10^{-16}	0.96201
3	3.6811	0.000232	3.6785	0.000235	3244	5.00×10^{-4}	0.95496	−1.4347	0.1514	−1.4309	0.1525	2878	0.0075	0.95799
4	5.7414	9.39×10^{-9}	5.6051	2.08×10^{-8}	4800	2.20×10^{-16}	0.95186	−2.1165	0.0343	−2.0812	0.03742	3076	0.0015	0.95884
5	5.3215	1.03×10^{-7}	5.1707	2.33×10^{-7}	4794	2.20×10^{-16}	0.95198	−1.9785	0.04788	−2	0.0455	3212	2.20×10^{-16}	0.95862
6	5.7997	6.65×10^{-9}	5.7624	8.30×10^{-9}	3984	2.20×10^{-16}	0.95311	−2.9184	0.00352	−3.0227	0.00251	2062	0.003	0.95907
7	6.2809	3.37×10^{-10}	5.9641	2.46×10^{-9}	4038	2.20×10^{-16}	0.953	−2.5313	0.01136	−2.6258	0.00865	1806	0.0165	0.95855
8	6.6003	4.10×10^{-11}	6.549	5.79×10^{-11}	4202	2.20×10^{-16}	0.95272	−3.2168	0.0013	−3.1945	0.0014	2254	5.00×10^{-4}	0.95951
9	6.7322	1.67×10^{-11}	6.6732	2.50×10^{-11}	4326	2.20×10^{-16}	0.9525	−1.5704	0.1163	−1.5638	0.1179	1144	0.1025	0.95736
10	6.7452	1.53×10^{-11}	6.7872	1.14×10^{-11}	4296	2.20×10^{-16}	0.95256	−1.5612	0.1185	−1.6002	0.1096	1134	0.1015	0.95735
11	6.7298	1.70×10^{-11}	6.5328	6.46×10^{-11}	4284	2.20×10^{-16}	0.95258	−1.5262	0.127	−1.5251	0.1272	1126	0.1075	0.95731
12	6.7298	1.70×10^{-11}	6.5587	5.43×10^{-11}	4284	2.20×10^{-16}	0.95258	−1.5203	0.1284	−1.5532	0.1204	1126	0.109	0.9573
13	6.7299	1.70×10^{-11}	6.6692	2.57×10^{-11}	4280	2.20×10^{-16}	0.95258	−1.4929	0.1355	−1.493	0.1354	1106	0.106	0.95728
14	6.7357	1.63×10^{-11}	6.5614	5.33×10^{-11}	4280	2.20×10^{-16}	0.95258	−1.5074	0.1317	−1.4917	0.1358	1110	0.11	0.95729
15	6.7343	1.65×10^{-11}	6.5028	7.88×10^{-11}	4280	2.20×10^{-16}	0.95258	−1.5041	0.1326	−1.5299	0.126	1112	0.0985	0.95729
16	6.7349	1.64×10^{-11}	6.6145	3.73×10^{-11}	4282	2.20×10^{-16}	0.95258	−1.4929	0.1355	−1.4704	0.1415	1110	0.0935	0.95728
Mean	5.417898	0.0950895	5.31412	0.0950709	4453.5	0.0580625	0.953766	−1.84517	0.128107	−1.85875	0.126951	3049.375	0.0535938	0.95814

Table A3. Performance measurements of log-map pre-processing with Power-of-Two (Po2) quantization scheme.

Bit-Width	**Frequency, AUC = 0.9601852**							**Spatial, AUC = 0.9555699**						
	Z	***p*-Value <**	**D**	***p*-Value <**	**E**	***p*-Value <**	**AUC**	**Z**	***p*-Value <**	**D**	***p*-Value <**	**E**	***p*-Value <**	**AUC**
1	25.447	2.2×10^{-16}	25.506	2.2×10^{-16}	11,6428	2.2×10^{-16}	0.75207	20.725	2.2×10^{-16}	20.959	2.2×10^{-16}	153,240	2.2×10^{-16}	0.68165
2	22.362	2.2×10^{-16}	22.256	2.2×10^{-16}	88,130	2.2×10^{-16}	0.80265	19.84	2.2×10^{-16}	20.344	2.2×10^{-16}	88,062	2.2×10^{-16}	0.79816
3	19.913	2.2×10^{-16}	20.342	2.2×10^{-16}	84,376	2.2×10^{-16}	0.80936	22.533	2.2×10^{-16}	22.49	2.2×10^{-16}	105,932	2.2×10^{-16}	0.76622
4	20.232	2.2×10^{-16}	20.433	2.2×10^{-16}	88,000	2.2×10^{-16}	0.80289	19.497	2.2×10^{-16}	19.802	2.2×10^{-16}	83,184	2.2×10^{-16}	0.80688
5	20.223	2.2×10^{-16}	19.888	2.2×10^{-16}	88,022	2.2×10^{-16}	0.80285	19.508	2.2×10^{-16}	19.398	2.2×10^{-16}	83,272	2.2×10^{-16}	0.80672
6	20.223	2.2×10^{-16}	20.225	2.2×10^{-16}	88,022	2.2×10^{-16}	0.80285	19.508	2.2×10^{-16}	19.584	2.2×10^{-16}	83,272	2.2×10^{-16}	0.80672
7	20.223	2.2×10^{-16}	19.875	2.2×10^{-16}	88,022	2.2×10^{-16}	0.80285	19.508	2.2×10^{-16}	19.562	2.2×10^{-16}	83,272	2.2×10^{-16}	0.80672
8	20.223	2.2×10^{-16}	20.242	2.2×10^{-16}	88,022	2.2×10^{-16}	0.80285	19.508	2.2×10^{-16}	18.998	2.2×10^{-16}	83,272	2.2×10^{-16}	0.80672
9	20.223	2.2×10^{-16}	20.889	2.2×10^{-16}	88,022	2.2×10^{-16}	0.80285	19.508	2.2×10^{-16}	19.222	2.2×10^{-16}	83,272	2.2×10^{-16}	0.80672
10	20.223	2.2×10^{-16}	20.267	2.2×10^{-16}	88,022	2.2×10^{-16}	0.80285	19.508	2.2×10^{-16}	19.979	2.2×10^{-16}	83,272	2.2×10^{-16}	0.80672
11	20.223	2.2×10^{-16}	20.254	2.2×10^{-16}	88,022	2.2×10^{-16}	0.80285	19.508	2.2×10^{-16}	19.242	2.2×10^{-16}	83,272	2.2×10^{-16}	0.80672
12	20.223	2.2×10^{-16}	20.436	2.2×10^{-16}	88,022	2.2×10^{-16}	0.80285	19.508	2.2×10^{-16}	19.129	2.2×10^{-16}	83,272	2.2×10^{-16}	0.80672
13	20.223	2.2×10^{-16}	20.101	2.2×10^{-16}	88,022	2.2×10^{-16}	0.80285	19.508	2.2×10^{-16}	19.363	2.2×10^{-16}	83,272	2.2×10^{-16}	0.80672
14	20.223	2.2×10^{-16}	19.796	2.2×10^{-16}	88,022	2.2×10^{-16}	0.80285	19.508	2.2×10^{-16}	19.663	2.2×10^{-16}	83,272	2.2×10^{-16}	0.80672
15	20.223	2.2×10^{-16}	20.032	2.2×10^{-16}	88,022	2.2×10^{-16}	0.80285	19.508	2.2×10^{-16}	19.576	2.2×10^{-16}	83,272	2.2×10^{-16}	0.80672
16	20.223	2.2×10^{-16}	20.425	2.2×10^{-16}	88,022	2.2×10^{-16}	0.80285	19.508	2.2×10^{-16}	19.022	2.2×10^{-16}	83,272	2.2×10^{-16}	0.80672
Mean	20.66438	2.2×10^{-16}	20.68544	2.2×10^{-16}	89,574.88	2.2×10^{-16}	0.80007	19.79319	2.2×10^{-16}	19.77081	2.2×10^{-16}	89,355.13	2.2×10^{-16}	0.795847

Table A4. Performance measurements of log-map pre-processing with Dynamic-Fixed-Point (DFP) quantization scheme.

Bit-Width	**Frequency, AUC = 0.9601852**							**Spatial, AUC = 0.9555699**						
	Z	***p*-Value <**	**D**	***p*-Value <**	**E**	***p*-Value <**	**AUC**	**Z**	***p*-Value <**	**D**	***p*-Value <**	**E**	***p*-Value <**	**AUC**
1	25.447	2.2×10^{-16}	25.264	2.2×10^{-16}	116,428	2.2×10^{-16}	0.75207	21.279	2.2×10^{-16}	20.926	2.2×10^{-16}	103,956	2.2×10^{-16}	0.76975
2	22.362	2.2×10^{-16}	22.913	2.2×10^{-16}	88,130	2.2×10^{-16}	0.80265	17.184	2.2×10^{-16}	17.45	2.2×10^{-16}	89,472	2.2×10^{-16}	0.79564
3	20.371	2.2×10^{-16}	20.187	2.2×10^{-16}	92,402	2.2×10^{-16}	0.79502	22.68	2.2×10^{-16}	22.463	2.2×10^{-16}	105,502	2.2×10^{-16}	0.76698
4	19.357	2.2×10^{-16}	18.697	2.2×10^{-16}	84,746	2.2×10^{-16}	0.8087	22.86	2.2×10^{-16}	23.514	2.2×10^{-16}	107,350	2.2×10^{-16}	0.76368
5	19.574	2.2×10^{-16}	19.717	2.2×10^{-16}	87,790	2.2×10^{-16}	0.80326	19.466	2.2×10^{-16}	19.577	2.2×10^{-16}	85,634	2.2×10^{-16}	0.8025
6	19.589	2.2×10^{-16}	19.444	2.2×10^{-16}	86,792	2.2×10^{-16}	0.80504	19.242	2.2×10^{-16}	18.842	2.2×10^{-16}	85,456	2.2×10^{-16}	0.80282
7	19.501	2.2×10^{-16}	20.113	2.2×10^{-16}	86,834	2.2×10^{-16}	0.80497	19.386	2.2×10^{-16}	19.155	2.2×10^{-16}	86,260	2.2×10^{-16}	0.80138
8	19.554	2.2×10^{-16}	19.13	2.2×10^{-16}	86,842	2.2×10^{-16}	0.80496	19.361	2.2×10^{-16}	19.45	2.2×10^{-16}	86,204	2.2×10^{-16}	0.80148
9	19.532	2.2×10^{-16}	19.15	2.2×10^{-16}	86,844	2.2×10^{-16}	0.80495	19.341	2.2×10^{-16}	18.899	2.2×10^{-16}	86,280	2.2×10^{-16}	0.80134
10	19.511	2.2×10^{-16}	19.569	2.2×10^{-16}	86,824	2.2×10^{-16}	0.80499	19.352	2.2×10^{-16}	19.555	2.2×10^{-16}	86,256	2.2×10^{-16}	0.80139
11	19.521	2.2×10^{-16}	19.997	2.2×10^{-16}	86,818	2.2×10^{-16}	0.805	19.349	2.2×10^{-16}	19.403	2.2×10^{-16}	86,252	2.2×10^{-16}	0.80139
12	19.517	2.2×10^{-16}	19.707	2.2×10^{-16}	86,814	2.2×10^{-16}	0.80501	19.35	2.2×10^{-16}	18.871	2.2×10^{-16}	86,270	2.2×10^{-16}	0.80136
13	19.524	2.2×10^{-16}	20.022	2.2×10^{-16}	86,822	2.2×10^{-16}	0.80499	19.349	2.2×10^{-16}	19.742	2.2×10^{-16}	86,250	2.2×10^{-16}	0.8014
14	19.523	2.2×10^{-16}	19.428	2.2×10^{-16}	86,824	2.2×10^{-16}	0.80499	19.35	2.2×10^{-16}	19.257	2.2×10^{-16}	86,254	2.2×10^{-16}	0.80139
15	19.523	2.2×10^{-16}	19.449	2.2×10^{-16}	86,824	2.2×10^{-16}	0.80499	19.348	2.2×10^{-16}	19.831	2.2×10^{-16}	86,254	2.2×10^{-16}	0.80139
16	19.524	2.2×10^{-16}	19.354	2.2×10^{-16}	86,822	2.2×10^{-16}	0.80499	19.35	2.2×10^{-16}	19.033	2.2×10^{-16}	86,254	2.2×10^{-16}	0.80139
Mean	20.12063	2.2×10^{-16}	20.13381	2.2×10^{-16}	89,034.75	2.2×10^{-16}	0.801035	19.76544	2.2×10^{-16}	19.748	2.2×10^{-16}	89,994	2.2×10^{-16}	0.794705

Table A5. Performance measurements of inverse log-map pre-processing with Power-of-Two (Po2) quantization scheme.

Bit-Width	**Frequency, AUC = 0.9601852**							**Spatial, AUC = 0.9555699**						
	Z	***p*-Value <**	**D**	***p*-Value <**	**E**	***p*-Value <**	**AUC**	**Z**	***p*-Value <**	**D**	***p*-Value <**	**E**	***p*-Value <**	**AUC**
1	10.459	2.20×10^{-16}	10.252	2.20×10^{-16}	32,948	2.20×10^{-16}	0.90151	10.272	2.20×10^{-16}	10.455	2.20×10^{-16}	34,392	2.20×10^{-16}	0.89409
2	12.387	2.20×10^{-16}	12.347	2.20×10^{-16}	35,646	2.20×10^{-16}	0.89647	10.788	2.20×10^{-16}	10.76	2.20×10^{-16}	32,438	2.20×10^{-16}	0.89759
3	11.824	2.20×10^{-16}	11.667	2.20×10^{-16}	33,776	2.20×10^{-16}	0.89981	10.396	2.20×10^{-16}	10.596	2.20×10^{-16}	30,954	2.20×10^{-16}	0.90024
4	12.313	2.20×10^{-16}	12.229	2.20×10^{-16}	34,682	2.20×10^{-16}	0.89819	10.134	2.20×10^{-16}	10.34	2.20×10^{-16}	31,348	2.20×10^{-16}	0.89954
5	12.447	2.20×10^{-16}	12.339	2.20×10^{-16}	36,004	2.20×10^{-16}	0.89583	10.288	2.20×10^{-16}	10.255	2.20×10^{-16}	31,156	2.20×10^{-16}	0.89988
6	12.447	2.20×10^{-16}	12.526	2.20×10^{-16}	36,004	2.20×10^{-16}	0.89583	10.288	2.20×10^{-16}	10.625	2.20×10^{-16}	31,156	2.20×10^{-16}	0.89988
7	12.447	2.20×10^{-16}	12.32	2.20×10^{-16}	36,004	2.20×10^{-16}	0.89583	10.288	2.20×10^{-16}	10.288	2.20×10^{-16}	31,156	2.20×10^{-16}	0.89988
8	12.447	2.20×10^{-16}	12.501	2.20×10^{-16}	36,004	2.20×10^{-16}	0.89583	10.288	2.20×10^{-16}	10.325	2.20×10^{-16}	31,156	2.20×10^{-16}	0.89988
9	12.447	2.20×10^{-16}	12.128	2.20×10^{-16}	36,004	2.20×10^{-16}	0.89583	10.288	2.20×10^{-16}	10.421	2.20×10^{-16}	31,156	2.20×10^{-16}	0.89988
10	12.447	2.20×10^{-16}	12.447	2.20×10^{-16}	36,004	2.20×10^{-16}	0.89583	10.288	2.20×10^{-16}	10.31	2.20×10^{-16}	31,156	2.20×10^{-16}	0.89988
11	12.447	2.20×10^{-16}	12.514	2.20×10^{-16}	36,004	2.20×10^{-16}	0.89583	10.288	2.20×10^{-16}	10.088	2.20×10^{-16}	31,156	2.20×10^{-16}	0.899878
12	12.447	2.20×10^{-16}	12.886	2.20×10^{-16}	36,004	2.20×10^{-16}	0.89583	10.288	2.20×10^{-16}	10.367	2.20×10^{-16}	31,156	2.20×10^{-17}	0.89988
13	12.447	2.20×10^{-16}	12.793	2.20×10^{-16}	36,004	2.20×10^{-16}	0.89583	10.288	2.20×10^{-16}	10.008	2.20×10^{-16}	31,156	2.20×10^{-18}	0.89988
14	12.447	2.20×10^{-16}	12.634	2.20×10^{-16}	36,004	2.20×10^{-16}	0.89583	10.288	2.20×10^{-16}	10.069	2.20×10^{-16}	31,156	2.20×10^{-19}	0.89988
15	12.447	2.20×10^{-16}	12.501	2.20×10^{-16}	36,004	2.20×10^{-16}	0.89583	10.288	2.20×10^{-16}	10.538	2.20×10^{-16}	31,156	2.20×10^{-20}	0.89988
16	12.447	2.20×10^{-16}	12.731	2.20×10^{-16}	36,004	2.20×10^{-16}	0.89583	10.288	2.20×10^{-16}	10.458	2.20×10^{-16}	31,156	2.20×10^{-21}	0.89988
Mean	12.27169	2.2×10^{-16}	12.30094	2.2×10^{-16}	35,568.75	2.2×10^{-16}	0.896619	10.31538	2.2×10^{-16}	10.36894	2.2×10^{-16}	31,437.75	1.52778×10^{-16}	0.899375

Table A6. Performance measurements of inverse log-map pre-processing with Dynamic-Fixed-Point (DFP) quantization scheme.

Bit-Width	**Frequency, AUC = 0.9601852**							**Spatial, AUC = 0.9555699**						
	Z	***p*-Value <**	**D**	***p*-Value <**	**E**	***p*-Value <**	**AUC**	**Z**	***p*-Value <**	**D**	***p*-Value <**	**E**	***p*-Value <**	**AUC**
1	10.459	2.2×10^{-16}	10.465	2.2×10^{-16}	32,948	2.20×10^{-16}	0.90151	9.3355	2.2×10^{-16}	9.3886	2.2×10^{-16}	31,530	2.20×10^{-16}	0.900483
2	12.387	2.2×10^{-16}	12.36	2.2×10^{-16}	35,646	2.20×10^{-16}	0.89647	11.028	2.2×10^{-16}	10.887	2.2×10^{-16}	32,748	2.20×10^{-16}	0.897033
3	12.176	2.2×10^{-16}	12.541	2.2×10^{-16}	34,758	2.20×10^{-16}	0.89806	9.7993	2.2×10^{-16}	9.548	2.2×10^{-16}	29,714	2.20×10^{-16}	0.902613
4	11.198	2.2×10^{-16}	11.251	2.2×10^{-16}	32,150	2.20×10^{-16}	0.90272	10.378	2.2×10^{-16}	10.258	2.2×10^{-16}	31,416	2.20×10^{-16}	0.899414
5	12.397	2.2×10^{-16}	12.222	2.2×10^{-16}	36,440	2.20×10^{-16}	0.89505	9.7374	2.2×10^{-16}	9.6062	2.2×10^{-16}	29,964	2.20×10^{-16}	0.902131
6	12.41	2.2×10^{-16}	12.462	2.2×10^{-16}	35,888	2.20×10^{-16}	0.89604	9.7943	2.2×10^{-16}	9.6758	2.2×10^{-16}	29,980	2.20×10^{-16}	0.902088
7	12.128	2.2×10^{-16}	11.943	2.2×10^{-16}	34,146	2.20×10^{-16}	0.89915	10.011	2.2×10^{-16}	10.015	2.2×10^{-16}	30,208	2.20×10^{-16}	0.901573
8	12.689	2.2×10^{-16}	12.435	2.2×10^{-16}	37,076	2.20×10^{-16}	0.89391	10.179	2.2×10^{-16}	10.241	2.2×10^{-16}	31,246	2.20×10^{-16}	0.899718
9	12.677	2.2×10^{-16}	12.947	2.2×10^{-16}	36,890	2.20×10^{-16}	0.89424	10.071	2.2×10^{-16}	9.8498	2.2×10^{-16}	30,332	2.20×10^{-16}	0.901351
10	12.769	2.2×10^{-16}	12.823	2.2×10^{-16}	37,116	2.20×10^{-16}	0.89384	10.144	2.2×10^{-16}	9.6202	2.2×10^{-16}	30,474	2.20×10^{-16}	0.901098
11	12.686	2.2×10^{-16}	12.723	2.2×10^{-16}	35,874	2.20×10^{-16}	0.89606	10.106	2.2×10^{-16}	9.9334	2.2×10^{-16}	30,412	2.20×10^{-16}	0.901208
12	12.689	2.2×10^{-16}	12.54	2.2×10^{-16}	35,888	2.20×10^{-16}	0.89604	10.112	2.2×10^{-16}	9.9988	2.2×10^{-16}	30,416	2.20×10^{-16}	0.901201
13	12.692	2.2×10^{-16}	12.417	2.2×10^{-16}	35,884	2.20×10^{-16}	0.89604	10.107	2.2×10^{-16}	10.105	2.2×10^{-16}	30,406	2.20×10^{-16}	0.901219
14	12.689	2.2×10^{-16}	12.831	2.2×10^{-16}	35,872	2.20×10^{-16}	0.89606	10.107	2.2×10^{-16}	10.377	2.2×10^{-16}	30,410	2.20×10^{-16}	0.901212
15	12.691	2.2×10^{-16}	12.605	2.2×10^{-16}	35,876	2.20×10^{-16}	0.89606	10.107	2.2×10^{-16}	9.6937	2.2×10^{-16}	30,410	2.20×10^{-16}	0.901212
16	12.689	2.2×10^{-16}	13.224	2.2×10^{-16}	35,872	2.20×10^{-16}	0.89606	10.107	2.2×10^{-16}	10.28	2.2×10^{-16}	30,408	2.20×10^{-16}	0.901216
Mean	12.33913	2.2×10^{-16}	12.36181	2.2×10^{-16}	35,520.25	2.2×10^{-16}	0.896706	10.07022	2.2×10^{-16}	9.967344	2.2×10^{-16}	30,629.63	2.2×10^{-16}	0.900923

Table A7. Performance measurements of retraining with Power-of-Two (Po2) quantization scheme.

Bit-Width	**Spatial, AUC = 0.9555699**						
	Z	***p*-Value <**	**D**	***p*-Value <**	**E**	***p*-Value <**	**AUC**
1	−1.484	0.1378	−1.5176	0.1291	11,484	2.20×10^{-16}	0.9614257
2	−1.3442	0.1789	−1.3351	0.1819	4444	0.0055	0.9589876
3	−1.4954	0.1348	−1.4923	0.1356	3618	0.028	0.9592128
4	−2.4204	0.0155	−2.4541	0.01412	4054	0.005	0.961272
5	−3.6926	0.000222	−3.7461	0.00018	5778	2.20×10^{-16}	0.9645181
6	−2.2286	0.02584	−2.2825	0.02246	3724	0.009	0.9606535
7	−3.2514	0.00115	−3.2259	0.00126	5118	2.20×10^{-16}	0.9633884
8	−1.9956	0.04598	−2.0536	0.04001	3674	0.0155	0.9602138
9	−1.9956	0.04598	−2.003	0.04518	3674	0.0165	0.9602138
10	−1.9956	0.04598	−2.0175	0.04364	3674	0.0155	0.9602138
11	−1.9956	0.04598	−2.0453	0.04082	3674	0.009	0.9602138
12	−1.9956	0.04598	−1.9694	0.0489	3674	0.016	0.9602138
13	−1.9956	0.04598	−1.9812	0.04757	3674	0.014	0.9602138
14	−1.9966	0.04587	−1.9966	0.04587	3674	0.01	0.9602138
15	−1.9956	0.04598	−1.9794	0.04777	3674	0.0125	0.9602138
16	−1.9956	0.04598	−2.0171	0.04368	3674	0.0115	0.9602138
Mean	−2.11738	0.056745	−2.13229	0.055503	4455.375	0.0105	0.960711394

Table A8. Performance measurements of retraining with Dynamic-Fixed-Point (DFP) quantization scheme.

Bit-Width	**Spatial, AUC = 0.9555699**						
	Z	***p*-Value <**	**D**	***p*-Value <**	**E**	***p*-Value <**	**AUC**
1	−0.84641	0.3973	−0.84277	0.3994	11,650	2.20×10^{-16}	0.95901
2	−1.6187	0.1055	−1.6063	0.1082	5118	2.20×10^{-16}	0.95978
3	−1.6091	0.1076	−1.607	0.1081	3542	0.007	0.95882
4	0.17147	0.8639	0.17203	0.8634	2644	0.0645	0.95524
5	2.6438	0.008199	2.6344	0.008428	3636	0.0095	0.94999
6	4.2815	1.86×10^{-5}	4.1887	2.81×10^{-5}	10,824	2.20×10^{-16}	0.93636
7	4.2447	2.19×10^{-5}	4.2655	2.00×10^{-5}	10,374	2.20×10^{-16}	0.93725
8	−0.5619	0.5742	−0.5581	0.5768	2694	0.024	0.9566
9	−0.3745	0.708	−0.3739	0.7085	2442	0.053	0.95627
10	−0.3324	0.7396	−0.3411	0.733	2492	0.054	0.95619
11	−0.3081	0.758	−0.3148	0.7529	2474	0.0545	0.95615
12	−0.3238	0.7461	−0.3141	0.7535	2490	0.0535	0.95617
13	−0.3181	0.7504	−0.3001	0.7641	2480	0.051	0.95616
14	−0.3239	0.746	−0.3152	0.7526	2474	0.054	0.95617
15	−0.3181	0.7504	−0.3222	0.7473	2480	0.0435	0.95616
16	−0.322	0.7475	−0.3166	0.7516	2476	0.0545	0.95617
Mean	0.255282	0.5001712	0.253044	0.5017423	4393.125	0.0326875	0.953907

References

1. Gardezi, A.; Malik, U.; Rehman, S.; Young, R.C.D.; Birch, P.M.; Chatwin, C.R. Enhanced target recognition employing spatial correlation filters and affine scale invariant feature transform. In *Pattern Recognition and Tracking XXX*; Alam, M.S., Ed.; International Society for Optics and Photonics, SPIE: Baltimore, MD, USA, 2019; Volume 10995, pp. 145–160. [CrossRef]
2. Gardezi, A.; Qureshi, T.; Alkandri, A.; Young, R.C.D.; Birch, P.M.; Chatwin, C.R. Comparison of spatial domain optimal trade-off maximum average correlation height (OT-MACH) filter with scale invariant feature transform (SIFT) using images with poor contrast and large illumination gradient. In *Optical Pattern Recognition XXVI*; Casasent, D.; Alam, M.S., Eds.; International Society for Optics and Photonics, SPIE: Baltimore, MD, USA, 2015; Volume 9477, pp. 21–35. [CrossRef]

3. P.D., S.M.; Lin, J.; Zhu, S.; Yin, Y.; Liu, X.; Huang, X.; Song, C.; Zhang, W.; Yan, M.; Yu, Z.; Yu, H. A Scalable Network-on-Chip Microprocessor With 2.5D Integrated Memory and Accelerator. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2017**, *64*, 1432–1443. [CrossRef]
4. Awan, A.B.; Bakhshi, A.D.; Abbas, M.; Rehman, S. Active contour-based clutter defiance scheme for correlation filters. *Electron. Lett.* **2019**, *55*, 525–527. [CrossRef]
5. Kay, S. *Fundamentals of Statistical Signal Processing: Detection Theory*; PTR Prentice-Hall: Upper Saddle River, NJ, USA, 1993.
6. Kumar, B.V.K.V.; Pochapsky, E. Signal-to-noise ratio considerations in modified matched spatial filters. *J. Opt. Soc. Am. A* **1986**, *3*, 777–786. [CrossRef]
7. Lugt, A.V. Signal detection by complex spatial filtering. *IEEE Trans. Inf. Theory* **1964**, *10*, 139–145. [CrossRef]
8. Kumar, B.; Juday, R.; Mahalanobis, A. *Correlation Pattern Recognition*; Cambridge University Press: Cambridge, UK, 2014.
9. Psaltis, D.; Casasent, D. Position, Rotation, And Scale Invariant Optical Correlation. *Appl. Opt.* **1976**, *15*, 1795–1799. [CrossRef]
10. Mersereau, K.; Morris, G.M. Scale, rotation, and shift invariant image recognition. *Appl. Opt.* **1986**, *25*, 2338–2342. [CrossRef] [PubMed]
11. Casasent, D. Unified synthetic discriminant function computational formulation. *Appl. Opt.* **1984**, *23*, 1620–1627. [CrossRef] [PubMed]
12. Kumar, B.V.K.V. Minimum-variance synthetic discriminant functions. *J. Opt. Soc. Am. A* **1986**, *3*, 1579–1584. doi:10.1364/JOSAA.3.001579. [CrossRef]
13. Bahri, Z.; Kumar, B.V.K.V. Generalized synthetic discriminant functions. *J. Opt. Soc. Am. A* **1988**, *5*, 562–571. [CrossRef]
14. Casasent, D.; Chang, W.T. Correlation synthetic discriminant functions. *Appl. Opt.* **1986**, *25*, 2343–2350. [CrossRef]
15. Mahalanobis, A.; Casasent, D.P. Performance evaluation of minimum average correlation energy filters. *Appl. Opt.* **1991**, *30*, 561–572. [CrossRef] [PubMed]
16. Mahalanobis, A.; Kumar, B.V.K.V.; Casasent, D. Minimum average correlation energy filters. *Appl. Opt.* **1987**, *26*, 3633–3640. [CrossRef] [PubMed]
17. Kumar, B.V.K.V. Tutorial survey of composite filter designs for optical correlators. *Appl. Opt.* **1992**, *31*, 4773–4801. [CrossRef] [PubMed]
18. Sudharsanan, S.I.; Mahalanobis, A.; Sundareshan, M.K. Unified framework for the synthesis of synthetic discriminant functions with reduced noise variance and sharp correlation structure. *Opt. Eng.* **1990**, *29*, 1021–1028. [CrossRef]
19. Alkanhal, M.; Vijaya Kumar, B.V.K.; Mahalanobis, A. Improving the false alarm capabilities of the maximum average correlation height correlation filter. *Opt. Eng.* **2000**, *39*, 1133–1141. [CrossRef]
20. Alkanhal, M.; Vijaya Kumar, B.V.K.; Mahalanobis, A. Eigen-extended maximum average correlation height (EEMACH) filters for automatic target recognition. *Automat. Target Recogn.* **2001**. [CrossRef]
21. Goyal, S.; Nishchal, N.K.; Beri, V.K.; Gupta, A.K. Wavelet-modified maximum average correlation height filter for rotation invariance that uses chirp encoding in a hybrid digital-optical correlator. *Appl. Opt.* **2006**, *45*, 4850–4857. [CrossRef]
22. Goyal, S.; Nishchal, N.K.; Beri, V.K.; Gupta, A.K. Wavelet-modified maximum average correlation height filter for out-of-plane rotation invariance. *Optik* **2009**, *120*, 62–67. [CrossRef]
23. Ang, T.; Tan, A.W.I.; Loo, C.K.; Wong, W.K. Wavelet MACH Filter for Omnidirectional Human Activity Recognition. *Int. J. Innov. Comput. Inf. Control IJICIC* **2012**, *8*, 3565–3584.
24. Rodriguez, A.; Boddeti, V.N.; Kumar, B.V.K.V.; Mahalanobis, A. Maximum Margin Correlation Filter: A New Approach for Localization and Classification. *IEEE Trans. Image Process.* **2013**, *22*, 631–643. [CrossRef]
25. Fernandez, J.A.; Vijaya Kumar, B.V.K. Partial-Aliasing Correlation Filters. *IEEE Trans. Signal Process.* **2015**, *63*, 921–934. [CrossRef]
26. Kiani, H.; Sim, T.; Lucey, S. Multi-channel correlation filte rs for human action recognition. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 1485–1489. [CrossRef]
27. Tehsin, S.; Rehman, S.; Bilal, A.; Chaudry, Q.; Saeed, O.; Abbas, M.; Young, R. Comparative analysis of zero aliasing logarithmic mapped optimal trade-off correlation filter. In *Pattern Recognition and Tracking XXVIII*; Alam, M.S., Ed.; International Society for Optics and Photonics, SPIE: Anaheim, CA, USA, 2017; Volume 10203, pp. 22–37. [CrossRef]
28. Tehsin, S.; Rehman, S.; Riaz, F.; Saeed, O.; Hassan, A.; Khan, M.A.; Alam, M.S. Fully invariant wavelet enhanced minimum average correlation energy filter for object recognition in cluttered and occluded environments. In *Defense + Security*; SPIE: Anaheim, CA, USA, 2017.
29. Tehsin, S.; Rehman, S.; Saeed, M.O.B.; Riaz, F.; Hassan, A.; Abbas, M.; Young, R.; Alam, M.S. Self-Organizing Hierarchical Particle Swarm Optimization of Correlation Filters for Object Recognition. *IEEE Access* **2017**, *5*, 24495–24502. [CrossRef]
30. Achuthanunni, A.; Kishore Saha, R.; Banerjee, P.K. Unconstrained Band-pass Optimized Correlation Filter (UBoCF): An application to face recognition. In Proceedings of the 2018 IEEE Applied Signal Processing Conference (ASPCON), Kolkata, India, 7–9 December 2018; pp. 297–303. [CrossRef]
31. Banerjee, P.K.; Datta, A.K. Band-pass correlation filter for illumination- and noise-tolerant face recognition. *SIViP* **2017**, 9–16. [CrossRef]
32. Akbar, N.; Tehsin, S.; Bilal, A.; Rubab, S.; Rehman, S.; Young, R. Detection of moving human using optimized correlation filters in homogeneous environments. In *Pattern Recognition and Tracking XXXI*; Alam, M.S., Ed.; International Society for Optics and Photonics, SPIE: Baltimore, MD, USA, 2020; Volume 11400, pp. 73–79. [CrossRef]

33. Akbar, N.; Tehsin, S.; ur Rehman, H.; Rehman, S.; Young, R. Hardware design of correlation filters for target detection . In *Pattern Recognition and Tracking XXX*; Alam, M.S., Ed. International Society for Optics and Photonics, SPIE: Baltimore, MD, United States, 2019; Volume 10995, pp. 71–79. [CrossRef]
34. Masood, H.; Rehman, S.; Khan, M.; Javed, Q.; Abbas, M.; Alam, M.; Young, R. Approximate Proximal Gradient-Based Correlation Filter for Target Tracking in Videos: A Unified Approach. *Arab. J. Sci. Eng.* **2019**, 9363–9380. [CrossRef]
35. Masood, H.; Rehman, S.; Khan, M.; Javed, Q.; Abbas, M.; Alam, M.; Young, R. A novel technique for recognition and tracking of moving objects based on E-MACH and proximate gradient (PG) filters. In Proceedings of the 2017 20th International Conference of Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 22–24 December 2017; pp. 1–6. [CrossRef]
36. Venkataramani, S.; Ranjan, A.; Roy, K.; Raghunathan, A. AxNN: Energy-efficient neuromorphic systems using approximate computing. In Proceedings of the 2014 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), La Jolla, CA, USA, 11–13 August 2014; pp. 27–32.
37. Rubio-González, C.; Cuong Nguyen.; Hong Diep Nguyen.; Demmel, J.; Kahan, W.; Sen, K.; Bailey, D.H.; Iancu, C.; Hough, D. Precimonious: Tuning assistant for floating-point precision. In Proceedings of the SC '13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, Denver, CO, USA, 17–22 November 2013; pp. 1–12.
38. Pandey, J.G.; Karmakar, A.; Shekhar, C.; Gurunarayanan, S. An FPGA-based fixed-point architecture for binary logarithmic computation. In Proceedings of the 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013), Shimla, India, 9–11 December 2013; pp. 383–388.
39. Mahalanobis, A.; Kumar, B.V. Optimality of the maximum average correlation height filter for detection of targets in noise. *Opt. Eng.* **1997**, *36*, 2642–2648. [CrossRef]
40. Mahalanobis, A.; Kumar, B.V.K.V.; Song, S.; Sims, S.R.F.; Epperson, J.F. Unconstrained correlation filters. *Appl. Opt.* **1994**, *33*, 3751–3759. [CrossRef] [PubMed]
41. Sabir, D.; Rehman, S.; Hassan, A. Fully invariant quaternion based filter for target recognition. In Proceedings of the 2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, Malaysia, 19–21 October 2015; pp. 509–513. [CrossRef]
42. Gardezi, A.; Alkandri, A.; Birch, P.; Young, R.; Chatwin, C. A space variant maximum average correlation height (MACH) filter for object recognition in real time thermal images for security applications. In *Optics and Photonics for Counterterrorism and Crime Fighting VI and Optical Materials in Defence Systems Technology VII*; Lewis, C., Burgess, D., Zamboni, R., Kajzar, F., Heckman, E.M., Eds.; International Society for Optics and Photonics, SPIE: Toulouse, France, 2010; Volume 7838, pp. 191–204. [CrossRef]
43. Gardezi, A.; Al-Kandri, A.; Birch, P.; Young, R.; Chatwin, C. Enhancement of the speed of space-variant correlation filter implementations by using low-pass pre-filtering for kernel placement and applications to real-time security monitoring. In *Optical Pattern Recognition XXII*; Casasent, D.P., Chao, T.H., Eds.; International Society for Optics and Photonics, SPIE: Orlando, FL, USA, 2011; Volume 8055, pp. 78–88. [CrossRef]
44. Awan, A.B. Composite filtering strategy for improving distortion invariance in object recognition. *IET Image Process.* **2018**, *12*, 1499–1509. [CrossRef]
45. Zhou, A.; Yao, A.; Guo, Y.; Xu, L.; Chen, Y. Incremental Network Quantization: Towards Lossless CNNs with Low-Precision Weights. *CoRR* **2017**, *abs/1702.03044*. Available online: http://xxx.lanl.gov/abs/1702.03044 (accessed on 25 August 2017).
46. Gysel, P. Ristretto: Hardware-Oriented Approximation of Convolutional Neural Networks. *CoRR* **2016**, *abs/1605.06402*. Available online: http://xxx.lanl.gov/abs/1605.06402 (accessed on 20 May 2016).
47. Hubara, I.; Courbariaux, M.; Soudry, D.; El-Yaniv, R.; Bengio, Y. Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations. *CoRR* **2016**, *abs/1609.07061*. Available online: http://xxx.lanl.gov/abs/1609.07061 (accessed on 22 September 2016).
48. Viksten, F.; Forssén, P.E.; Johansson, B.; Moe, A. Comparison of Local Image Descriptors for Full 6 Degree-of-Freedom Pose Estimation. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009.
49. Venkatraman, E. A permutation test to compare receiver operating characteristic curves. *Biometrics* **2000**, *56*, 1134–1138. [CrossRef] [PubMed]
50. Venkatraman, E.S.; Begg, C.B. A distribution-free procedure for comparing receiver operating characteristic curves from a paired experiment. *Biometrika* **1996**, *83*, 835–848. Available online: http://xxx.lanl.gov/abs/https://academic.oup.com/biomet/article-pdf/83/4/835/703326/83-4-835.pdf (accessed on 1 December 1996). [CrossRef]
51. Hanley, J.A.; McNeil, B.J. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology* **1983**, *148*, 839–843.10.1148/radiology.148.3.6878708. Available online: http://xxx.lanl.gov/abs/https://doi.org/10.1148/radiology.148.3.6878708 (accessed on 1 September 1983). [CrossRef] [PubMed]
52. DeLong, E.R.; DeLong, D.M.; Clarke-Pearson, D.L. Comparing the Areas under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach. *Biometrics* **1988**, *44*, 837–845. [CrossRef]