Decoding UXO Visual Signatures: A Comparative Study of Feature Extraction and Classifiers

Gheorghe Marian Craioveanu
Automation and Industrial Informatics
University Politehnica of Bucharest
Bucharest, Romania
gheorghe.craioveanu@upb.ro

Grigore Stamatescu
Automation and Industrial Informatics
University Politehnica of Bucharest
Bucharest, Romania
grigore.stamatescu@upb.ro

***Abstract*—Classification of unexploded ordnance (UXO) is a critical-safety application in which misclassification can endanger both operators and civilians. Although deep learning methods have achieved strong results in general instance classification, their deployment on resource-constrained edge devices remains challenging due to high memory requirements, inference latency, and interpretability. Because UXOs are often located in remote or inaccessible areas, these models need to be deployed on embedded devices. This paper presents a comparative study of classical computer vision pipelines designed for UXO classification. Six feature extraction methods: raw pixel intensities, HOG, LBP, SIFT with Bag-of-Visual-Words, Gabor filters, and Haar features, are combined with five machine learning classifiers, namely: Logistic Regression, SVM, Random Forest, KNN, and XGBoost, resulting in 30 distinct pipeline configurations. These pipelines are evaluated in terms of accuracy, F1-score, inference latency, memory footprint, and power consumption on a NVIDIA Jetson Xavier NX. The findings show that the Gabor+SPM–XGBoost pipeline offers the best accuracy–latency trade-off, achieving 83.3% accuracy with an inference latency of 171.7 ms, and is therefore identified as the most suitable configuration for real-world field deployment.**

***Index Terms*—unexploded ordnance, feature extraction, machine learning, edge deployment, image classification**

I. Introduction

Unexploded Ordnance (UXO), also known as Explosive Remnants of War (ERW), constitutes a persistent global threat that affects post-conflict zones in multiple countries. According to the United Nations Mine Action Service [1], UXO contamination affects more than 70 countries, with thousands of civilian casualties reported annually. The identification and disposal of these hazards, a procedure known as Explosive Ordnance Disposal (EOD), is highly dependent on the expertise of human technicians who must make decisions under pressure and time constraints. Operational deployments occur frequently in field conditions where cloud connectivity is unavailable, power sources are limited, and form factors are severely constrained, precisely the characteristics that define edge computing environments [2]. To address this, deep neural networks (DNNs) can represent a viable option. Deep learning approaches have achieved state-of-the-art results in general instance classification [3], but deploying modern models such as Vision Transformer on single-board computers remains a challenge. Classical computer vision pipelines offer good computational metrics, small memory footprints, and predictable inference times. Classical feature extractors achieve strong performance in recognising textures and shapes, and when paired with lightweight classifiers, they can operate in real time on edge devices. However, for unexploded ordnance, no existing benchmark was found that simultaneously evaluates different feature extractor families together with various classifier types while evaluating both accuracy and hardware constraints relevant to edge UXO classification. This paper fills that gap using the CTX-UXO dataset [4], a comprehensive public collection of UXO images labelled by an EOD specialist in various visual contexts. Figure 1 exemplifies a classical image processing approach based on HOG [5], applied to a CTX-UXO sample. The main contributions of this manuscript are:

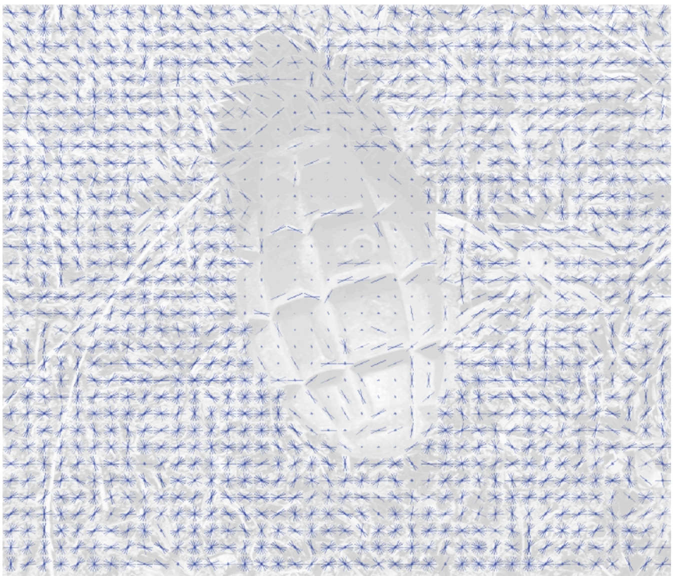

Fig. 1. HOG descriptor results showing a detailed pattern of gradient orientations that represent the distinctive shape and surface texture of the UXO instance

- A comparison of six feature extraction techniques combined with five classifiers, resulting in 30 distinct pipeline

The authors gratefully acknowledge the financial support provided by the University of Science and Technology POLITEHNICA of Bucharest for the publication of this article through the PubArt programme.

configurations assessed on the CTX-UXO dataset;
- A hardware benchmark on the NVIDIA Jetson Xavier NX, a representative and modern edge platform;
- A detailed analysis of the tradeoff between primary and secondary metrics, providing practical recommendations for deploying AI systems for UXO mitigation;

This work advances the state of the art mainly by offering a benchmark and an in-depth, hardware-focused study. The rest of the manuscript is structured as follows. Section II reviews related work on classical feature extractors and classifiers. Section III describes the pipeline, the preprocessing steps, the methodology, and the experimental setup. Section IV reports primary metrics, secondary metrics, and comparisons. Section V presents the conclusions.

II. Related Work

UXO detection research has traditionally focused on geophysical sensing, such as magnetometry [6], [7] and ground-penetrating radar [8], [9], while computer vision methods were limited in usage due to the lack of representative datasets. The authors of the current manuscript addressed this by introducing the CTX-UXO dataset [4], with more than 10,000 downloads since its initial publication in 2024. The dataset has been validated in several previous studies, such as [10]–[13].

Early visual recognition methods based on template matching and global colour histograms often fail under realistic changes in viewpoint and illumination [14], as in the CTX-UXO dataset [4]. Histogram of Oriented Gradients (HOG) [5] improves robustness by capturing local gradient orientations within normalised blocks and became widely used for rigid-object detection and deformable part models. However, HOG assumes a fixed aspect ratio, is sensitive to image rotation, and does not explicitly encode texture. These shortcomings are critical for UXO, where object poses vary greatly and fine-grained surface texture is crucial for distinguishing visually similar ordnance types. Texture descriptors such as Local Binary Patterns (LBP) [15] address these challenges by representing local micro-textures through thresholded circular neighbourhoods, consisting of 0 and 1, based on the pixel intensity, producing computationally efficient histograms that remain robust under varying illumination. The uniform LBP variant further improves robustness to noise and imaging artefacts e.g., corrosion, partial burial, common to UXO. However, using a single global LBP histogram removes the spatial layout, limiting discrimination between objects with similar textures but different shapes. Spatial Pyramid Matching (SPM) [16] addresses this by partitioning the image into multi-scale subregions and concatenating their histograms, reintroducing coarse spatial structure at a reduced additional cost. For this study, SPM is applied to both the LBP and Gabor descriptors. Unlike LBP, which relies on one circular neighbourhood and simple binary intensity comparisons, Gabor filters [17] provide a combined spatial–frequency analysis over various scales and orientations, thereby closely approximating the response characteristics of basic orientation-selective cells. Global Gabor statistics are sensitive to intra-class spatial variation. SPM mitigates this sensitivity by preserving local Gabor statistics. Viola and Jones [18] demonstrated that Haar-like features enable fast encoding of rectangular contrast patterns, supporting real-time face detection. Although efficient in resource-constrained devices, Haar-like features mainly capture rectangular patterns and rely on PCA for dimensionality reduction. To mitigate the fixed-grid limitations of HOG and Haar, Scale-Invariant Feature Transform (SIFT) [19] detects keypoints as scale-space extrema in a Difference-of-Gaussian pyramid and represents each with a gradient orientation descriptor, providing scale and rotation invariance. This invariance is relevant for UXO instances, where images are acquired from various consumer devices at different distances and viewing angles. Since SIFT descriptors produce variable-size keypoint sets, in this study, a Bag-of-Visual-Words (BoVW) representation [20] is used to obtain fixed-length vectors.

Among classical classification methods, the Support Vector Machine (SVM) [21] continues to be a robust option for moderate dimensional feature spaces, exploiting kernel-based margin maximisation, and has been effectively used with HOG and SIFT descriptors in classification benchmarks [22]. Despite the need for careful feature normalisation and an inference cost that increases with the number of support vectors, Random Forests [23] offer a complementary, scale-invariant ensemble approach with intrinsic feature-importance estimates, which makes them well-suited to very high-dimensional descriptors such as HOG. Gradient Boosting algorithms, with XGBoost [24], deliver competitive results on tabular classification tasks by iteratively correcting residual errors, whereas k-Nearest Neighbours (KNN) [25] functions as a simple non-parametric reference method whose accuracy is expected to degrade on high-dimensional descriptors. In this manuscript, Logistic Regression is included as a linear baseline providing interpretable decision boundaries, enabling direct comparison with more expressive non-linear models. All detailed methods, both extractors and classifiers, are included in Section III.

III. Methodology

A. Dataset and Preprocessing

This study uses the CTX-UXO (Contextual Vision for Unexploded Ordnances) dataset [4], an open-access collection available on IEEE DataPort. It contains 15,449 real-world images of ordnance. The images are RGB JPEG files with a median resolution of 2,124×2,124 pixels, acquired primarily through the use of smartphone devices of the consumer-level. The dataset supports both binary (UXO vs. non-UXO) and multi-class classification problems (multiple UXO types). In this study, the focus is on **UXO / NON-UXO binary classification**. The final dataset includes an equal number of UXO and non-UXO instances. The non-UXO samples are generated by selecting regions from the CTX-UXO images that do not overlap with any labelled bounding boxes. The data is pre-split into training sets (70%), validation sets (15%), and test sets (15%). All performance metrics reported in this manuscript are computed on the test set. Each UXO instance is first isolated from the full image using the provided bounding box annotations. The bounding box crop is then symmetrically padded to a square aspect ratio before resizing in order to preserve the original object geometry. The input size for feature extractors is a square 128×128 grayscale input, unless otherwise stated in Section III-B. The decision to convert the images to grayscale is based on real-world scenarios. In field settings, the colour data are extremely inconsistent and unreliable due to rust, dirt, mud, shadows, and different levels of paint wear. As a result, colour has no useful discriminative information for this classification task. For classifiers that are sensitive to absolute feature magnitude, specifically SVM and Logistic Regression, a *StandardScaler* is additionally applied to the feature vectors after descriptor extraction. For interpretability, no data augmentation is performed. As a note, the training set is representative enough (real-world UXOs).

B. Feature Extraction Methods

The study evaluates six distinct feature representations, spanning a range of complexity from raw pixel baselines to multi-scale keypoint encodings, as detailed in Section II. Table I reports the input resolutions, the output dimensionalities, and the computational complexity.

TABLE I
SUMMARY OF FEATURE EXTRACTION METHODS

Method	Input size	Output Dimension	Complexity
Raw Pixels	128×128	256	$\mathcal{O}(N)$
LBP	128×128	295	$\mathcal{O}(NP)$
Gabor	128×128	400	$\mathcal{O}(N{\cdot}SO)$
HOG	128×128	8,100	$\mathcal{O}(N)$
Haar-like	128×128	256	$\mathcal{O}(1)$/feature
SIFT + BoVW	512×512	512	$\mathcal{O}(N\log N)$

For baseline settings, every 128×128 grayscale image is first flattened into a 16,384-dimensional vector. Principal Component Analysis (PCA) is then performed, retaining $n = 256$ components. To address the complete lack of structural encoding in the raw pixel baseline, Local Binary Patterns (LBP) [15] introduce a lightweight texture descriptor based on local intensity comparisons. For each pixel (x, y) with central intensity g_c, the LBP is computed [15] over circularly sampled neighbours P at radius r:

$$\mathrm{LBP}_{P,r}(x,y) = \sum_{p=0}^{P-1} s(g_p - g_c)\cdot 2^p, \quad s(u) = \begin{cases} 1 & u \geq 0 \\ 0 & u < 0 \end{cases} \tag{1}$$

In Eq. 1, g_p is the intensity of the p-th neighbour. The uniform LBP variant (binary, 0 or 1) is used to reduce noise sensitivity, producing a histogram per region ($P = 8$, $r = 1$). To overcome LBP's widely recognized deficiency in capturing spatial structure [15], Spatial Pyramid Matching (SPM) [16] is applied. The 128×128 image is partitioned into a 1×1 grid (whole image) and a 2×2 grid (four quadrants), giving 5 non-overlapping regions. An LBP histogram is computed independently in each region, and the five histograms are concatenated, yielding 295 dimensions (5×59). To address the limitation of the LBP method to a single circular neighbourhood, Gabor filters offer a framework for combined spatial and frequency analysis over a range of scales and orientations. A 2-D Gabor filter is characterized by a Gaussian-shaped envelope that is multiplied by an underlying sinusoidal carrier wave:

$$g(x,y;\lambda,\theta,\sigma,\gamma) = \exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right)\cos\left(\frac{2\pi x'}{\lambda}\right) \tag{2}$$

In Eq. 2, $x' = x\cos\theta + y\sin\theta$, $y' = -x\sin\theta + y\cos\theta$, λ is the sinusoidal wavelength, θ is the orientation, $\sigma = 0.56\lambda$ is the Gaussian standard deviation (standard value) and $\gamma = 0.5$ is the spatial aspect ratio. A filter bank of 5 scales ($\lambda \in \{4, 8, 16, 32, 64\}$ pixels) and 8 orientations ($\theta \in \{0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135°, 157.5°\}$) is convolved with the 128×128 grayscale image, producing 40 response maps. To retain spatial layout information, a weakness of Gabor statistics [17], the same SPM scheme is applied. The mean and standard deviation of each response map are extracted within each region, resulting in $5 \times 40 \times 2 = 400$ dimensions.

The Histogram of Oriented Gradients (HOG) captures the distribution of gradient directions over a grid of local cells, thereby encoding spatial structure directly in the descriptor. Horizontal and vertical gradients are computed using Sobel filters, from which the magnitude of the gradient m and the unsigned orientation θ are derived for each pixel (x, y).

$$m(x,y) = \sqrt{G_x^2 + G_y^2} \tag{3}$$

$$\theta(x,y) = \arctan\left(\frac{G_y}{G_x}\right) \in [0°, 180°) \tag{4}$$

The HOG descriptor is calculated using the following settings: orientations = 9, pixels_per_cell = (8, 8), cells_per_block = (2, 2). This results in 8100 dimensions.

Haar-like features [18] make use of integral images to evaluate rectangular intensity contrasts in $\mathcal{O}(1)$ time per feature. The integral image $img(x, y)$ at location (x, y) contains the sum of the pixels above and to the left of (x, y), defined as:

$$img(x,y) = \sum_{x' \leq x, y' \leq y} img(x', y') \tag{5}$$

In Eq. 5, $img(x', y')$ is the original image intensity. The original input for Haar-like features uses a 64×128 pedestrian-orientated size [18]. Since UXO objects do not have a canonical aspect ratio (as pedestrians), for the current task, the implementation uses a 128×128 square input instead (see Section III-A). Five types of canonical features are computed at multiple scales over the 128×128 grayscale input, resulting in a high-dimensional intermediate representation. PCA is applied to reduce this to 256 components. To avoid dependence on a fixed spatial structure, SIFT (Scale-Invariant Feature Transform) [19] locates keypoints as scale-space extrema via the Difference-of-Gaussian (DoG) function $D(x, y, \sigma)$, defined as the difference between two Gaussian-blurred images at scales separated by a constant factor k:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (6)$$

Here, $G(x, y, \sigma)$ is a Gaussian kernel with variable scale and $I(x, y)$ is the original image. The contrast threshold is 0.04. Since SIFT produces variable-length keypoint sets, a Bag-of-Visual-Words (BoVW) [20] encoding is used to create a fixed-length representation. A visual vocabulary of K=512 words is constructed by applying k-means clustering [26], using only SIFT descriptors from the training set to avoid leakage from validation or test data. The value K=512 is determined by examining the k-means distortion curve and selecting the elbow point [27].

C. Classifiers

Five classifiers are evaluated, ranging from linear to non-linear and from instance-based to ensemble methods. All hyperparameters were determined through 5-fold stratified cross-validation applied to the training set. The Support Vector Machine (SVM) algorithm identifies the separating hyperplane that maximizes the margin. Using the Radial Basis Function (RBF) kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$, the resulting decision function is:

$$f(\mathbf{x}) = \operatorname{sign}\left(\sum_{i \in \mathcal{S}} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right)$$

For this equation, configuration is: $C = 10$; $\gamma = \gamma_{\text{scale}} = 1/(d \cdot \mathrm{Var}(\mathbf{X}_{\text{train}}))$ where d is the number of features. The study also utilizes a random forest classifier composed of an ensemble of $T = 200$ decision trees, where each tree is trained on a bootstrap-resampled subset of the training set. The Gini impurity [28] serves as a splitting criterion. The Gini impurity for a given node m is calculated as:

$$G(m) = 1 - \sum_{i=1}^{C} p_i^2 \quad (7)$$

In Eq. 7, C is the total number of classes and p_i is the fraction of instances belonging to class i at node m. The depth of each tree is restricted to a maximum of 20 levels. A KNN classifier is also used. The classification is carried out by majority vote among the nearest neighbours $k = 7$, where the contribution of each neighbour is weighted by the inverse of its Euclidean distance. Formally, the predicted class $\hat{y}$ for a query instance x is determined by:

$$\hat{y} = \arg\max_{c} \sum_{i=1}^{k} \frac{1}{d(x, x_i)} I(y_i = c) \quad (8)$$

In Eq. 8, $d(x, x_i)$ represents the Euclidean distance between the query point and its i-th nearest neighbour, and $I(\cdot)$ is the indicator function. Logistic regression (LR) models the conditional probability P(y = 1 | x) = $\sigma(w^{\mathrm{T}} x + b)$ using a sigmoid activation function. The model is configured with L2 regularisation, C = 1.0, the L-BFGS optimiser, a maximum of 1,000 iterations and a one vs all strategy. This approach (one vs all strategy) has previously proven effective in the authors' work on UXO localisation and classification [10]. Input feature vectors are standardised with StandardScaler before training. XGBoost uses a gradient-boosted ensemble of regression trees. At every iteration t, the algorithm optimises the regularised objective function in order to obtain the new tree f_t:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \quad (9)$$

In Eq. 9, l is a differentiable convex loss function measuring the difference between the target y_i and the current prediction $\hat{y}_i^{(t-1)}$, and $\Omega(f_t)$ penalizes the complexity of the tree. In this study, it is configured with 200 estimators, a maximum tree depth of 6, a learning rate of $\eta = 0.1$, and a subsample ratio of 0.8.

D. Evaluation Metrics

All classifiers are evaluated on the test set using the following metrics: overall accuracy, defined as the proportion of correctly classified samples; F1-score, computed as the harmonic mean of precision and recall; ROC-AUC, the Area Under the Receiver Operating Characteristic Curve, obtained by sweeping the decision threshold over classifier posterior probabilities (or decision scores), providing a threshold-independent measure of discrimination that is more informative than accuracy for safety-critical binary tasks; inference latency, defined as the median forward-pass time over all images; peak RAM usage. The target hardware platform is the NVIDIA Jetson Xavier NX, which was operated in the 15 W power mode throughout all experiments and features a 6-core Carmel ARM v8.2 CPU running at 1.9 GHz, a 384-core Volta GPU with 48 Tensor Cores, and 8 GB of unified LPDDR4x memory. NVIDIA Jetson is the leading AI computing platform for GPU-accelerated parallel processing in embedded systems, making it well suited to field-deployable UXO classification. Hyperparameters were selected using 5-fold stratified cross-validation applied to the training data. All metrics presented were calculated on the predefined test set.

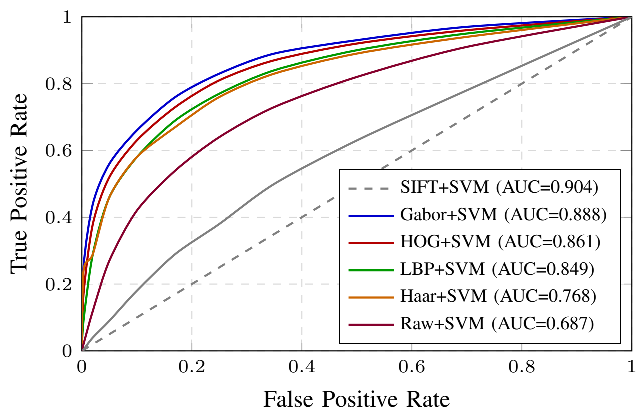

Fig. 2. ROC curves for the six best-XGBoost pipelines on the binary (UXO / non-UXO) test partition.

TABLE II
TEST-SET PERFORMANCE FOR ALL 30 PIPELINE CONFIGURATIONS. ACC = ACCURACY; F1 = MACRO-F1; AUC = ROC-AUC (REPORTED IN [0,1]). BOLD INDICATES THE HIGHEST ACCURACY FOR EACH FEATURE EXTRACTOR.

	SVM			**RF**			**KNN**			**LR**			**XGBoost**		
Extractor	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC
Raw+PCA	64.00	64.00	0.687	69.50	69.16	0.757	64.50	64.15	0.733	60.67	60.62	0.648	**69.50**	69.47	0.759
LBP+SPM	76.67	76.65	0.849	**78.67**	78.65	0.869	67.67	67.63	0.745	70.17	70.08	0.744	77.00	76.95	0.863
Gabor+SPM	80.50	80.46	0.888	80.33	80.31	0.887	62.17	62.16	0.690	75.50	75.42	0.807	**83.33**	83.32	0.903
HOG	**77.67**	77.63	0.861	72.17	72.07	0.814	63.33	60.67	0.762	70.00	69.95	0.725	71.67	71.54	0.822
Haar+PCA	72.00	71.94	0.768	70.33	69.55	0.781	70.33	70.11	0.791	58.33	58.33	0.596	**73.33**	73.32	0.807
SIFT+BoVW	81.67	81.66	0.904	80.00	80.00	0.878	79.83	79.83	0.861	75.00	74.88	0.840	**82.67**	82.66	0.896

IV. RESULTS

Table II reports accuracy, macro-averaged F1-score, and ROC-AUC for all 30 (feature extractor $\times$ classifier) configurations on the test set. The best overall result is achieved by Gabor+SPM+XGBoost, followed closely by SIFT+BoVW+XGBoost. XGBoost is the top-performing classifier on the majority of feature extractors. SIFT+BoVW shows the most consistent performance across all five classifiers. The Raw+PCA baseline confirms that PCA-reduced raw pixels cannot substitute for structured feature engineering. The SIFT+BoVW+SVM configuration achieves the highest overall ROC-AUC score among all 30 pipeline configurations, demonstrating a superior threshold-independent classification capability that is highly desirable in safety-critical unexploded ordnance identification/classification. Fig. 2 reports ROC curves for the top SVM pipelines (one per feature extractor). Table III presents 5-fold stratified cross-validation accuracy (mean $\pm$ std), used for hyperparameter selection. Gabor+SPM+SVM achieves the highest CV accuracy (89.8%) with moderate variance ($\pm$1.3 pp). LBP+SPM+SVM displays a similarly high CV accuracy (88.6%) with identical variance. XGBoost-based pipelines exhibit notably low variance (Gabor+SPM+XGBoost: $\pm$0.5 pp; LBP+SPM+XGBoost: $\pm$0.2 pp), indicating particularly stable generalisation across data splits. The performance of Support Vector Machines

TABLE III
STRATIFIED 5-FOLD CROSS-VALIDATION ACCURACY (%) ON THE TRAINING PARTITION (MEAN $\pm$ STD). THE SIX BEST-PERFORMING PIPELINES ARE REPORTED.

Pipeline	**Mean Accuracy (%)**	**Std. (%)**
Gabor+SPM+SVM	89.8	$\pm$1.3
LBP+SPM+SVM	88.6	$\pm$1.3
LBP+SPM+XGBoost	88.0	$\pm$0.2
Gabor+SPM+XGBoost	87.9	$\pm$0.5
HOG+SVM	87.3	$\pm$0.9
SIFT+BoVW+XGBoost	87.0	$\pm$1.2

depends heavily on the dimensionality of the feature space. Fig. 3 plots test accuracy (best SVM result per extractor) against feature vector dimensionality on a logarithmic scale. The relationship is non-monotonic: SIFT+BoVW achieves the highest SVM accuracy (81.7%) at 512 dimensions, while HOG delivers only 77.7% despite the largest dimensionality (8,100). Raw+PCA achieves 64.0% at 256 dimensions, which is the weakest overall performance. Among the structured descriptors, Gabor+SPM (80.5% at 400 feature dimensionality) provides the best balance between accuracy and dimensionality.

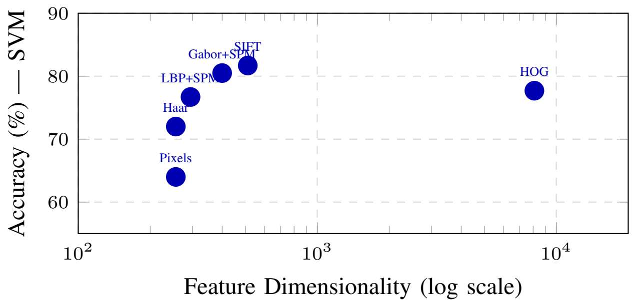

Fig. 3. SVM accuracy vs. feature dimensionality. SIFT+BoVW achieves the best SVM accuracy at 512 dimensions.

Table IV presents the inference latency, and process RAM measured on NVIDIA Jetson Xavier NX. Metrics are calculated as median over 100 experiments. Logistic Regression provides the fastest inference across all feature extractors (1.6 ms), followed by SVM and XGBoost. The exception is HOG+SVM at 55.6 ms, due to the large number of support vectors induced by the 8,100-dimensional descriptor. Random Forest is the slowest classifier due to ensemble tree traversal, while KNN latency grows with feature dimensionality. Principal findings of the evaluation are summarised next:

1) **Accuracy is strongly affected by both the feature extractor and the classifier.** The performance variation across classifiers for a fixed feature extractor can be as large as or larger than the variation across feature extractors for a fixed classifier (e.g., XGBoost). This shows that feature extractors and classifiers should be jointly optimized rather than tuned independently.
2) **Best accuracy–efficiency trade-off: LBP+SPM and HOG.** Methods based on Local Binary Patterns with Spatial Pyramid Matching, as well as those using Histograms of Oriented Gradients, achieve the most favourable compromise. On the embedded NVIDIA Jetson platform, these approaches offer a robust balance between the primary metric (accuracy) and secondary metrics (such as computational cost and latency).

3) **Highest accuracy: Gabor Filters with Spatial Pyramid Matching and an XGBoost classifier.** When accuracy is treated as the primary objective, the combination Gabor+SPM+XGBoost consistently delivers the best results and stands out as the top-performing pipeline.

TABLE IV
SECONDARY METRICS MEASURED ON NVIDIA JETSON XAVIER NX

Pipeline	Latency (ms)	Peak RAM (MB)
Raw+PCA+LR	1.6 ± 0.1	193.5 ± 0.1
Raw+PCA+XGBoost	2.0 ± 0.1	193.5 ± 0.1
Raw+PCA+SVM	4.5 ± 0.1	196.0 ± 0.1
Haar+PCA+LR	6.1 ± 0.1	189.1 ± 0.1
Haar+PCA+XGBoost	6.5 ± 0.1	189.3 ± 0.1
Haar+PCA+SVM	7.5 ± 0.1	191.5 ± 0.1
LBP+SPM+LR	10.1 ± 0.1	185.0 ± 0.1
LBP+SPM+XGBoost	10.5 ± 0.2	197.8 ± 0.0
LBP+SPM+SVM	11.5 ± 0.2	185.0 ± 0.1
LBP+SPM+KNN	14.3 ± 0.1	197.8 ± 0.0
Haar+PCA+KNN	18.0 ± 0.3	191.0 ± 0.1
Raw+PCA+KNN	21.5 ± 0.3	195.0 ± 0.1
HOG+LR	30.7 ± 0.3	185.0 ± 0.1
HOG+XGBoost	31.1 ± 0.1	190.0 ± 0.0
HOG+SVM	55.6 ± 1.1	309.0 ± 0.1
Haar+PCA+RF	59.0 ± 1.8	194.5 ± 0.1
Raw+PCA+RF	60.7 ± 1.8	202.0 ± 0.1
LBP+SPM+RF	88.2 ± 2.6	189.4 ± 0.1
HOG+RF	104.2 ± 3.1	188.9 ± 0.1
Gabor+SPM+LR	171.3 ± 1.7	185.0 ± 0.1
Gabor+SPM+XGBoost	171.7 ± 0.5	189.7 ± 0.0
Gabor+SPM+SVM	172.7 ± 3.5	188.0 ± 0.1
HOG+KNN	183.6 ± 0.2	282.7 ± 0.0
Gabor+SPM+KNN	212.6 ± 6.6	188.0 ± 0.2
Gabor+SPM+RF	240.8 ± 7.2	190.0 ± 0.1
SIFT+BoVW+LR	280.1 ± 2.8	193.0 ± 0.1
SIFT+BoVW+XGBoost	280.5 ± 5.6	193.0 ± 0.1
SIFT+BoVW+SVM	283.0 ± 5.7	198.0 ± 0.1
SIFT+BoVW+KNN	302.0 ± 4.5	197.0 ± 0.1
SIFT+BoVW+RF	333.8 ± 10.0	200.0 ± 0.1

V. CONCLUSION

This paper presents a benchmark of classical computer vision pipelines for UXO classification using the CTX-UXO dataset. Among 30 feature extractor–classifier combinations, Gabor+SPM+XGBoost emerged as the top configuration with 83.3% accuracy, an AUC of 0.903. XGBoost was the top-performing classifier for four of six feature extractors. Hardware benchmarking on the Jetson Xavier NX confirmed that most tested pipelines satisfy real-time constraints, making classical feature extraction a viable option for field-deployed EOD platforms. Future work will compare quantised lightweight CNNs, such as MobileNetV3-Small in INT8 format.

REFERENCES

[1] United Nations Mine Action Service, "United nations mine action service — threat of landmines and explosive remnants of war," 2026, accessed: 2026-05-10. [Online]. Available: https://unmas.org/en

[2] H. Zhai, J. Du, Y. Ai, and T. Hu, "Edge deployment of deep networks for visual detection: a review," *IEEE Sensors Journal*, vol. 25, 2024.

[3] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE communications surveys & tutorials*, vol. 22, 2020.

[4] G. M. Craioveanu and G. Stamatescu, "Ctx-uxo: A comprehensive dataset for detection and identification of unexploded ordnances," 2024. [Online]. Available: https://dx.doi.org/10.21227/cwnm-de53

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2005, pp. 886–893.

[6] K. M. Churchill, C. Link, and C. C. Youmans, "A comparison of the finite-element method and analytical method for modeling unexploded ordnance using magnetometry," *IEEE transactions on geoscience and remote sensing*, vol. 50, no. 7, pp. 2720–2732, 2011.

[7] P. Ściegienka and M. Blachnik, "On the development of a neural network architecture for magnetometer-based uxo classification," *Applied Sciences*, vol. 15, no. 15, p. 8274, 2025.

[8] L. E. Besaw, "Detecting buried explosive hazards with handheld gpr and deep learning," in *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXI*, vol. 9823. SPIE, 2016, pp. 187–197.

[9] A. Bredeck, V. Schmidt, and J.-P. Schmoldt, "Novel approaches of borehole-gpr data processing and visualization–application for uxo detection," *Near Surface Geophysics*, vol. 22, no. 4, pp. 482–489, 24.

[10] M. G. Craioveanu and G. Stamatescu, "Detection and identification of unexploded ordnance using a two-step deep learning methodology," in *2024 32nd Mediterranean Conference on Control and Automation (MED)*. IEEE, 2024, pp. 257–262.

[11] G.-M. Craioveanu and G. Stamatescu, "Evaluation of the robustness-runtime efficiency trade-off of edge ai models in uxo localisation and classification," in *2025 33rd Mediterranean Conference on Control and Automation (MED)*, 2025.

[12] M. Craioveanu, G. Stamatescu, and D. Popescu, "Ensemble strategy with multi-step hard sample mining for improved uxo localisation and classification," *IEEE Access*, 2025.

[13] G. M. Craioveanu, G. Stamatescu, and O. Saukh, "Look, reason, defuse: Bridging perception and domain knowledge for real-world unexploded ordnance identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2026, pp. 9113–9124.

[14] J. R. Parker, *Algorithms for image processing and computer vision*. John Wiley & Sons, 2010.

[15] M. Pietikäinen, "Local binary patterns," *Scholarpedia*, vol. 5, no. 3, p. 9775, 2010.

[16] S. Lazebnik, C. Schmid, J. Ponce *et al.*, "Spatial pyramid matching," *Object Categorization: Computer and Human Vision Perspectives*, vol. 3, no. 4, p. 2009, 2009.

[17] R. Mehrotra, K. R. Namuduri, and N. Ranganathan, "Gabor filter-based edge detection," *Pattern recognition*, vol. 25, no. 12, pp. 1479–1494, 92.

[18] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2001, pp. 511–518.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[20] H. Kato and T. Harada, "Image reconstruction from bag-of-visual-words," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 955–962.

[21] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[22] S. Bakheet, "An svm framework for malignant melanoma detection based on optimized hog features," *Computation*, vol. 5, no. 1, 2017.

[23] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[24] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 785–794.

[25] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "Knn model-based approach in classification," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2003.

[26] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[27] A. J. Onumanyi, D. N. Molokomme, S. J. Isaac, and A. M. Abu-Mahfouz, "Autoelbow: An automatic elbow detection method for estimating the number of clusters in a dataset," *Applied Sciences*, 2022.

[28] R. A. Disha and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using gini impurity-based weighted random forest (giwrf) feature selection technique," *Cybersecurity*, vol. 5, no. 1, p. 1, 2022.