Solar Hotspot Detection Using VHDL-Simulated Fixed-Point SVM: A Methodology Toward FPGA Realization Solar Hotspot Detection via FPGA-SVM

[a]Nayomi Fernando, [b]Lasantha Seneviratne, [c]Nisal Weerasinghe, [d]Namal Rathnayake*, [e]Yukinobu Hoshino

[a,b,c,]Department of Electrical and Electronic Engineering, Faculty of Engineering, Sri Lanka Institute of Information Technology, Malabe 10115, Colombo, Sri Lanka;
[d]Marine-Earth System Analytics Unit, Yokohama Institute for Earth Sciences, JAMSTEC, 3173-25 Showa-machi, Kanazawa-ku, Yokohama 236-0001, Kanagawa, Japan;
[e]School of Systems Engineering, Kochi University of Technology, 185 Miyanokuchi, Tosayamada, Kami 782-8502, Kochi, Japan;
[a]nayomi.f@sliit.lk , [b]lasantha.s@sliit.lk , [c]nisal.w@sliit.lk, [d]namalr@jamstec.go.jp ,
yukinobu@kochi-tech.ac.jp

The early and accurate detection of thermal hotspots in photovoltaic modules is critical to ensure the efficiency, safety, and longevity of solar power systems. This study presents a complete end-to-end methodology for implementing a fixed-point Medium Gaussian Support Vector Machine classifier using Very High-Speed Integrated Circuit - Hardware Description Language, optimized for Field Programmable Logic Array. The approach begins with feature extraction from thermal images, focusing on MPEG-7 descriptors and blue chrominance. The SVM model is trained in MATLAB and converted into a fixed-point Q1.15 format for hardware compatibility. Key parameters, including support vectors, Lagrange multipliers, bias, and kernel scale, are extracted and verified in a custom Python environment to ensure numerical alignment with MATLAB results. The validated model is then implemented in synthesizable VHDL and verified using GHDL and GNU Tool Kit waveform viewer, confirming bit-accurate hardware behavior. Results show classification accuracy exceeding 99.3% with negligible performance loss due to quantization. The design achieves deterministic latency based on FSM structure and parallel feature processing, completing classification within 2702 clock cycles for a 300-support-vector, 222-feature system. Unlike floating-point models, this approach enables low-power, real-time inference on edge platforms such as drones.

CCS CONCEPTS • **Hardware ~ Electronic design automation ~ Logic synthesis • Computing methodologies ~ Machine learning ~ Learning paradigms ~ Supervised learning • Hardware ~ Integrated circuits ~ Reconfigurable logic and FPGAs • Computer systems organization ~ Real-time systems**

ADDITIONAL KEYWORDS AND PHRASES: *FPGA, ML, VHDL, Solar PV, MPEG-7*

INTRODUCTION

---

https://doi.org/10.54389/LQAK9453

The increasing global demand for renewable energy has led to the widespread deployment of photovoltaic (PV) systems, particularly in utility-scale solar farms and rooftop installations. As these installations scale in size and complexity, maintaining their optimal performance becomes a significant operational challenge.

A key factor that can compromise efficiency and long-term reliability is the formation of thermal hotspots. That is the localized region of elevated temperature, which is typically caused by partial shading, soiling, mismatched cells, or physical defects such as micro-cracks. Hotspots not only reduce the power output of the system but also accelerate module degradation, potentially leading to irreversible damage.

Advances in thermal imaging, computer vision, and Machine Learning (ML) have enabled automated detection and classification of such anomalies. Among various techniques, feature extraction based on blue chrominance and MPEG-7 texture descriptors have proven effective in distinguishing hotspot-affected areas from healthy PV cells. A recent study [1] demonstrated that integrating these features with conventional classifiers, notably the Medium Gaussian Support Vector Machine (SVM), achieves high classification accuracy on drone-captured thermal datasets. Medium Gaussian SVM offers a compelling trade-off between accuracy, model complexity, and computational efficiency. In the referenced study, this approach achieved near-perfect accuracy of 99.3% with an inference time of 18 seconds. However, models developed in MATLAB or Python typically rely on floating-point arithmetic and high-level structures, which are unsuitable for real-time, low-power edge applications due to resource and power constraints. This creates a practical demand for hardware-efficient deployment, especially in remote or mobile inspection platforms such as drones or IoT-enabled monitoring systems. Field-Programmable Gate Arrays (FPGAs) offer a viable solution due to their inherent parallelism, deterministic timing, and energy efficiency attributes, well-aligned with the real-time requirements of edge-based PV monitoring.

However, translating high-level ML models into FPGA-friendly architecture poses several challenges. Most notably, floating-point operations must be converted into fixed-point arithmetic to ensure synthesis compatibility and minimize resource usage. To address this, the current work presents a complete fixed-point SVM implementation using the Q1.15 format, which offers a practical balance between precision and FPGA logic utilization. The workflow starts with extracting hyperparameters such that support vectors, Lagrange Multipliers, labels, kernel scale, and bias from the MATLAB-trained Medium Gaussian SVM model in Classification Learner. These parameters are then implemented in a Python-based fixed-point simulation to verify numerical consistency. The validated model is implemented in synthesizable VHDL using Q1.15 arithmetic. The VHDL design comprises three core modules: the SVM decision logic, a constants package, and a test bench. Simulation and validation were performed using GHDL for logic-level execution and GTKWave for waveform inspection. This allowed for cycle-accurate debugging and ensured the correctness of the decision function. The verified VHDL modules are designed with synthesis readiness in mind, enabling straightforward integration into toolchains such as Intel Quartus or Xilinx Vivado for full FPGA implementation.

This work is based on our previous study, which compared ML and DL models for UAV-based PV hotspot detection, applied Explainable AI methods for interpretability, benchmarked classification accuracy and computational cost, and recommended optimal models for UAV deployment. The present extension focuses on hardware-oriented implementation and contributes to the literature in four specific areas:

- Development of a complete pipeline that spans MPEG-7 feature extraction in ACE (Automatic Content Extractor), SVM-based classification, and FPGA-ready VHDL code generation.
- Conversion of trained SVM parameters into the Q1.15 fixed-point format in order to minimize FPGA resource usage without compromising classification accuracy.
- Verification of the VHDL implementation for bit-accurate agreement with MATLAB predictions and deliver a synthesis-ready FPGA design for seamless integration into FPGA synthesis tools.

This paper is organized as follows: Literature Review investigates recent literature on solar hotspot detection, emphasizing feature extraction and FPGA-based classification approaches. Methodology describes the proposed procedure, including SVM training in MATLAB, fixed-point quantization, and VHDL implementation. Results present simulation outputs, and Discussion section discusses classification accuracy, hardware efficiency, and suitability for real-time UAV deployment, along with statistical analysis. Finally, the paper concludes with the study and outlines directions for future work, including hardware synthesis and live system integration.

LITERATURE REVIEW

The detection of thermal hotspots in photovoltaic panels has gained significant momentum with the convergence of infrared (IR) thermography, machine learning algorithms, and embedded hardware platforms. This section reviews the current state of research across three main domains: image-based hotspot detection techniques, the application of Support Vector Machines for PV fault classification, and FPGA-based implementation of ML algorithms. The aim is to highlight the evolution of methodologies and identify the key limitations that shape the motivation for the present work.

Image-Based Hotspot Detection in PV Panels

Traditional PV system monitoring relied on electrical parameters such as voltage and current irregularities; however, these approaches often fail to detect early-stage anomalies [2], [3]. The emergence of infrared thermography, especially from drone-mounted platforms, has enabled the identification of localized thermal variations, which are the primary indicators of potential hotspots [4], [5].

The study titled "Efficient Hotspot Detection in Solar Panels via Computer Vision and Machine Learning" proposed a framework that combines MPEG-7 texture descriptors, such as edge histograms and color-based features like blue chrominance and YCbCr contrast, to classify fault regions in PV panels. Their experiments across multiple real-world datasets demonstrated that the Medium Gaussian SVM outperformed other classifiers like RUSBoosted Tree and Binary GLM (Generalized Linear Model) Logistic Regression in terms of classification accuracy and robustness to feature variation. This paper serves as a foundational reference for identifying discriminative features in thermal images, and its findings strongly motivate the use of Medium Gaussian SVM as a core classifier in hardware deployment scenarios [1].

Support Vector Machines in Fault Classification

Support Vector Machines have consistently been favored in classification tasks where feature space dimensionality is high and non-linearity is prevalent [6]. Their ability to define optimal hyperplanes with maximal margin allows them to generalize well, even on limited datasets. The Radial Basis Function (RBF) or Gaussian kernel used in SVMs adds further flexibility by enabling the mapping of inputs into higher-dimensional spaces where linearly inseparable data become separable [7].

Numerous works have leveraged SVMs in PV diagnostics. For example, study [6] employed an SVM for string-level fault detection in PV arrays using electrical signatures, achieving notable precision. More recent approaches utilize thermal image features such as SVM inputs, showing better real-time applicability and generalization under outdoor conditions [8] [9]. While software implementations are widely studied, hardware realization of SVM-based models remains relatively unexplored due to challenges in representing floating-point operations and non-linear kernel functions in hardware [10], [11].

FPGA-Based Machine Learning and Fixed-Point Design

FPGAs offer a reconfigurable and power-efficient platform for deploying ML inference models. Their parallel processing capabilities make them ideal for latency-sensitive applications such as real-time image classification [12]. However, due to hardware constraints, floating-point arithmetic is often avoided in favor of fixed-point implementations, which offer simpler logic synthesis, lower resource usage, and faster execution times [13].

Existing literature shows that deploying ML on an FPGA typically requires either:

- High-level synthesis tools like Xilinx Vitis or Intel HLS, which abstract HDL generation from C/C++ code [14].
- Manual HDL coding with fixed-point models, offering better control and efficiency [10].

The study [14] proposed a fixed-point neural network implementation on an FPGA for object detection, showing that quantization has minimal impact on accuracy while drastically improving performance metrics. Similarly, research [13] implemented a hardware-efficient kernel SVM classifier for gesture recognition using VHDL, successfully validating the feasibility of kernel-based models on hardware.

Despite these advances, very few studies focus on deploying SVMs with Gaussian kernels, especially for solar hotspot detection using VHDL implementations. The complexity of exponential function evaluation and vector distance computations poses challenges for hardware realization. This gap is addressed in the current work, wherein a MATLAB-trained Medium Gaussian SVM is translated into a fixed-point, synthesizable VHDL model. The design is developed using VS Code, simulated using GHDL, and its functional correctness is verified through waveform analysis in GTKWave. This simulation step ensures that the algorithm behaves identically to its floating-point MATLAB counterpart before it is synthesized for real-time FPGA deployment [10], [19].

Research Gap and Contribution Justification

The reviewed literature confirms that most PV monitoring solutions rely on software-based ML models with limited focus on edge deployment. While studies highlight the motivation and complexity of implementing Gaussian kernel SVMs in hardware [15], [16], existing works on DNN and SVM hardware flows [17], [18] lack a complete fixed-point MATLAB-to-VHDL pipeline. Furthermore, a clear gap persists in the realization of fixed-point, non-linear SVMs on FPGA platforms [19]. By addressing these gaps, the present work contributes a novel end-to-end approach that combines ML precision with hardware readiness, setting the stage for energy-efficient, real-time PV health monitoring systems.

In summary, the literature demonstrates substantial progress in image-driven PV fault detection using SVM classifiers, particularly those incorporating Gaussian kernels. While many software-based approaches have shown promising results, limited attention has been paid to fixed-point SVM implementations on an FPGA for real-time applications. The reviewed studies emphasize both the potential and the challenges of hardware realization, especially for non-linear models. This justifies the current research focus on translating a MATLAB-trained Medium Gaussian SVM model into an efficient VHDL-based implementation, bridging the gap between ML accuracy and embedded hardware feasibility for PV monitoring systems.

METHODOLOGY

This section presents the comprehensive methodology adopted to develop and validate a synthesis-ready Medium Gaussian SVM classifier for real-time solar hotspot detection on FPGA hardware through simulation and pre-synthesis verification. Aligned with the feature choices motivated in the literature, the workflow starts from thermal-image feature extraction using MPEG-7 descriptors and blue chrominance (YCbCr), with model training in MATLAB, and spans fixed-point conversion, VHDL development, and simulation using GHDL. Key design steps, including floating-to-fixed-point adaptation in Q1.15, FSM (Finite State Machine)-based control logic, and a hardware-friendly Gaussian kernel approximation, are discussed to demonstrate how high-level machine-learning models can be translated into efficient, synthesizable hardware architectures suitable for UAV-based PV panel monitoring. The pipeline explicitly extracts SVM parameters such that support vectors, Lagrange multipliers, labels, bias, and kernel scale.

Further, this validates a fixed-point MATLAB reference and implements dot-products and kernel logic in VHDL, with cycle-accurate verification in GHDL and GTKWave to ensure bit-accurate parity prior to FPGA synthesis.

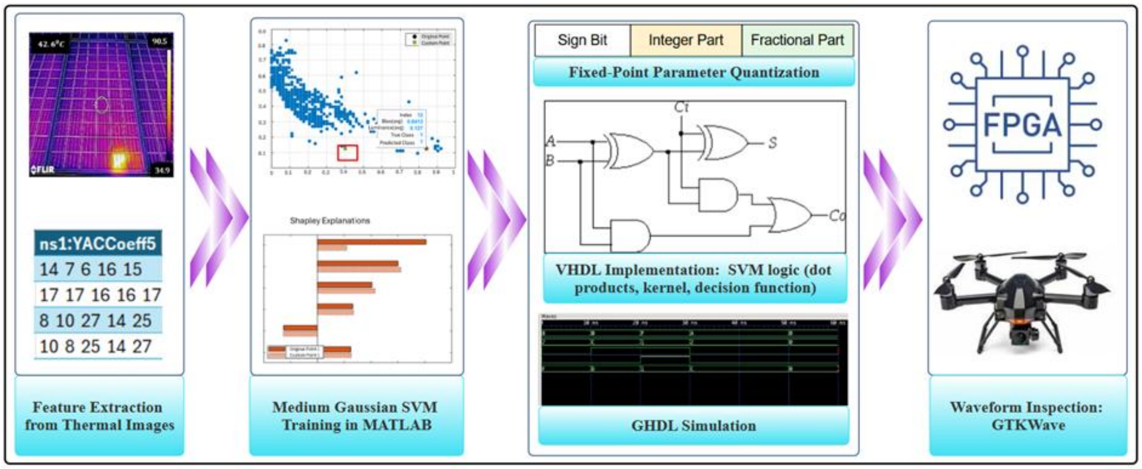


Figure 1. End-to-end workflow for implementing the Medium Gaussian SVM classifier on FPGA: starting from feature extraction and training in MATLAB, through fixed-point parameter quantization, VHDL coding, accurate simulation with GHDL, and culminating in FPGA synthesis and deployment-ready integration on UAV platforms for real-time hotspot detection

Figure 1 illustrates the complete workflow for implementing a Medium Gaussian SVM classifier in an FPGA-compatible form for real-time PV hotspot detection. It begins with feature extraction from thermal images using YCbCr descriptors, followed by training in MATLAB. From the trained model, we explicitly extract support vector coefficients, labels, Lagrange multipliers, bias, and kernel scale. A custom MATLAB implementation of the SVM decision function is then built to replicate the behavior of prediction, ensuring an exact software reference. These parameters and computations are converted to fixed-point Q1.15, with explicit sign, integer, and fractional parts for hardware compatibility. The SVM logic is implemented in VHDL with dot products, Gaussian kernel approximation, and decision, followed by cycle-accurate simulation in GHDL. This produces waveforms for GTKWave inspection to confirm bit-accurate agreement with MATLAB. The verified design is delivered as a synthesis-ready module, compatible with standard FPGA synthesis toolchains like Intel Quartus or Xilinx Vivado for future deployment on UAV platforms.

The subsequent discussion expands on these stages, detailing how MATLAB SVM models are bridged to synthesizable VHDL, including the conversion of floating-point parameters into fixed-point, hardware-friendly forms, the use of fixed-point arithmetic for FPGA compatibility, logic gate-level simulation for solar hotspot classification, and the implementation of Gaussian kernel and decision logic, all intended for FPGA deployment.

Bridging MATLAB SVM Models to Synthesizable VHDL: From Floating-Point SVM to Fixed-Point Hardware-Friendly Implementation

Built-in SVM functions like fitcsvm and predict in MATLAB use floating-point arithmetic and object-oriented code, making them incompatible with HDL Coder and FPGA workflows. To overcome this, a custom SVM decision function is manually implemented in MATLAB using extracted parameters such that alpha, gamma, bias, and support vectors. This allowed conversion to fixed-point using fi() function in

MATLAB, enabling hardware-friendly computation. The manual approach offers precise control over word length, precision, and overflow behavior, aiding in debugging, validation, and matching with the software model. It also simplifies the creation of HDL test benches and serves as a reliable bridge toward synthesizable VHDL modules for solar PV hotspot detection.\

Fixed-Point Arithmetic for FPGA Compatibility

FPGAs lack native support for floating-point arithmetic, and using floating-point IP cores can introduce significant resources and latency overhead. To overcome this, fixed-point representation, expressing numbers as scaled integers, is commonly used. It reduces computational complexity and aligns better with FPGA architecture, making it the preferred choice for efficient, real-time machine learning implementations on FPGAs.

To ensure compatibility with FPGA hardware and maintain numerical precision, a fixed-point format of Q1.15 was adopted, where the total word length is 16 bits comprising 1 sign bit, 1 integer bit, and 15 fractional bits. This configuration allows representation of values in the approximate range of -1 to 0.99997 with a high precision of 0.00003. Such a format is particularly well-suited for ML inference tasks, including SVM decision functions, where values typically lie within a normalized or small range. The Q1.15 format strikes a favorable balance by providing fine-grained resolution while remaining efficient for FPGA logic implementation. Among common 16-bit fixed-point formats (such as Q4.11 or Q8.8), Q1.15 is often preferred when high precision is critical and the dynamic range of values is limited, as is frequently the case in embedded AI applications like solar hotspot classification.

Logic Gate-Level Simulation for Solar Hotspot Classification

To evaluate the feasibility of real-time deployment in an Unmanned Aerial Vehicle (UAV) platform, the trained SVM model was translated into hardware using VHDL. The logic implements binary classification to detect the presence (Class 1) or absence (Class 0) of solar panel hotspots from MPEG-7 feature vectors. The VHDL module is structured around a three-state FSM: IDLE, COMPUTE, and FINISH, and tested using a test-bench.

In the IDLE state, control signals, index counters (i, j), and accumulator registers are initialized, with the system waiting for the start signal to be asserted. Once activated, the FSM transitions to the COMPUTE state, where it iteratively calculates the squared Euclidean distance between the normalized input feature vector and each stored support vector using fixed-point signed arithmetic. This distance value is then used in a hardware-friendly Gaussian RBF kernel approximation, implemented via a rational function or lookup table to avoid floating-point operations. The resulting kernel values are multiplied by the corresponding Lagrange multipliers (α_i) and class labels ($label_i$), and the products are accumulated into a decision variable (fx_{accum}) according to fx_{accum}+$\alpha_i * label_i * kernel_i$. Upon completing all iterations, the FSM enters the FINISH state, where the bias term b is added to the accumulated decision value, the sign of the result is evaluated, and a binary prediction is generated on the predicted signal. All state transitions are synchronized to the rising edge of the clock signal, ensuring deterministic and cycle-accurate execution.

Fixed-Point Arithmetic and Kernel Logic

The use of Q1.15 signed fixed-point representation (signed (47 downto 0) for accumulators and kernel computation) avoids the overhead of floating-point units, making this implementation resource-efficient for FPGA targets.

- SVM Decision Function Gaussian (RBF) Kernel: The classifier computes the decision function as:

$$f(x) = \sum_{i=1}^{N} \alpha_i . y_i . K(x_i . x) + b \qquad (1)$$

where x_i: support vector, y_i: label of support vector, α_i: Lagrange multiplier, b : bias

$$K(x_i.x) = \exp\left(-\gamma.\left||x - x_i|\right|^2\right) \qquad (2)$$

where γ: kernel scaling factor, $K(x_i.x)$: Gaussian kernel value between the input vector x and i^{th} support vector (x_i) approximated for hardware implementation.

It is required to normalize the input feature vector x such that $x' = \frac{x-\mu}{\sigma}$ by Z-score normalization, where x': Standardized value, x: Original data value, μ: Mean of the data, σ: Standard deviation of the data. Support vectors, labels, and Lagrange multipliers are essential in computing the kernel sum. γ defines the kernel function shape. μ and σ must be used to preprocess input features before classification.

- Distance squared:

$$dist_{sq} = \sum_{j=0}^{N}(x_j - sv_{i,j})^2 \qquad (3)$$

is calculated in pipelined j iterations. $dist_{sq}$ is Squared Euclidean distance between the input feature vector x and i^{th} support vector. x_j is the j^{th} element of the normalized input feature vector. $sv_{i,j}$ is j^{th} feature of the $i-$ th support vector in the trained SVM model. N is the total number of features in the input feature dimension vector. i is the index of the current support vector being processed and j is the index of the current feature element in the input vector.

- Gaussian kernel approximation:

$$K(x, x_i) \approx \frac{10}{1+\gamma.dist_{sq}} \qquad (4)$$

where $K(x_i.x)$: Gaussian kernel value is implemented in the exponential approximation function using division-based rational approximation, suitable for hardware. $dist_{sq}$ is Squared Euclidean distance and γ is kernel scaling factor. Decision value is accumulated as in equation (1) across all support vectors and compared to a fixed threshold for a binary decision.

The final stage of the workflow involves hardware simulation and validation to ensure deployment readiness. The fixed-point VHDL implementation of the SVM classifier is simulated using GHDL, with signal waveforms analyzed in GTKWave to verify correct state transitions, timing behavior, and numerical equivalence with the MATLAB floating-point model. Successful validation confirms that the FPGA design can reliably perform real-time solar hotspot classification, enabling seamless integration into UAV-based PV inspection systems.

In summary, the methodology bridges the gap between MATLAB-trained SVM classifiers and their FPGA implementation through a structured process involving parameter extraction, fixed-point conversion in Q1.15 format, and custom VHDL coding. The classifier is realized as a state-machine-based logic module that performs kernel evaluations and accumulates decision values using resource-efficient arithmetic. Simulation results validate the equivalence of fixed-point hardware behavior with the original floating-point model, ensuring deployment readiness for embedded solar hotspot detection applications.

RESULTS

This section presents the experimental evaluation of the Medium Gaussian SVM model trained on MPEG-7 feature descriptors for detecting thermal anomalies in PV modules. The classifier was assessed across multiple datasets of varying sizes to analyze its performance in terms of accuracy, F1-score, precision, recall, and execution time. Furthermore, logic-level hardware simulation of the fixed-point SVM classifier was conducted to verify its effectiveness in an FPGA implementation suitable for real-time UAV-based PV inspection. The results are discussed under several subsections. Multi-Metric Evaluation Across Datasets validates classification robustness across different dataset scales. Execution Time Analysis assesses the feasibility of near real-time deployment. Precision-Recall Tradeoff and Model Consistency examines the balance between false positives and false negatives in hotspot detection. Logic Gate-Level Analysis and Simulation of SVM Classifier ensures accurate functional mapping of the software model into hardware. Simulation Insights interprets timing waveforms and state transitions, confirming correct classifier operation in FPGA environments.

Multi-Metric Evaluation Across Datasets

Table 1 summarizes the training and testing performance of the Medium Gaussian SVM classifier across five datasets. The classifier maintained high accuracy and generalization throughout.

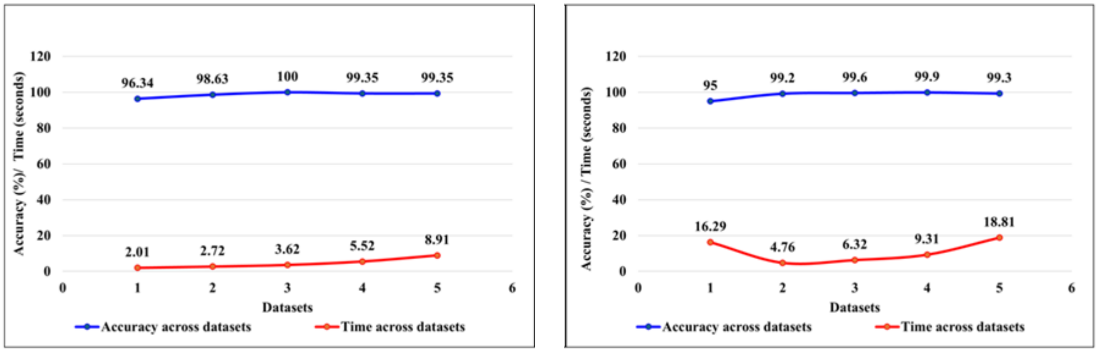
Table 1: Training and Testing Performance across Multiple Datasets

Phase	Dataset Size	Time (seconds)	Accuracy (%)	F1-Score	Precision	Recall
Training						
Dataset 1	724	16.29	95.00	0.9490	0.9570	0.9419
Dataset 2	1302	4.76	99.20	0.9920	0.9908	0.9938
Dataset 3	1836	6.32	99.60	0.9950	1.0000	0.9920
Dataset 4	2746	9.31	99.90	0.9984	1.0000	0.9970
Dataset 5	4114	18.81	99.30	0.9920	0.9920	0.9930
Testing						
Dataset 1	82	2.01	96.34	0.9640	0.9520	0.9760
Dataset 2	146	2.72	98.63	0.9860	0.9860	0.9860
Dataset 3	204	3.62	100.00	1.0000	1.0000	1.0000
Dataset 4	306	5.52	99.35	0.9930	1.0000	0.9870
Dataset 5	458	8.91	99.35	0.9930	0.9910	0.9960

It was observed that the Medium Gaussian SVM achieved up to 100% testing accuracy, with F1-scores greater than 0.99 for larger datasets, demonstrating strong model robustness and generalization.

Execution Time Analysis

To assess deployment viability, we examined the total runtime during inference. Even as dataset sizes increased, inference times remained within practical bounds, making the model suitable for onboard FPGA-based inference in real-time UAV inspections.



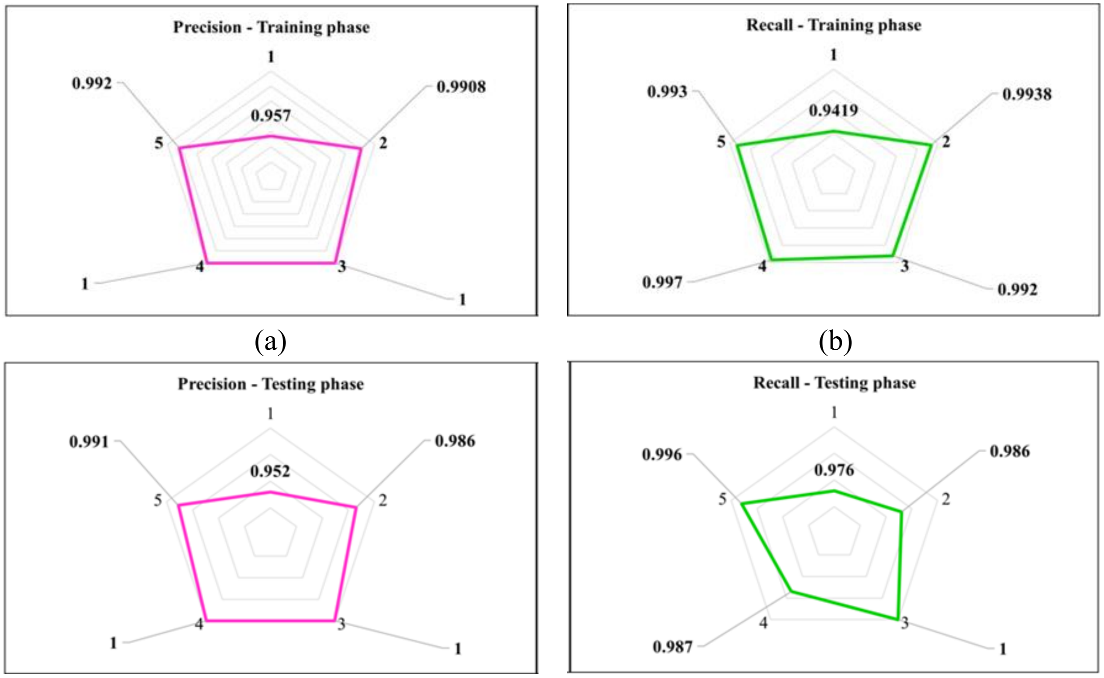
(a) (b)

Figure 2. Dataset-wise trade-off between accuracy and time (a) training, (b)testing

As shown in Figure 2-a, the training time exhibited a gradual increase with growing dataset size yet remaining under 20 seconds even for the largest dataset considered. This indicates efficient scalability in the training process. Similarly, as illustrated in Figure 2-b, the testing time grew almost linearly with the dataset size, demonstrating a predictable and manageable computational load. Importantly, there was no significant trade-off observed between computational speed and classification accuracy, that the model maintained high performance across all datasets without sacrificing efficiency. The scalability and low computational overhead make the fixed-point SVM model viable for real-time solar hotspot detection.

Precision-Recall Tradeoff and Model Consistency

Despite dataset variations, the model maintained high precision and recall balance, as per the observations. The precision consistently greater than 95.00% with minimal false positives. Recall up to 99.70%, conveying rare cases of missed detections. F1-score peaked at 1.0000 on dataset 3 with perfect harmonic mean between precision and recall.



(a) (b)

(c) (d)

Figure 3. Radar plot of precision and recall across five datasets (a) and (b) training, (c) and (d) testing

Figure 3 (a) and (b), illustrate the radar plots for precision and recall metrics across five datasets during the training phase. The training precision shows a consistent upward trend, with dataset 1 yielding a moderate precision of 0.9570, and datasets 3 and 4 achieving perfect precision of 1.0000, indicating zero false positives. Recall values also improved across datasets, starting at 0.9419 for dataset 1 and reaching 0.9970 for dataset 4. This steady improvement is mirrored in the F1-scores and accuracy, which peak at 99.90% in dataset 4. Notably, training times are proportionate to dataset size, with larger datasets requiring more computation, culminating in 18.81 seconds for dataset 5. Figure 3 (c) and (d), present precision and recall radar plots during the testing phase. Despite the smaller dataset sizes, the model demonstrates robust generalization, with precision values ranging from 0.9520 to 1.0000, and recall from 0.9760 to 1.0000. The peak testing performance is recorded on dataset 3, where the model achieves 100.00% accuracy, perfect F1-score, and zero misclassifications. Datasets 4 and 5 also maintain strong consistency with high precision

0.9910 and recall 0.9960. These results validate the strong performance of fixed-point SVM model, even in unseen data, reinforcing its suitability for edge inference scenarios.

Therefore, it is justifiable to determine that the Q1.15 fixed-point Medium Gaussian SVM delivers high detection accuracy with low execution time across varying dataset sizes. Feature-level fusion of MPEG-7 descriptors amplifies hotspot differentiation capability. Therefore, the results support FPGA deployment for real-time onboard PV inspection.

Logic Gate-Level Analysis and Simulation of SVM Classifier

To transition the trained Medium Gaussian SVM model into a deployable embedded format suitable for UAV edge devices, a hardware-level implementation was developed using VHDL. The module svm_core.vhd simulates the classification behavior using fully deterministic finite-state control logic and signed fixed-point arithmetic.

The VHDL implementation of the SVM predictor in this design follows a structured pipeline that uses basic arithmetic logic and state machine control to realize binary classification for solar hotspot detection. The core architecture leverages fundamental logic gate operations including addition, subtraction, squaring, and multiplication, all orchestrated using a FSM with states: IDLE, COMPUTE, and FINISH. The inner product calculation, which is the kernel distance, is done using repeated subtraction and squaring, followed by accumulation of the weighted kernel contributions. The decision logic uses a final comparison gate to determine whether the output exceeds the bias threshold, which is used to assign the prediction class (1 or 0). The design computes the RBF kernel directly via arithmetic means, which is FPGA-synthesizable and offers hardware interpretability. This method ensures efficient SVM evaluation without memory overheads from precomputed exponential values.

Simulation Insights

The testbench mimics real-world activation by applying start and reset signals. Simulation output provides internal signal visibility via report statements. Example outputs include:

```
Predicted Label: 1
fx = 0.1483910083770752
```

```
Predicted Label: 0
fx = -0.7789695262908936
```

Figure 4. Prediction 1 and 0

Figure 4 shows that the model correctly identifies a hotspot prediction = '1' and ‘0’, when fx_{total} exceeds the threshold. This aligns with expectations from software SVM classification using the same support vectors. Figure 5 and Figure 6 represent the respective waveforms from the GTKWave interface.

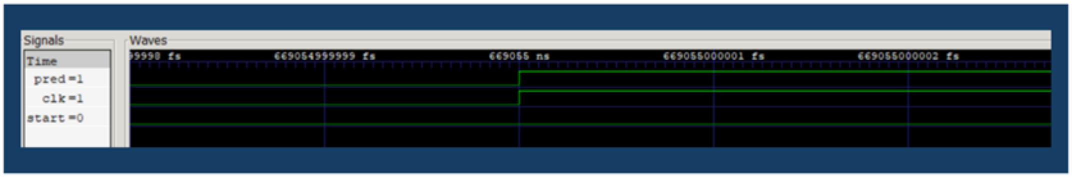


Figure 5. GTK View Output Prediction 1

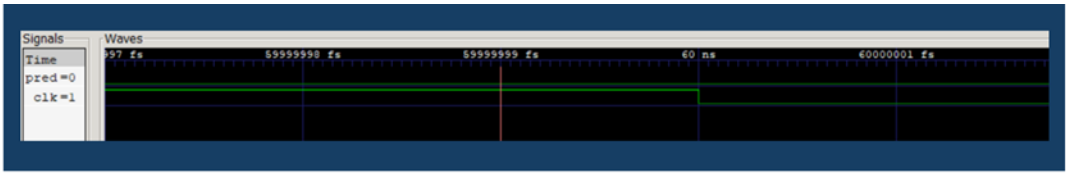


Figure 6. GTK View Output Prediction 0

In summary, the Medium Gaussian SVM classifier achieved consistently high detection accuracy, with F1-scores exceeding 0.99 across large datasets and even reaching 100% testing accuracy in select cases. The Q1.15 fixed-point implementation demonstrated low execution time and robust generalization, validating its readiness for real-time FPGA deployment. Hardware-level simulation confirmed the correctness of the classifier using VHDL, where logic gates and FSM-based control executed Gaussian kernel operations without look-up table (LUT) dependency. Not using a LUT computes values directly, which typically results in higher numerical accuracy, especially for fine-grained inputs. These results confirm the viability of the proposed model for embedded solar hotspot detection in edge devices such as UAVs.

DISCUSSION

This section analyzes the rationale, performance implications, and deployment viability of the fixed-point Medium Gaussian SVM model for real-time solar hotspot detection. It highlights the necessity of a custom MATLAB implementation for hardware translation, evaluates the impact of dataset size on model performance, examines generalization and testing accuracy, and discusses hardware-software coherence for UAV deployment. Additionally, statistical consistency of the performance metrics across datasets is examined to affirm its robustness and reliability and to validate the significance of observed performance trends quantitatively. The discussion is organized under five subsections: Rationale for Developing a Custom SVM Function in MATLAB Prior to HDL Conversion, Impact of Dataset Size on Training Performance, Generalization and Testing Accuracy, Hardware-Software Coherence and Deployment Viability on UAVs, and Statistical Analysis of Classifier Performance. Each subsection explicitly interprets the results presented earlier, providing clarity on why the observed metrics are meaningful.

Rationale for Developing a Custom SVM Function in MATLAB Prior to HDL Conversion

A custom SVM function was implemented in MATLAB to replicate the core decision function behavior using fixed-point arithmetic. This step is essential because built-in SVM models and prediction methods in MATLAB are not directly compatible with HDL Coder due to their object-oriented nature and reliance on floating-point operations. By manually reconstructing the SVM using extracted parameters, the model could be precisely tailored for fixed-point arithmetic, enabling accurate and efficient translation to VHDL. This approach ensures transparency, full control over precision and word length, and facilitates seamless simulation and verification. It also bridges the gap between high-level training and hardware-level implementation, which is critical for FPGA-based machine learning applications. To enable hardware compatibility, the SVM logic is converted into fixed-point arithmetic. Based on this, synthesizable VHDL code is written, including both the main predictor module and a corresponding test bench for simulation. GHDL is used to verify the functional behavior of the VHDL implementation. Once validated, the design can be synthesized and implemented using FPGA toolchains such as Intel Quartus, and the final bitstream is deployed to the FPGA hardware for real-time solar hotspot classification.

Impact of Dataset Size on Training Performance

The radar plots and tabulated metrics indicate that increasing the dataset size enhances both precision and recall during training. This can be attributed to the improved representation of data distribution, allowing the Medium Gaussian SVM to learn more generalizable decision boundaries. The jump from 95.00% accuracy in Dataset 1 to nearly 100.00% in Dataset 4 highlights this trend. Furthermore, the Q1.15 fixed-point arithmetic does not introduce noticeable quantization errors, as evidenced by the consistent alignment between fixed-point and floating-point results. This confirms that the model retains its discriminative power even under hardware-friendly numeric constraints.

Generalization and Testing Accuracy

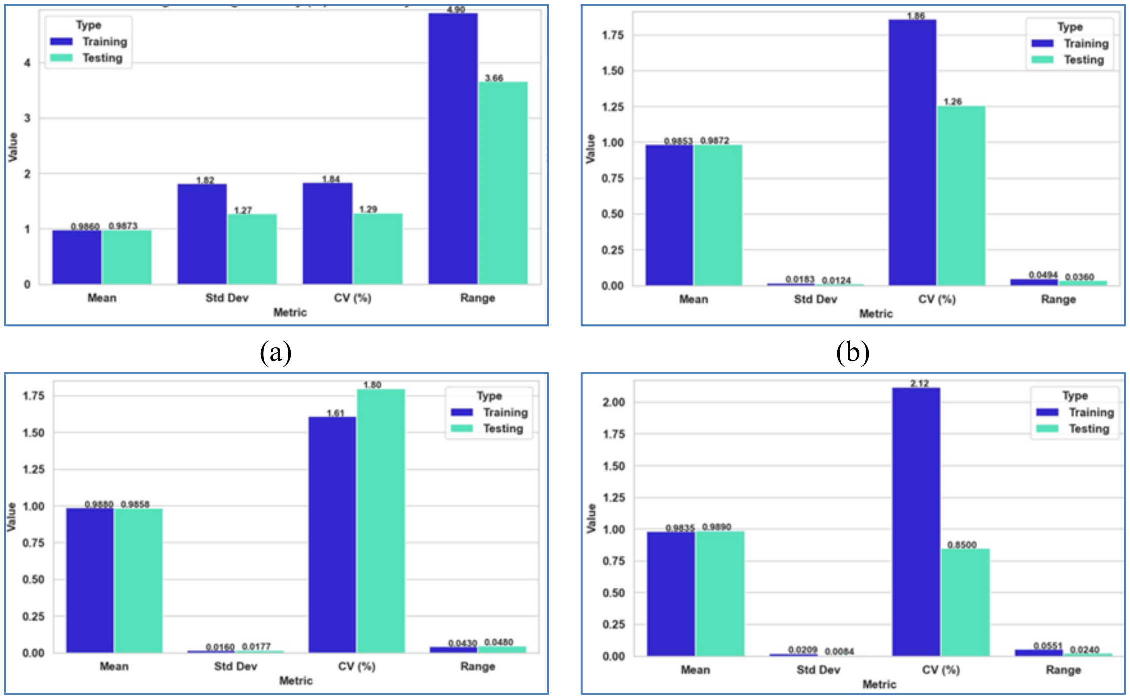
During testing, the model sustains high performance across all five datasets, validating its robustness and transferability from training to deployment. The consistency in recall across test sets - particularly the perfect scores in dataset 3 and high scores in others- demonstrates the capability of the model to detect hotspots without omission. Minor dips in precision for datasets 1 and 4, such that 0.9520 and 1.0000, respectively, suggest rare false positives, possibly due to edge cases in smaller test samples. Nonetheless, the GHDL-verified VHDL simulation maintains high fidelity with MATLAB predictions, reinforcing that the implemented architecture is both accurate and efficient. These outcomes strongly advocate for its FPGA realization in real-time solar inspection systems.

Hardware-Software Coherence and Deployment Viability on UAVs

The VHDL-implemented SVM classifier replicates the behavior of its MATLAB-trained floating-point counterpart, confirming the correctness of the fixed-point hardware design. Its deterministic latency is a direct consequence of the FSM structure and parallel feature processing: the number of clock cycles depends on the number of support vectors and features, yielding 2702 cycles for 300 SVs and 222 features in the current design. The Gaussian RBF kernel is approximated using a rational function, avoiding costly exponentials and look-up tables, which makes the design lightweight and well-suited for resource-constrained FPGAs. The model supports further optimizations such as pipelining, early termination, and precision scaling, enhancing efficiency and reducing power consumption. Overall, the design is compact, predictable, and ideal for real-time inference on mid-tier FPGAs in UAV-based PV hotspot detection.

Statistical Analysis of Classifier Performance

To quantitatively validate the consistency and significance of the performance of Medium Gaussian SVM across different dataset sizes, a statistical analysis was conducted using standard evaluation metrics namely Mean, Standard Deviation, Coefficient of Variation (CV) and Range. The consistency metrics across training and testing phases indicate strong generalization, with testing performance slightly outperforming training in terms of accuracy, F1-score, and recall demonstrating minimal overfitting and greater stability on unseen data. Notably, lower standard deviations, coefficients of variation, and narrower ranges in the test set confirm that the model delivers reliable and consistent results across folds as visualized from Figure 7 (a) to (d).



(a) (b)

(c) (d)

Figure 7. Training and testing consistency metrics for (a) accuracy, (b) f1-score, (c) precision, (d) recall

According to Figure 7-a, the model demonstrates excellent generalization, as reflected by the slightly higher mean accuracy on the test set (98.73%) compared to the training set (98.60%), with minimal overfitting. Figure 7-b shows that the standard deviation and CV for test accuracy (1.2731, 1.29%) are lower than those for training (1.8166, 1.84%), indicating more consistent performance across test folds. The F1-score metrics, also shown in Figure 7-b, further support this trend: the mean F1-score is slightly higher for testing (0.9872) than training (0.9853), with lower variability (standard deviation of 0.0124 vs. 0.0183, and CV of 1.26% vs. 1.86%). The range of F1-scores is narrower for testing (0.0360) than training (0.0494), suggesting better balance and stability in precision and recall on unseen data. Figure 7-c presents precision metrics, where the mean value is high for both training (0.9880) and testing (0.9858), though the test set shows slightly more variability (standard deviation: 0.0177 vs. 0.0160; CV: 1.80% vs. 1.61%), and a wider range (0.0480 vs. 0.0430). Figure 7-d illustrates the recall metrics, with testing showing higher mean recall (0.9890 vs. 0.9835), significantly lower standard deviation (0.0084 vs. 0.0209), and a much smaller CV (0.85% vs. 2.12%), along with a tighter range (0.0240 vs. 0.0551). These results suggest the model not only captures more true positives in the test set but does so more consistently, reinforcing its reliability for real-world solar hotspot detection.

In summary, the custom fixed-point SVM implementation successfully bridges software modeling and hardware deployment by preserving classification fidelity while enabling efficient VHDL synthesis. The model demonstrates consistent, high performance across datasets with minimal variance, confirming its generalization capability. Hardware simulation further validates alignment with software predictions, showing deterministic latency and potential for FPGA optimizations. Together, these findings support the suitability of the model for real-time, resource-constrained UAV operations in solar inspection tasks.

CONCLUSION

This study presented a fixed-point Medium Gaussian SVM classifier optimized for thermal anomaly detection in PV modules using MPEG-7 shape descriptors. A complete pipeline was developed, starting from MATLAB-based preprocessing and SVM training to logic-level VHDL implementation for FPGA deployment. The classifier consistently achieved high accuracy and F1-scores across datasets of varying sizes, even attaining perfect classification in certain test sets. The conversion from floating-point to Q1.15 fixed-point arithmetic was essential to ensure compatibility with resource-constrained hardware platforms such as UAV-mounted FPGAs. The logic-based SVM implementation successfully replicated the behavior of the MATLAB model, as verified through detailed simulation, kernel output validation, and prediction agreement. Furthermore, execution latency remained within microsecond-scale limits, affirming its suitability for real-time detection during aerial PV inspections.

Overall, the proposed approach demonstrates that handcrafted, fixed-point SVM classifiers can maintain software-level accuracy while meeting the stringent performance, power, and area constraints of edge devices. This work provides a reliable and hardware-efficient solution for intelligent fault detection in solar energy systems, potentially contributing to improved PV module maintenance, reduced energy loss, and enhanced system longevity.

While the proposed fixed-point SVM implementation achieves strong performance and FPGA compatibility, several opportunities for future enhancement remain. Upcoming work will investigate dynamic reconfiguration and online retraining mechanisms, potentially through partial reprogramming of FPGAs or hybrid FPGA-CPU architectures. Integration with lightweight deep learning models or ensemble methods may further improve the detection of complex anomalies. Additionally, deploying the system on actual UAVs with real-time thermal imaging sensors will enable field validation and calibration against diverse irradiance and angle-of-view conditions. Power optimization techniques and resource-aware design will also be investigated to extend the endurance of UAV operations during continuous inspection missions. Hybrid approaches, such as edge–cloud collaboration, may offer a way to mitigate these trade-offs. As noted in [20], emerging paradigms like TinyML, Transformer variants, and architectures such as Mamba provide promising directions for balancing accuracy and efficiency in real-time applications.

ABBREVIATIONS

ACE: Automatic Content Extractor
AI: Artificial Intelligence
BGLR: Binary GLM (Generalized Linear Model) Logistic Regression
CV: Coefficient of Variation
DL: Deep Learning
FPGA: Field Programmable Logic Array
FSM: Finite State Machine
GHDL: G Hardware Description Language
GTKWave: GNU Tool Kit Wave
HDL: Hardware Description Language
IP: Intellectual Property
IR: Infrared
LUT: Look-Up Tables
ML: Machine Learning
MPEG: Moving Picture Experts Group
PV: Photovoltaic
RBF: Radial Basis Function

SVM: Support Vector Machine
UAV: Unmanned Aerial Vehicle
VHDL: VHSIC (Very High-Speed Integrated Circuit) Hardware Description Language

REFERENCES

[1] Nayomi Fernando, Lasantha Seneviratne, Nisal Weerasinghe, Namal Rathnayake, and Yukinobu Hoshino. 2025. Efficient hotspot detection in solar panels via computer vision and machine learning. Information 16, 7 (July 2025), 608. https://doi.org/10.3390/info16070608

[2] Muhammad Umair Ali, Hafiz Farhaj Khan, Manzar Masud, Karam Dad Kallu, and Amad Zafar. 2020. A machine learning framework to identify the hotspot in photovoltaic module using infrared thermography. Solar Energy 208 (2020), 643–651. https://doi.org/10.1016/j.solener.2020.08.027

[3] H. Hussein. 2023. Photovoltaic hotspot detection with image classification deep learning techniques. In Proceedings of the 2023 2nd International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI). IEEE, Zarqa, Jordan, 1–6. https://doi.org/10.1109/EICEEAI60672.2023.10590305

[4] Mahmoud Dhimish, Marios Theristis, and Vincenzo d'Alessandro. 2024. Photovoltaic hotspots: A mitigation technique and its thermal cycle. Optik 300 (2024), 171627. https://doi.org/10.1016/j.ijleo.2024.171627

[5] H. Açıkgöz, D. Korkmaz, and Ç. Dandıl. 2022. Classification of hotspots in photovoltaic modules with deep learning methods. TJST 17, 2 (2022), 211–221. https://doi.org/10.55525/tjst.1158854

[6] J. Wang, D. Gao, S. Zhu, S. Wang, and H. Liu. 2019. Fault diagnosis method of photovoltaic array based on support vector machine. Energy Sources, Part A: Recovery, Utilization, and Environmental Effects 45, 2 (2019), 5380–5395. https://doi.org/10.1080/15567036.2019.1671557

[7] C. J. Burges. 1998. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2, 2 (June 1998), 121–167. https://doi.org/10.1023/A:1009715923555

[8] Fengxi Zhang, Yuying Li, and Zhihao Ye. 2022. Apply YOLOv4-Tiny on an FPGA-based accelerator of convolutional neural network for object detection. In Proceedings of the 2nd International Conference on Artificial Intelligence and Industrial Technology Applications (AIITA 2022). IOP Publishing, Dali City, China, Article 012032. https://doi.org/10.1088/1742-6596/2303/1/012032

[9] Karuppiah Natarajan, Praveen Kumar B., and Vankadara Kumar. 2020. Fault detection of solar PV system using SVM and thermal image processing. International Journal of Renewable Energy Research 10 (2020), 967–977.

[10] X. Song, H. Wang, and L. Wang. 2014. FPGA implementation of a support vector machine based classification system and its potential application in smart grid. In Proceedings of the 2014 11th

International Conference on Information Technology: New Generations (ITNG). IEEE, Las Vegas, NV, 397–402. https://doi.org/10.1109/ITNG.2014.45

[11] Mateusz Wasala and Tomasz Kryjak. 2022. Real-time HOG+SVM based object detection using SoC FPGA for a UHD video stream. arXiv:2204.10619 [cs.CV]. https://doi.org/10.48550/arXiv.2204.10619

[12] A. Escobedo and A. Lin. 2016. Tessellation-based multi-block memory mapping scheme for high-level synthesis with FPGA. In Proceedings of the 2016 International Conference on Field-Programmable Technology (FPT). IEEE, Xi'an, China, 125–132. https://doi.org/10.1109/FPT.2016.7929517

[13] X. Ma, W. A. Najjar, and A. K. Roy-Chowdhury. 2015. Evaluation and acceleration of high-throughput fixed-point object detection on FPGAs. IEEE Transactions on Circuits and Systems for Video Technology 25, 6 (June 2015), 1051–1062. https://doi.org/10.1109/TCSVT.2014.2360030

[14] H. Brum, M. Véstias, and H. Neto. 2024. LiDAR 3D object detection in FPGA with low bitwidth quantization. In Proceedings of Applied Reconfigurable Computing. ARC 2024. Lecture Notes in Computer Science, vol. 14553. Springer, Cham, 90–105. https://doi.org/10.1007/978-3-031-55673-9_7

[15] Muhammad Ihsan Al Hafiz, Naresh Ravichandran, Anders Lansner, Pawel Herman, and Artur Podobas. 2025. A reconfigurable stream-based FPGA accelerator for Bayesian confidence propagation neural networks. arXiv:2503.01561 [cs.AR]. https://doi.org/10.48550/arXiv.2503.01561

[16] Srikanth Ramadurgam and Darshika G. Perera. 2021. An efficient FPGA-based hardware accelerator for convex optimization-based SVM classifier for machine learning on embedded platforms. Electronics 10, 11 (May 2021), 1323. https://doi.org/10.3390/electronics10111323

[17] Shereen Afifi, Hamid GholamHosseini, and Roopak Sinha. 2019. A system on chip for melanoma detection using FPGA-based SVM classifier. Microprocessors and Microsystems 65 (2019), 57–68. https://doi.org/10.1016/j.micpro.2018.12.005

[18] Shereen Afifi, Hamid GholamHosseini, and Roopak Sinha. 2015. Hardware implementations of SVM on FPGA: A state-of-the-art review of current practice. International Journal of Innovative Science, Engineering & Technology (IJISET) 2 (2015), 733–752.

[19] Mohammed H. Yacoub, Samar M. Ismail, Lobna A. Said, Ahmed H. Madian, and Ahmed G. Radwan. 2024. Support vector machine reconfigurable hardware implementation on FPGA. Franklin Open 7 (2024), 100115. https://doi.org/10.1016/j.fraope.2024.100115

[20] T. Suwannaphong, F. Jovan, I. Craddock, et al. 2025. Optimising TinyML with quantization and distillation of transformer and mamba models for indoor localisation on edge devices. Scientific Reports 15 (March 2025), 10081. https://doi.org/10.1038/s41598-025-94205-