Hardware-Aware Neural Feature Extraction for Resource-Constrained Devices

Francesco Tosini[1] Simone Pedroni[1] Christian Veronesi[1] Pietro Bartoli[1]
Andrea Giudici[1] Marco Paracchini[1] Marco Marcon[1]
Diana Trojaniello[2]
[1]Department of Electronics, Information and Bioengineering (DEIB), Politecnico di Milano, Italy
[2]Smart Eyewear Lab, EssilorLuxottica, Milan, Italy
francesco.tosini@polimi.it

Abstract

Visual SLAM is a core component of spatial computing systems, yet deploying learned local feature extractors on microcontroller-class hardware remains challenging due to memory, bandwidth, and quantization constraints. While modern neural descriptors provide strong robustness, their practical adoption is often hindered by system-level bottlenecks that are not captured by FLOP-based efficiency metrics. In this work, we introduce Gideon, a hardware-aware neural feature extractor explicitly designed for resource-constrained devices. Our approach combines relational knowledge distillation from a SuperPoint teacher with differentiable neural architecture search (DNAS) under strict memory and operator constraints. Unlike conventional design pipelines, we treat quantization stability and dynamic-range compactness as first-class objectives. We show that architectural choices such as replacing Batch Normalization with Affine layers significantly improve INT8 robustness, and that descriptor dimensionality directly governs quantization resilience. Deployed on STM32N6, Gideon achieves 9.003 ms inference time (111 fps) while remaining below 1.5 MB memory footprint. Remarkably, INT8 quantization induces negligible degradation and occasionally matches full-precision performance. These results demonstrate that robust learned feature extraction can be reconciled with embedded hardware constraints through holistic hardware–algorithm co-design.

1. Introduction

Despite living in an era of unprecedented technological advancement, our interface with the digital world remains paradoxically primitive. In fact, we are still relying on small 2D rectangles to experience the virtual world, which causes a significant amount of friction. Similarly to the transition from command-line interfaces to graphical user interfaces, we are experiencing a paradigm shift led by the introduction of Spatial Computing. Smart Eyewear, in particular, stands at the forefront of this revolution with its promise to weave the virtual domain directly into our physical perception. However, this requires solving several engineering challenges, including the need for robust, real-time machine perception on a device with extreme thermal, energetic, and usability constraints.

Visual Simultaneous Localization and Mapping (SLAM) is at the core of the Smart Glasses' perception stack: it provides the world knowledge required to anchor virtual content to the real world, understanding ego-motion and the 3D structure of the environment.

Modern SLAM algorithms, such as ORBSLAM-3 [2], rely heavily on the extraction of visual features, i.e. distinctive keypoints and their associated descriptors, which serve as the fundamental "anchors" for tracking. Ideally, these features must be repeatable across drastic changes in viewpoint and illumination, yet cheap enough to compute at high frame rates.

The current State of the Art presents a sharp dichotomy. On one hand, traditional "hand-crafted" feature extractors (such as ORB [25], FAST [24], or BRISK [14]) are computationally efficient and have served as the backbone of SLAM for years. Yet, they struggle to stay reliable in challenging scenarios, such as texture-less environments or with dynamic lighting. On the other hand, deep learning models such as SuperPoint [4] and R2D2 [22], as well as methods such as D2-Net [5] and DISK [29], managed to reframe the feature extraction process as a fully differentiable problem, producing "learned" features that offer unprecedented robustness and repeatability. However, this performance comes at a prohibitive cost on wearable devices: these models typically rely on heavy backbones that far exceed the budget of low-power edge neural processing units (NPUs), which becomes especially true for newer transformer-based approaches, such as LoFTR [27] and LightGlue [15]. In fact, deploying a full-scale feature extraction model on

Smart Eyewear would result in unacceptable latency and rapid battery depletion.

The State of the Art offers lightweight architectures such as XFeat-Micro [21] and ALIKED-Tiny [30]. However, despite their theoretical efficiency, these models are often not well-suited for microcontroller-class devices due to system-level bottlenecks not captured by FLOP-based metrics. In particular, semi-dense or high-dimensional descriptors introduce significant data movement overhead, saturating memory bandwidth, while operations such as dilated or deformable convolutions lead to irregular memory access patterns that degrade cache locality. As a result, these architectures, although effective on edge GPUs and mobile NPUs, remain difficult to deploy on MCUs without substantial architectural modifications.

Gideon circumvents these architectural pitfalls through strict hardware-algorithm co-design. By favoring a topology comprised of standard convolutional patterns, the model ensures contiguous memory access and a highly predictable data flow.

This design choice maximizes cache hit rates and eliminates the overhead associated with sparse operations. Consequently, Gideon keeps its entire execution footprint (both weights and activations) strictly under the SRAM threshold, allowing it to fully leverage the NPU's acceleration capabilities without triggering bandwidth or memory bottlenecks.

Beyond lightweight feature extractors, recent works have demonstrated fully onboard SLAM and visual–inertial odometry on highly resource-constrained platforms. For instance, NanoSLAM [20] enables complete SLAM pipelines on ultra-low-power processors, while LEVIO [6] proposes a lightweight visual–inertial odometry system tailored for embedded devices. These efforts highlight the feasibility of real-time perception under strict power and memory budgets.

However, such systems typically rely on handcrafted or fixed feature extraction modules and do not explicitly address quantization-aware learning of local features under microcontroller-class constraints. In contrast, our work focuses on the hardware-aware learning of compact and quantization-stable feature representations, which can serve as drop-in components within broader SLAM pipelines and enable distributed edge-SLAM paradigms where feature extraction runs on constrained endpoints. While full SLAM integration is beyond the scope of this work, we report feature-level metrics such as repeatability and matching correctness, which are widely accepted indicators of downstream SLAM performance.

In this work, we bridge this gap between the robustness of learned features and the efficiency of hand-crafted ones with Gideon, a novel training paradigm and neural architecture specifically designed for the intrinsic limitations of wearable devices.

Rather than training from scratch, our approach relies on Knowledge Distillation to transfer the robust "intuition" of a SuperPoint teacher into a lightweight student. Instead of manually designing the network, which is expensive and error-prone, we employ Differentiable Network Architecture Search (DNAS) to automatically identify an efficient topology that balances accuracy and cost. This approach accounts for the tight constraints of wearable devices, allowing us to deploy robust neural perception at the edge.

Unlike prior approaches that optimize network architecture, distillation, and deployment independently, we explicitly formulate feature extraction as a constrained system-level problem. In our setting, memory footprint, dataflow regularity, and quantization stability are treated as primary optimization objectives during training, rather than post-hoc constraints.

2. Methodology

In this section, we describe the design and training strategy underlying *Gideon*. Rather than treating architecture, distillation, and deployment as independent components, we adopt a unified hardware-aware perspective in which model topology, loss formulation, and optimization are jointly shaped by embedded system constraints.

2.1. Hardware-Aware Design Principles

The design of *Gideon* is driven by system-level constraints rather than purely architectural preferences. On microcontroller-class devices, latency and robustness are dominated not only by model size or FLOPs, but by memory footprint, data movement, and quantization stability.

First, the limited on-chip SRAM budget constrains both weights and intermediate activations, requiring compact feature maps and strictly bounded descriptor dimensionality to ensure full in-memory execution. Second, irregular memory access patterns can severely degrade throughput on embedded NPUs; therefore, we restrict the architecture to standard convolutional operators with predictable and contiguous dataflow. Third, INT8 inference imposes tight constraints on activation dynamic range, making quantization stability a primary design objective rather than a post-training adjustment.

These considerations redefine the optimization target from raw representational capacity to *system-level density*, where robustness, dynamic-range compactness, and hardware compatibility are jointly optimized.

2.2. Topological Structure

Guided by the hardware constraints outlined in Section 2.1, we established a baseline functional architecture represented in Fig. 1, composed of a single shared encoder and two task-specific decoder heads similarly to the original SuperPoint.

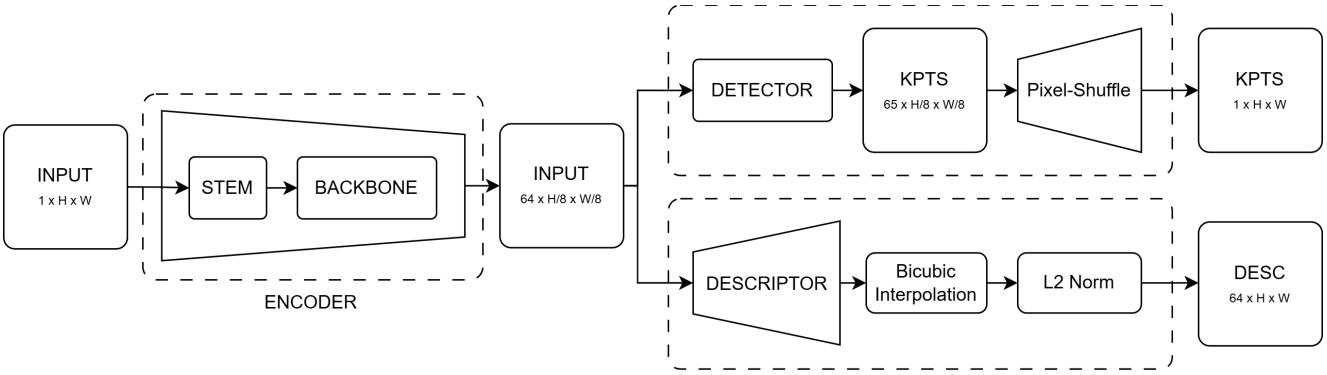

Figure 1. Overview of the baseline functional topology, inspired by the original SuperPoint [4].

The shared encoder is made up of a fixed stem and a searchable architectural block, which is dynamically determined by the DNAS pipeline. This improves search efficiency and results in better models. This bipartite encoder processes the input image to extract a dense latent representation, which is then fed into two parallel branches: a detector head, which computes a full-resolution spatial probability map of interest points through pixel-shuffling, and a descriptor head, which generates a semi-dense map of L2-normalized, high-dimensional feature vectors. This design allows the network to optimize both keypoint localization and description, while reducing the computational cost of the backbone.

2.3. Knowledge Distillation and NAS

Training a compact model from scratch under strict parameter and memory budgets often leads to sub-optimal feature representations. To overcome this limitation, we adopt a knowledge distillation strategy in which the student network, *Gideon*, is trained to mimic the activation manifold of a frozen SuperPoint teacher [8, 23]. Rather than rediscovering robust features from raw supervision alone, the student learns to map an already structured feature space, enabling efficient compression of the teacher's representational power.

However, distillation alone does not guarantee hardware efficiency. Given the stringent latency, memory, and quantization constraints of embedded NPUs, the student architecture itself must be explicitly optimized for deployment. Manual design under such constraints proved unstable and suboptimal, particularly for shallow networks operating near the edge of their capacity.

To jointly address representational fidelity and hardware compatibility, we introduce a Differentiable Neural Architecture Search (DNAS) stage inspired by DARTS [16]. During the search phase, a super-network ("SuperGideon") replaces each architectural block with a stochastic mixture of candidate operations (e.g., residual bottlenecks, Inception-like modules, standard convolutions). Both network weights and architectural parameters are optimized end-to-end.

The discrete operator selection is relaxed through the Gumbel-Softmax reparameterization [11, 18], enabling differentiable exploration of the search space. As the temperature is annealed, the architecture progressively converges toward a deterministic topology that satisfies hardware constraints while preserving the relational structure distilled from the teacher.

The search space includes standard convolutional blocks and Residual, Bottleneck and Inception-like modules with varying kernel sizes. Hardware constraints are implicitly enforced through architectural priors, rather than explicit latency terms, ensuring compatibility with MCU deployment.

2.4. Loss Design

Initially, training relied on MSE to match teacher outputs, which proved suboptimal as it constrains the student to replicate absolute activations rather than optimizing for downstream performance.

Furthermore, since the network performs detection and description in parallel, effectively weighting the individual contribution of each task is crucial. To ensure a stable and balanced training process, we employed uncertainty-based weighting to dynamically scale the respective loss functions.

2.4.1. Detection Loss

The number of background pixels vastly outnumbers the keypoints, which are ideally sparse peaks in the signal. As a result, standard losses such as Binary Cross-Entropy or Mean Squared Error tend to bias the model toward trivial background predictions and penalize near-miss activations as false positives.

To overcome these issues, we adopt a variant of the Focal Loss originally proposed in CornerNet. In addition to reweighting hard examples, we assign soft labels to the neighborhood of each keypoint using a 2D Gaussian kernel,

reducing the penalty for spatially close predictions while preserving strong gradients for true positives. This dual mechanism, spatial penalty reduction and dynamic confidence scaling, encourages the optimization process to focus on hard negatives and precise center localization, allowing the model to produce sharper, better-localized heatmaps while preventing the background signal from dominating the gradient.

Furthermore, SuperPoint typically operates with a notably low detection threshold (0.005), which often correlates with noisy predictions. In contrast, by anchoring our optimization to binarized ground truth locations, even while smoothing the loss surface with the aforementioned Gaussian penalty, the network is forced to suppress ambiguity. This has proven to yield significantly sharper activation peaks, resulting in higher detection confidence and accuracy.

To further benefit from this, we have introduced an Adaptive Thresholding mechanism that dynamically adjusts the detection sensitivity during training by computing the exponential moving average (EMA) of the average activation of the top pixels. This introduces a soft constraint on the mean number of keypoints and ensures that the extracted features are robust.

The implicit target number of keypoints is not fixed a priori, but emerges from the EMA dynamics of high-confidence activations. In practice, this results in stable keypoint densities across diverse scenes, including both indoor and outdoor environments.

2.4.2. Descriptor Loss

SuperPoint descriptors feature low variance, suggesting that most of their information is either redundant or unnecessary. Moreover, forcing a compact student network to perfectly replicate the absolute, high-dimensional metric space of the teacher often limits its representational flexibility and can negatively affect convergence.

To address this, we departed from standard Euclidean minimization, shifting the distillation objective from absolute feature matching to relational matching. Consequently, the student is trained to replicate the internal geometric relationships, i.e. the similarity manifold, of the teacher's feature space rather than attempting to align individual descriptors.

This works by computing a dense self-similarity matrix through the cosine similarity between descriptors across all spatial locations for both the student and the frozen teacher. These correlation matrices are then converted into probability distributions over the spatial dimensions using a temperature-scaled softmax operation. This yields a probabilistic map representing how every local feature relates to all other features within the image.

Finally, the student is optimized to match the teacher's relational distribution by minimizing the Kullback-Leibler (KL) Divergence [13]:

$$\mathcal{L}_{desc} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{KL}\left(\sigma\left(\frac{S_i^{gt}}{\tau}\right) \middle\| \sigma\left(\frac{S_i^{pred}}{\tau}\right) \right) \quad (1)$$

where S_i^{gt} and S_i^{pred} denote the dense self-similarity matrices for the i-th spatial location of the teacher and the student respectively, σ represents the softmax operation, τ is a temperature scaling parameter controlling the sharpness of the probability distribution, and N is the total number of spatial locations.

This approach allows the student to construct its own highly efficient embedding space, provided it faithfully preserves the same structural correlation topology as the teacher.

2.5. Training Procedure

2.5.1. Dataset and Data Augmentation

The proposed model is trained using 29000 images from the TUM-VI dataset [26] (23200 for training, 5800 for validation and test purposes). Input images are initially resized to 256×256 pixels and subsequently center-cropped to a spatial resolution of 192×256. To enhance the network's rotational and viewpoint invariance without incurring CPU bottlenecks, we employ dynamic, on-the-fly geometric data augmentations executed directly on the GPU. During each training iteration, inputs and their corresponding pseudo-ground truth targets undergo random spatial transformations, including $90°$, $180°$, and $270°$ rotations, as well as horizontal and vertical flips.

2.5.2. Optimization and Scheduling

The network parameters are optimized using the AdamW optimizer with an initial learning rate of 1×10^{-3} and a weight decay of 1×10^{-4}. To ensure training stability and mitigate the risk of exploding gradients, gradient clipping is enforced with a maximum norm of 5.0. The learning rate is dynamically adjusted using a *ReduceLROnPlateau* scheduler, which monitors the validation loss and reduces the learning rate by a factor of 0.5 if no improvement is observed for a patience of 5 epochs. The two loss functions described in Sections 2.4.1 and 2.4.2 are dynamically balanced via uncertainty weighting [12] during training, whereas they are simply summed for validation purposes.

2.6. Inference Protocol

During training, the network predictions are used directly without Non-Maximum Suppression (NMS), and losses are computed densely on pixel-shuffled logits and L_2-normalized descriptors. Teacher heatmaps are filtered with NMS (spatial radius $= 4$) and thresholded at 0.005.

At inference time, predicted heatmaps undergo spatial NMS ($r = 4$) followed by adaptive thresholding

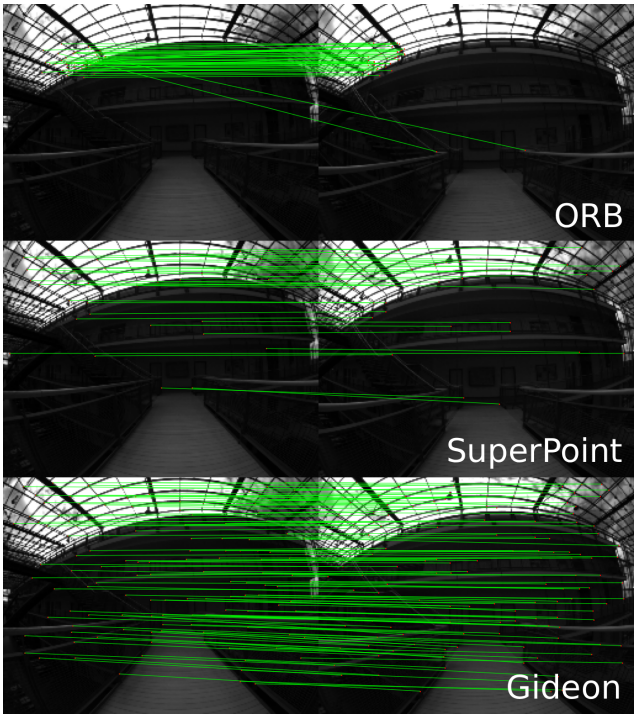

Figure 2. Qualitative results on TUM-VI. The green lines indicate predicted correspondences. The depicted image pair is not part of the training set.

(Section 2.4.1) to maintain stable keypoint density across scenes. Although descriptor learning relies on cosine similarity, matching is performed using Euclidean (L_2) distance, which is monotonically related for L_2-normalized vectors.

2.7. On-Device Deployment

To validate practical deployability, we perform end-to-end inference of *Gideon* on STM32N6 with its embedded NPU.

Deployment proceeds in two stages. First, functional correctness and toolchain compatibility are verified on the STM32N6 Developer Kit using the ST Edge AI runtime. Second, quantitative latency and energy measurements are obtained on a dedicated STM32N657A platform under fixed-frequency operation (800 MHz), representative of always-on scenarios.

Deployment is constrained by the 4.2 MB on-chip SRAM budget, which must accommodate weights, activations, and runtime buffers. We explicitly verify that the compiled network executes fully within internal memory without external transfers, minimizing bandwidth overhead and latency variability. This requirement directly motivates compact feature maps and regular memory access patterns.

3. Results and Discussion

We evaluate Gideon along three axes: embedded runtime, quantization robustness, and descriptor dimensionality. To evaluate the performance and robustness of our proposed architecture, we conducted validation on the widely adopted HPatches dataset [1], which is recognized as the standard benchmark for local feature evaluation.

HPatches provides a comprehensive suite of image sequences featuring severe illumination changes and drastic viewpoint variations. This environment perfectly mirrors the challenging conditions typically encountered in wearable SLAM applications and real-world hardware deployments.

As shown in Table 1, Gideon achieves 9.003 ms inference time (111.07 fps) on the STM32N6 microcontroller and remains strictly under 1.5 MB for weights (600.78 kB) and activations (827.25 kB).

On the other hand, the proposed model achieves solid performance on HPatches, with an average repeatability of approximately 0.52, an illumination correctness exceeding 91%, and a viewpoint correctness above 60%. It yields slightly lower metrics than SuperPoint, as expected in a distillation setup, but significantly surpasses the correctness of the standard ORB extractor.

Quantization induces a marginal degradation in keypoint localization (consistent with prior work on neural network quantization [10, 19]), yet offers an unexpected improvement in descriptor matching performance. The average correctness of the INT8 descriptors surpasses that of the full-precision baseline, rising from 91.9% to 93.7% for illumination sequences and from 59.3% to 61.4% for viewpoint sequences. Although low-precision discretization has been reported to exhibit regularization-like effects in certain settings [3, 7], these modest improvements may be partially attributed to statistical fluctuations. These improvements should be interpreted cautiously, as our training does not explicitly optimize for quantization. Further experiments would be needed to determine whether these effects are systematic.

As illustrated in Fig. 2, Gideon yields a rich set of geometrically coherent matches across the scene, qualitatively complementing the quantitative improvements reported in Section 3.

3.1. Quantization as a Design Constraint

Gideon's architecture features a rapidly down-sampling stem followed by a heterogeneous mix of Inception-like parallel paths and Residual connections. Instead of more classical Batch Normalization layers, the NAS pipeline heavily favoured Affine layers, where scale and bias are treated as learnable parameters.

We conducted an ablation study evaluating four distinct network configurations to systematically isolate the contri-

Table 1. Comparison of Feature Extractors on Embedded Hardware using the HPatches dataset. Best results are highlighted in bold, computed separately for full-precision (float32) and quantized (int8) models.

Model	Precision	Time (ms)	FPS	Weights size	Activation size	HPatches Performance			
						Rep (I)	Rep (V)	Cor (I)	Cor (V)
Gideon	float32	—	—	—	—	0.594	0.478	0.919	0.593
	int8	**9.003**	**111.07**	**600.77 KB**	**827.25 KB**	0.574	0.474	**0.937**	0.614
SuperPoint	float32	—	—	—	—	**0.723**	**0.692**	**0.954**	**0.681**
	int8	2108.604	0.47	1.24 MB	4.61 MB	**0.720**	**0.693**	**0.937**	**0.668**
ORB	—	—	—	—	—	0.658	0.639	0.407	0.332

butions of specific architectural components, particularly concerning their impact on quantization resilience. The BatchNorm + ReLU represents a more conventional feature extraction design that works as the standard baseline. The BatchNorm + PWL (introducing piecewise-linear activations) and the Affine + ReLU (i.e. Gideon) configurations (replacing Batch Normalization with Affine layers) serve to independently assess the impact of our proposed modifications. Finally, we present Affine + PWL, which integrates both modifications.

3.1.1. BatchNorm Collapse

The baseline Gideon architecture relying on BatchNorm and standard ReLU achieves the highest peak performance in full-precision. However, it suffers a near-total collapse when quantized to int8 as demonstrated in Table 2.

In fact, Batch Normalization produces a wide dynamic range of feature maps, significantly increasing the activation spread across channels. This severe cross-channel variance catastrophically amplifies quantization noise due to activation clipping, as a wider continuous range must be mapped onto a fixed, low-resolution discrete grid.

3.1.2. Improving Quantization Resilience

The proposed design works around a fundamental architectural trade-off between representational capacity and quantization readiness. In fact, Gideon sacrifices a marginal fraction of theoretical peak accuracy in the continuous domain to ensure stability in the discrete domain which has proven to be a necessary and highly advantageous compromise for edge deployment.

Similarly, the use of piecewise linear activation functions, such as Hardsigmoid and Hardtanh, improves the expressive power by increasing the information density in the activations, leading to slightly better quantization resilience.

Ultimately, replacing BatchNorm with Affine layers has proven to stabilize the network, effectively neutralizing post-training quantization degradation. As reported in Table 2, the quantized Affine + ReLU variant of Gideon not only maintains most of its original full-precision performance, but also slightly improves on a few metrics. While minor improvements are observed in some metrics after quantization, we note that our training does not employ quantization-aware optimization. Therefore, such variations may reflect statistical effects rather than a systematic regularization mechanism. Nevertheless, prior studies have reported regularization-like behavior induced by low-precision representations [3, 7], which may provide a partial explanation.

3.1.3. Adaptive Thresholding

The proposed architectural variants of Gideon and the SuperPoint baseline were benchmarked under four thresholding conditions: three fixed confidence thresholds (0.005, 0.1, and 0.3) and one dynamic adaptive threshold as described in Section 2.4.1, with consistent results.

A highly permissive threshold (0.005) maximizes Repeatability, approaching 80% for illumination changes, but reduces Viewpoint Correctness to approximately 30–38%. This is due to many weak, unstable keypoints producing low-quality matches that shift the consensus away from the correct solution.

Conversely, a restrictive threshold (0.3) acts as a strict stress test, extracting only a few, high-confidence points. Under these conditions, BatchNorm-based models collapse, whereas the Affine-based models demonstrate remarkable algorithmic robustness.

Nonetheless, the proposed Adaptive Threshold emerges as the optimal choice. Our tests show that it acts as an intelligent filter: it removes noise while preserving structural features, thereby boosting Viewpoint Correctness to 60.3% (up from 33.2% at 0.005) without sacrificing overall Repeatability.

3.2. Descriptor Dimensionality

The loss formulated in Section 2.4.2 enables the distillation of the teacher's descriptors into representations of varying sizes. In this section, we investigate the impact of descriptor dimensionality on both representational capacity and quantization robustness through a comprehensive ablation study,

Table 2. Relative performance change (Δ%) on the HPatches dataset due to int8 quantization. Positive values indicate an improvement after quantization.

Architecture	Δ Detector Repeatability		Δ Descriptor Correctness	
	Illum.	View.	Illum.	View.
BatchNorm + ReLU	-42.7%	-51.1%	-65.6%	-100.0%
BatchNorm + PWL	-27.8%	-34.3%	-32.3%	-93.1%
Affine + ReLU	-3.3%	-0.8%	**+1.9%**	**+3.4%**
Affine + PWL	-2.3%	-5.2%	-6.4%	-30.9%
SuperPoint	**-0.3%**	**+0.1%**	-1.8%	-2.0%

scaling the channel depth from 8 to 512 components.

3.2.1. Theoretical Analysis

The standard deviation of L2-normalized descriptors decreases as their dimensionality increases. This behavior is mathematically expected: since the sum of the squared components must equal one, the average magnitude of each individual element must necessarily decrease proportionally to $1/\sqrt{D}$.

An analysis of the intra-frame distribution of descriptors shown in Table 3 reveals that the representational space associated with lower dimensionalities (8, 16, and 32 channels) appears too constrained. This prevents the network from fully exploiting it, causing an information bottleneck that collapses the descriptors into lower-dimensional submanifolds.

Conversely, for dimensions of 64 and above, the ratio between the empirical standard deviation and the ideal theoretical value stabilizes at approximately 0.75. This indicates that, from 64 channels onward, the information is distributed in an almost perfectly isotropic manner (i.e. uniformly across all directions within the hyperspherical space).

Finally, high-dimensional descriptors, such as those with 128 (e.g. SIFT[17]), 256 (e.g. SuperPoint) or 512 floating-point components, exhibit a negligible standard deviation (0.047 and 0.033, respectively). This occurs because the majority of the values generated by the projection layer oscillate very close to zero. This narrow dynamic range precludes effective INT8 quantization as a significant portion of the representational expressiveness is lost due to the limited resolution of the discrete grid, causing this setup to become highly suboptimal for edge hardware deployments.

3.2.2. Empirical Validation

Empirical results on the HPatches dataset, reported in Table 4, strongly corroborate our theoretical analysis regarding intra-frame standard deviation presented in Section 3.2.1.

At lower dimensions, we observe a classical underparameterization phenomenon. For instance, with an 8-dimensional representation, performance is severely degraded, yielding a Viewpoint Correctness of approximately 0.24. Scaling to 16 dimensions provides a substantial performance leap, with Illumination Correctness increasing from 0.64 to 0.84, although viewpoint-related metrics remain suboptimal. At these lower dimensions, the representational vector is fundamentally too compressed to effectively encode the geometric and photometric complexity of a local patch.

Interestingly, expanding the capacity to 32 dimensions reveals a counterintuitive behavior where the Viewpoint Correctness in INT8 (0.613) surpasses that of its FP32 counterpart (0.583). The INT8 variant occasionally matches or slightly surpasses the full-precision model, probably due to statistical variability.

Building upon these observations, our ablation study demonstrates that constraining the network to 64 channels acts as a highly effective structural regularizer, yielding the ideal equilibrium for edge hardware. At this dimensionality, the learned representation carries sufficient information distributed isotropically across its vector space. This forces the projection layer to output highly discriminative features, achieving an optimal convergence between representational capacity and quantization robustness. In fact, floating-point performance stabilizes at excellent values (0.922 for Illumination and 0.613 for Viewpoint), while the INT8 version closely matches it with negligible degradation, proving that the dynamic range at 64 dimensions remains broad enough to robustly withstand 8-bit discretization.

In contrast, at 256 or 512 dimensions, the network exhibits capacity underutilization, populating many channels with near-zero values. This implicit sparsity does not improve matching robustness but increases memory and latency.

Even though increasing the dimensionality continues to yield marginal theoretical benefits in the floating-point domain, with Viewpoint Correctness slowly climbing to a peak of 0.654 at 512 dimensions, this expansion compresses the activation values closer to zero, resulting in a shrinking standard deviation. Consequently, this constrained dynamic range precludes effective INT8 quantization, as the narrow

Table 3. Analysis of intra-frame standard deviation across different descriptor dimensions. The theoretical standard deviation is calculated as $1/\sqrt{D}$, assuming a perfectly isotropic distribution on the L2-normalized hypersphere.

Dim.	Theor. Std.	Meas. Std.	Ratio
8	0.3536	0.1701	0.4811
16	0.2500	0.1381	0.5526
32	0.1768	0.1139	0.6443
64	0.1250	0.0899	0.7188
128	0.0884	0.0672	0.7606
256	0.0625	0.0472	0.7545
512	0.0442	0.0332	0.7510

distribution requires severe approximation to be mapped onto the 256 discrete bins of the INT8 format. Indeed, at 512 dimensions, the INT8 network achieves a lower Viewpoint Correctness (0.586) than it does with merely 32 dimensions (0.613).

Ultimately, a 64-channel descriptor emerges as the optimal architectural choice. Its broader dynamic range ensures that, during INT8 quantization, information is distributed across a significantly higher number of discrete bins, thereby fully preserving the discriminative capacity of the descriptor when deployed on the edge.

3.3. On-Device Energy Evaluation

Peak runtime validation was performed on the STM32N6 Developer Kit (ST Edge AI v3.0.0) at 1 GHz, yielding 9.003 ms latency (111.07 fps). For power characterization, we evaluated the same compiled model on a dedicated STM32N657A platform under fixed-frequency operation (800 MHz).

End-to-end inference completes in 10.87 ms with an average current of 274.6 mA, corresponding to $Q = 2.984$ mC and $E_{\text{inf}} = 5.372$ mJ per inference at 1.8 V. At 60 fps continuous operation, this maps to an inference-only power of approximately 322 mW.

From a memory perspective, the INT8 deployment requires 600.77 KB of weights and 827.25 KB peak activations, remaining well within the 4.2 MB on-chip SRAM budget and avoiding external memory transfers.

4. Conclusions

In this work we presented *Gideon*, a hardware-aware neural feature extractor designed for deployment on microcontroller-class devices. Rather than optimizing solely for theoretical efficiency, we re-framed feature extraction as a system-level problem, jointly considering memory footprint, dataflow regularity, and quantization stability as first-class design constraints.

Table 4. Ablation study on descriptor dimensionality evaluated on the HPatches dataset. Values in parentheses show the relative change ($\Delta\%$) due to quantization.

		Descriptor Correctness	
Dim (D)	**Precision**	**Illum.**	**View.**
8	float32	0.6421	0.2475
	int8	0.6316 (-1.6%)	0.2339 (-5.5%)
16	float32	0.8491	0.5085
	int8	0.8596 (+1.2%)	0.5186 (+2.0%)
32	float32	0.9123	0.5831
	int8	0.8947 (-1.9%)	**0.6136** (+5.2%)
64	float32	0.9193	0.5932
	int8	**0.9368** (+1.9%)	**0.6136** (+3.4%)
128	float32	0.9404	0.6136
	int8	0.9053 (-3.7%)	0.6068 (-1.1%)
256	float32	**0.9439**	0.6339
	int8	0.9193 (-2.6%)	0.6000 (-5.3%)
512	float32	0.9298	**0.6542**
	int8	0.9158 (-1.5%)	0.5864 (-10.4%)

By combining relational knowledge distillation with differentiable architecture search under embedded constraints, we obtained a compact convolutional topology that preserves the structural geometry of a strong teacher model while remaining fully deployable on STM32N6-class hardware.

Our experiments reveal that quantization robustness is largely architectural: replacing Batch Normalization with Affine layers and controlling descriptor dimensionality substantially improves INT8 stability. With 9 ms latency and a total footprint below 1.5 MB, Gideon demonstrates that learned local features can operate reliably in always-on embedded scenarios. While full SLAM integration is beyond the scope of this work, we report feature-level metrics such as repeatability and matching correctness, which are widely accepted indicators of downstream SLAM performance. The strong results obtained on HPatches therefore suggest promising applicability within complete SLAM pipelines.

Ultimately, the key contribution of this work lies in framing feature extraction as a system-level co-design problem, where memory constraints, dataflow regularity, and quantization stability are explicitly integrated into the training process, rather than addressed only at deployment time. This perspective enables the development of robust and efficient perception modules tailored for next-generation wearable devices, aligning with broader trends in efficient neural network design for resource-constrained environments [9, 28].

References

[1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017. 5

[2] Carlos Campos, Richard Elvira, Juan J. Gomez Rodriguez, Jose M. M. Montiel, and Juan D. Tardos. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6): 1874–1890, 2021. 1

[3] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations, 2016. 5, 6

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. *arXiv preprint arXiv:1712.07629*, 2018. 1, 3

[5] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 1

[6] Kühne et al. Levio: Lightweight embedded visual-inertial odometry for resource-constrained devices, 2026. 2

[7] Angela Fan, Pierre Stock, Benjamin Graham, Edouard Grave, Remi Gribonval, Herve Jegou, and Armand Joulin. Training with quantization noise for extreme model compression, 2021. 5, 6

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning Workshop*, 2015. 3

[9] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. cite arxiv:1704.04861. 8

[10] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018. 5

[11] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017. 3

[12] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, 2018. 4

[13] Solomon Kullback and Richard A Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 1951. 4

[14] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555, 2011. 1

[15] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 1

[16] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *ICLR*, 2019. 3

[17] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004. 7

[18] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017. 3

[19] Markus Nagel, Mart Van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1325–1334, 2019. 5

[20] Vlad Niculescu, Tommaso Polonelli, Michele Magno, and Luca Benini. Nanoslam: Enabling fully onboard slam for tiny robots. *IEEE Internet of Things Journal*, 2023. 2

[21] Guilherme Potje, Felipe Cadar, Andre Araujo, Renato Martins, and Erickson R. Nascimento. Xfeat: Accelerated features for lightweight image matching, 2024. 2

[22] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 1

[23] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 3

[24] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *ECCV*, pages 430–443. Springer, 2006. 1

[25] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011. 1

[26] David Schubert, Thomas Goll, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers. The tum vi benchmark for evaluating visual-inertial odometry. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018. 4

[27] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 1

[28] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 8

[29] MichałTyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. In *Advances in Neural Information Processing Systems*, pages 14254–14265. Curran Associates, Inc., 2020. 1

[30] Xiaoming Zhao, Xingming Wu, Weihai Chen, Peter C. Y. Chen, Qingsong Xu, and Zhengguo Li. Aliked: A lighter keypoint and descriptor extraction network via deformable transformation, 2023. 2