Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

YOLOv8 Screw Detection Performance variation with the Composition of Real and Synthetic Training Data

Dennis Bäcker

Dennis.Baecker@aas.tu-freiberg.de

TU Bergakademie Freiberg

Amir Nemati

TU Bergakademie Freiberg

Krishna Kumar Murali

TU Bergakademie Freiberg

Robert Weidner

TU Bergakademie Freiberg

Research Article

Keywords:

Posted Date: November 18th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11600216/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Real versus Synthetic Training Data for YOLOv8-Based Screw Detection: A Sim-to-Real Study with FCM-ANFIS Post-Classification

D. Bäcker[1], A. Nemati[1], K. K. Murali[1], R. Weidner[1]

[1]Chair for Automated and Autonomous Systems, TU Bergakademie Freiberg, Bernhard-von-Cotta-Str. 4, 09599,Freiberg,Germany.

Contributing authors: Dennis.Baecker@aas.tu-freiberg.de;
Amir.Nemati@aas.tu-freiberg.de;
Krishna-Kumar.Murali@student.tu-freiberg.de;
robert.weidner@aas.tu-freiberg.de;

Abstract

Deep learning object detectors are useful for automatic inspection, but they normally need many labelled images. Collecting real images of industrial parts takes time, while synthetic images can be produced more easily. This work studies how the amount of real and synthetic training data affects a YOLOv8s screw detector. Two screw classes, M3 × 8 and M3 × 12, were used. Eight valid combinations were evaluated using real data levels of 0%, 20% and 50% and synthetic data levels of 0%, 50% and 100%. Each combination was trained twice, using random seeds 0 and 1, while the same 30% real test set was kept fixed. A small hyperparameter study was conducted before the main runs and selected SGD, a learning rate of 1×10^{-3} and a batch size of 32. The best two-seed mean overall detection-classification score was 83.74% with 50% real and 100% synthetic training data. Synthetic-only training produced substantially lower detection-classification scores, showing a clear synthetic-to-real gap. At 50% real data, adding synthetic images gave a small increase in the mean detection-classification score, while at 20% real data the detection-classification score decreased as more synthetic images were added. In the second stage, Fuzzy C-Means (FCM) and an Adaptive Neuro-Fuzzy Inference System (ANFIS) were applied to seven features extracted from matched YOLO detections. The three-rule hybrid model achieved 82.35% classification accuracy and an F1 score of 80.85%, compared with 80.39% and 79.45% for the YOLO classification baseline on the same matched test detections. The results show that synthetic data can help, but its effect depends on the amount of real data and on the remaining domain gap. However, the

hybrid model's accuracy improvement was not statistically significant according to McNemar's test ($p = 0.629$)

Keywords: screw detection; YOLOv8; Fuzzy C-Means; ANFIS; industrial vision

1 Introduction

Screws and other small fastening parts are used in many assembly processes. In an automated assembly line, the system must verify not only whether a screw is present, but also whether the correct screw type has been supplied. Manual inspection may be adequate for small production volumes, but it becomes time-consuming when many parts must be inspected. A camera-based system can perform this inspection without touching the part and can also be integrated with a robot or sorting unit. Deep-learning-based object detectors have facilitated the development of automated inspection systems. The YOLO family is widely used because it performs object localization and classification within a single-stage framework and supports fast inference [1, 2]. A major challenge is the availability of suitable training data. The training data must capture sufficient variation in position, orientation, lighting, background and image quality before the trained model can work reliably on new images. For industrial parts, collecting and labelling a large real dataset is time-consuming. Nowruzi et al. [3] showed that synthetic data can replace part of the real data, but its effectiveness depends on how closely the synthetic images represent the real-image domain. Ljungqvist [4] concluded that there is no universally optimal real-to-synthetic ratio that works for every object detection problem. These findings motivate a joint investigation of the amounts of real and synthetic training data rather than the assumption of a fixed ratio.

The present work has two connected parts. First, a YOLOv8s screw detector is trained using different amounts of real and synthetic data. The same real test set is used for every run, and the measured results are visualized over the measured experimental grid. Before the main grid is trained, a preliminary hyperparameter study is used to select the optimizer, learning rate and batch size. Second, features extracted from matched YOLO detections are used as inputs to an FCM-initialized ANFIS model to evaluate whether the class decision can be improved. The study addresses three questions: (1) To what extent does the amount of real training data affect performance on real test images? (2) Does increasing the amount of synthetic training data consistently improve performance? and (3) Can an FCM–ANFIS post-classification stage improve the class predictions produced by YOLO for matched detections?

2 Related Work

2.1 Synthetic data and the synthetic to real gap

Synthetic images are useful because large numbers of labeled samples can be generated without manually capturing each image. A major challenge is that rendered images differ from real camera images. Reflections, shadows, textures, sensor noise and small surface details may differ between the two domains. This discrepancy is commonly referred to as the synthetic-to-real, or sim-to-real, gap. Domain randomization is one strategy for mitigating this gap. Tobin et al. [5] randomized multiple properties of simulated scenes so that real images would appear as plausible variations of the training distribution. Tremblay et al. [6] also used randomized synthetic images and demonstrated that such randomization can facilitate transfer to real-world scenes. In manufacturing applications, Zhu et al. [7] investigated variations in material, background, lighting, camera settings and post-processing. Horváth et al. [8] also reported effective sim-to-real object detection using domain randomization and joint training on synthetic images and copies of a single real image. Lu et al. [9] investigated the context-aware placement of synthetic objects for augmenting construction-machinery datasets. Collectively, these studies indicate that synthetic data can support object detection, but its effectiveness depends strongly on how the synthetic images are generated.

2.2 Combining Real and Synthetic Training Data

Several studies have compared real-only, synthetic-only, and mixed real–synthetic training sets. Khan Mohammadi et al. [10] constructed real-only, synthetic-only, and mixed datasets for swimmer localization and evaluated them using YOLO. Arshad and Kumar [11] also reported that YOLOv8 helmet-detection performance varied substantially with the composition of the real and synthetic training data. Ljungqvist [4] concluded that adding synthetic data can improve performance; however, there is no consensus on an optimal real-to-synthetic ratio. Cecchetti et al. [12] investigated the balance between real and synthetic images in multiclass detection systems. Synthetic-only training can still be effective when the generated images adequately represent the variability encountered in real-world images. Niño and Gardi [13], for example, trained a YOLOv11 model exclusively on synthetic images generated using domain-randomization strategies, but noted that high validation performance on synthetic data does not necessarily translate into high performance on real images. Accordingly, all scenarios are evaluated using the same fixed real-image test set, while the real–synthetic composition of the training set is varied.

2.3 FCM-Based ANFIS Initialization

ANFIS integrates a fuzzy inference system with an adaptive network structure. A Sugeno-type fuzzy inference system is commonly used because the consequent of each rule can be expressed as a simple function of the input variables [14]. Karaboga and Kaya [15] reviewed different ANFIS training methods and emphasized that the choice

of membership functions and rule structure can substantially affect model performance. Fuzzy C-Means (FCM) [16] can be used before ANFIS to find soft clusters in the input data. A sample can belong partly to more than one cluster instead of being forced into only one group. Nemati et al. [17] applied FCM-initialized ANFIS models with particle swarm optimization (PSO) and genetic algorithms (GA) to a prediction problem in bidirectional deep drawing. Unlike the approach in [17], the present study does not apply PSO or GA. FCM is used to initialize the ANFIS membership functions and rule structure, after which the premise and consequent parameters are optimized using the detection-level training set. FCM–ANFIS was included as an exploratory post-classifier because it provides soft clustering and interpretable nonlinear rules for combining geometry, appearance, and confidence features extracted from YOLO detections. Adam was used to optimize the ANFIS premise parameters, while the consequent coefficients were estimated using ridge-regularized least squares.

3 Materials and Methods

3.1 Overall workflow

Figure 1 summarizes the overall experimental workflow. The real and synthetic image pools are first prepared and then combined according to the predefined real-data (R) and synthetic-data (S) training levels. A preliminary hyperparameter study is then performed to select a common training configuration, including the optimizer, initial learning rate, and batch size. The selected model architecture and training hyperparameters are kept fixed across the main YOLOv8s experiments, while the dataset composition and random seed are varied. Each valid dataset combination is trained twice using random seeds 0 and 1, and the same fixed real-image test set is used in all experiments. For the post-classification stage, detections produced by the selected YOLOv8s model are matched one-to-one with ground-truth objects and represented using seven numerical features. FCM is used to initialize ANFIS, and the final ANFIS or hybrid class decision is compared with the original YOLO class result. The detection-level dataset was generated using the YOLOv8s model from Scenario 2, which contained 20% real and 100% synthetic training data. This scenario was selected because it represented the lowest nonzero real-data level combined with the complete synthetic dataset and therefore provided a challenging data-limited setting for evaluating the post-classification stage. Seed 0 was used as the reference run for this exploratory analysis.

3.2 Screw Classes and Image Datasets

Two screw classes were used in the experiments: M3 $\times$ 8 and M3 $\times$ 12 (Figure 2). The real images were captured using a Raspberry Pi High Quality Camera and were divided into training, validation and test subsets. Synthetic images were created as a separate training pool using Autodesk Inventor. A total of 494 images were rendered using the ray-tracing tool, with both screw classes placed within the camera field of view. The images were annotated using Roboflow. To increase photometric variability and reduce the appearance difference between the synthetic and real images, brightness, grayscale,

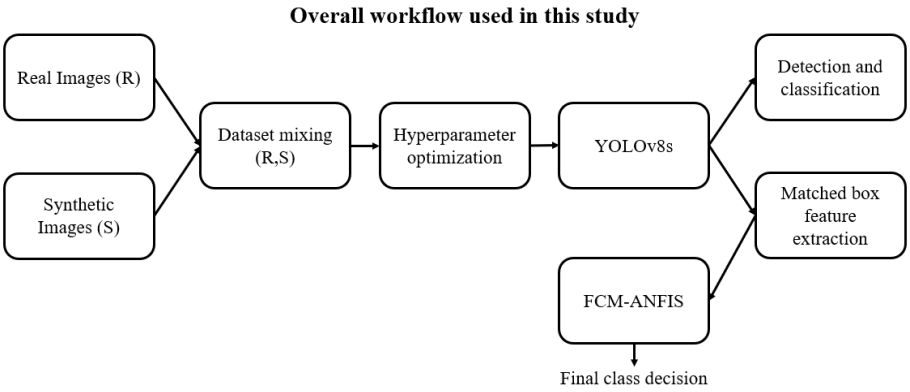


Fig. 1 Overall workflow used in this study. Hyperparameters are selected before the main YOLOv8 grid is trained; matched detections are then converted to features and evaluated with FCM-ANFIS.

and exposure augmentations were applied using Roboflow. After offline augmentation, the synthetic dataset contained 1,482 images. The complete real dataset consisted of 514 images with 538 annotated screw instances, including 312 M3 $\times$ 8 instances and 226 M3 $\times$ 12 instances. The dataset counts are summarized in Table 1.

Table 1 Number of images and annotated screw instances in the real and synthetic datasets.

Dataset	**Images**	**M3 $\times$ 8 instances**	**M3 $\times$ 12 instances**	**Total annotated objects**
Real	514	312	226	538
Synthetic (raw)	494	493	492	985
Synthetic (augmented)	1482	1482	1482	2964



Fig. 2 Synthetic representations of the two screw classes used in this study.

A fixed set of 154 real images, corresponding to approximately 30% of the complete real dataset, was used for testing in every experiment. When 256 real images, approximately 50% of the complete real dataset, were used for training, 104 real images were used for validation. When 104 real images, approximately 20% of the dataset, were used for training, 256 real images were used for validation. Using the same fixed real-image test set enables direct comparison across the evaluated training-data scenarios.

3.3 Real and Synthetic Experimental Design

The real-data factor R was defined as the percentage of the complete 514-image real dataset allocated to training, whereas the synthetic-data factor S was defined as the percentage of the 1,482-image augmented synthetic pool used for training:

$$R = \frac{N_{\text{real,train}}}{514} \times 100\ \%$$

$$S = \frac{N_{\text{synthetic,train}}}{1482} \times 100\ \%$$

The nominal experimental levels were ($R \in \{0\%, 20\%, 50\%\}$) and ($S \in \{0\%, 50\%, 100\%\}$). The combination $R = 0\%,\ S = 0\%$ was omitted because it contained no training images. The resulting experimental grid therefore contained eight valid data-composition scenarios. Each scenario was trained twice using random seeds 0 and 1, resulting in 16 YOLOv8s training runs.

Table 2 Experimental grid used for the real and synthetic data study. Percentages and corresponding image counts are reported for the real training, validation and test subsets and for the synthetic training subset.

Experiment ID	Scenario	Real train, R (image count)	Synthetic train, S (image count)	Real validation (image count)	Real test (image count)
E1	1	50% (256)	100% (1482)	20% (104)	30% (154)
E2	2	20% (104)	100% (1482)	50% (256)	30% (154)
E3	3	50% (256)	0% (0)	20% (104)	30% (154)
E4	4	20% (104)	0% (0)	50% (256)	30% (154)
E5	5	20% (104)	50% (741)	50% (256)	30% (154)
E6	6	0% (0)	100% (1482)	50% (256)	30% (154)
E7	7	50% (256)	50% (741)	20% (104)	30% (154)
E8	8	0% (0)	50% (741)	50% (256)	30% (154)

The measured points can show the main interaction between real and synthetic data in the investigated range, but they are not sufficient to claim a global optimum for every possible ratio.

3.4 Hyperparameter optimization

Before training the main real–synthetic experimental grid, a preliminary sequential hyperparameter-selection study was conducted (Table 3). The aim was to identify a single training configuration that would subsequently be applied unchanged to all eight dataset combinations. Hyperparameter selection was performed using synthetic images in the training and validation sets. First, Adam, AdamW, RAdam, and SGD were compared using an initial learning rate of 1×10^{-2} and a batch size of 64. SGD achieved the highest overall detection-classification score (D) of 8.82 % and was therefore selected. The overall detection-classification score (D) was calculated as

$$D = \frac{N_{correct}}{N_{correct} + N_{misclassified} + N_{false\ detections} + N_{missed}}$$

A screw assigned to the wrong screw class contributes once to $N_{misclassified}$; an unmatched prediction contributes to $N_{false\ detections}$; and an unmatched ground-truth screw contributes to N_{missed}. With SGD fixed, the initial learning rate was varied from 1×10^{-3} to 1×10^{-8} using a batch size of 64; an initial learning rate of 1×10^{-3} achieved the highest overall detection-classification score of 25.8%. Finally, batch sizes of 16, 32, and 64 were compared using SGD and an initial learning rate of 1×10^{-3}; a batch size of 32 was selected. The final common configuration used in the main experiments therefore consisted of SGD, an initial learning rate of 1×10^{-3}, and a batch size of 32. Because each selection stage used different fixed settings, the validation results reported in Table 3 are not directly comparable across rows or with the final test results reported in Section 4.

Table 3 Summary of the hyperparameter study used to select the final YOLOv8 training setup.

Selection stage	**Values tested**	**Selected value**	**Overall detection-classification score (D) (%)**
Optimizer	Adam, AdamW, RAdam, SGD	SGD	8.82
Initial learning rate	1×10^{-3} to 1×10^{-8}	1×10^{-3}	25.8
Batch size	16, 32, 64	32	27.9

3.5 YOLOv8 training configuration

YOLOv8s was used as the object-detection model. The model was initialized with COCO-pretrained weights from the Ultralytics yolov8s.pt checkpoint. The input images were resized to 416 × 416 pixels, and each model was trained for a maximum of 300 epochs with an early-stopping patience of 25 epochs, terminating training when the validation mAP 0.5:0.95 failed to improve for 25 consecutive epochs. An input size of 416 × 416 pixels was used. The selected training configuration used SGD with an initial learning rate of 1×10^{-3}, cosine learning-rate scheduling, and a batch size of 32. The model architecture and all training hyperparameters were kept fixed across

the eight dataset combinations. Only the composition of the training dataset, the size of the real validation subset, and the random seed differed among the experimental runs. Each dataset combination was trained twice, using random seeds 0 and 1.

Table 4 Main YOLOv8 training settings used for the real-synthetic experiments.

Parameter	Setting
Model	YOLOv8s
Number of classes	2
Input image size	416 × 416 pixels
Maximum epochs	300
Early stopping patience	25
Optimizer	SGD
Initial learning rate	1×10^{-3}
Learning-rate schedule	Cosine
Batch size	32
Training combinations	8
Seeds	Random seeds: 0 and 1
Total YOLO runs	16
Fixed real test set	154 images (30% of the real dataset)

During training, Ultralytics reports box loss, classification loss and distribution focal loss together with precision, recall, mAP50 and mAP50-95. For the real-synthetic comparison in this paper, the common value available for all eight scenarios was the real test detection-classification score, so this value was used as the response surface height.

3.6 Detection-level dataset for FCM–ANFIS

The ANFIS stage was trained using numerical features extracted from YOLO detections rather than directly from raw images. The detection-level dataset used for the FCM–ANFIS analysis was generated using the YOLOv8s model from Scenario 2 trained with random seed 0. This model was trained using 104 real images, corresponding to 20% of the complete real dataset, and all 1,482 augmented synthetic images. YOLO was then run on the training, validation and test splits to obtain the detections used for the ANFIS dataset. Scenario 2 was selected because it contained the lowest nonzero level of real training data investigated in this study, allowing the FCM–ANFIS refinement to be evaluated under a challenging data-limited condition. Predicted boxes were matched to ground truth objects using Intersection over Union (IoU). Greedy one to one matching was performed at IoU ≥ 0.50. Class labels were not used during IoU-based matching. Class labels were not used during IoU-based matching. Matching was based only on the bounding-box IoU, and matched detections with incorrect YOLO class predictions were retained. This allowed the post-classification stage to evaluate whether ANFIS could correct YOLO class errors for detections that were already localized correctly. A minimum YOLO confidence threshold of 0.001 was used during dataset construction to retain low-confidence detections that could

potentially benefit from post-classification. The image size was 416 × 416 pixels. Five images with mixed annotation row formats were excluded during construction of the detection level dataset.

After matching, 3,481 detection level samples were available: 3,054 for training, 274 for validation and 153 for testing. The ground truth screw class was used as the target for each matched detection. Overall accuracy was calculated from the matched detections only. Accordingly, this metric evaluates screw-type classification conditional on successful object matching and should not be interpreted as an end-to-end object-detection metric such as mAP. For each evaluation split, matched-detection classification accuracy was calculated as $(N_{\text{correct}}/N_{\text{matched}})$, where (N_{correct})denotes the number of matched detections assigned the correct screw class and (N_{matched}) denotes the total number of matched detections.

The two classes were close to balance in all splits, as shown in Table 5.

Table 5 Class distribution of the detection-level dataset used for FCM-ANFIS.

Split	**Class 0 (M3 × 8)**	**Class 1 (M3 × 12)**	**Total**
Train	1559	1495	3054
Validation	146	128	274
Test	78	75	153

Seven numerical features were extracted from each matched detection, as summarized in Table 6. The YOLO predicted class was deliberately not used as an ANFIS input. This design was intended to prevent the fuzzy model from merely reproducing YOLO's original class prediction.

Table 6 Features used as inputs to the FCM-ANFIS model.

Feature	**Meaning**
long_side_norm	Length of the longer bounding-box side divided by the image diagonal
short_side_norm	Length of the shorter bounding-box side divided by the image diagonal
aspect_ratio	Ratio of the longer bounding-box side to its shorter side
box_area_norm	Bounding-box area divided by the total image area
edge_density	Proportion of edge pixels within the detected crop after Gaussian smoothing and Canny edge detection
gray_std	Standard deviation of grayscale intensities within the detected crop, normalized by 255
yolo_conf	YOLO confidence score associated with the matched detection

Before edge extraction, each grayscale detection crop was smoothed using a 3 × 3 Gaussian kernel. The Gaussian standard deviation was determined automatically by OpenCV (sigmaX = 0). Canny edge detection was then applied using lower and

upper thresholds of 50 and 150, respectively. Before FCM–ANFIS training, all seven input features were standardized to zero mean and unit variance using the mean and standard deviation estimated exclusively from the training subset. The same training-derived transformation was applied unchanged to the validation and test subsets.

3.7 Fuzzy C-Means initialization

FCM was used to find fuzzy groups in the seven-dimensional feature space. FCM clustering was fitted exclusively to the 3,054 training samples. The validation subset was used for model selection and decision-threshold tuning, while the test subset was reserved for final evaluation. For N samples and C clusters, the usual FCM objective can be written as

$$J_m = \sum_{i=1}^{N} \sum_{k=1}^{C} u_{ik}^m \left\| x_i - c_k \right\|^2$$

where u_{ik} is the membership of sample i in cluster k, (x_i) is the seven-dimensional feature vector of sample (i), (c_k) is the center of cluster (k) and m controls the amount of fuzziness. Before FCM clustering, each input feature was standardized using the mean and standard deviation estimated exclusively from the training subset. The training-derived scaling parameters were subsequently applied unchanged to the validation and test subsets. FCM was configured with Euclidean distance, a fuzziness exponent of m = 2.0, a random seed of 42, a maximum of 1,000 iterations, and a convergence tolerance of 1×10^{-5} based on the Frobenius norm of the change in the membership matrix between successive iterations. Candidate cluster counts C $\in$ {2, 3, 4, 5, 6} were evaluated. The solutions for C = 2, 3, 4, 5, and 6 converged after 68, 56, 89, 47, and 94 iterations, respectively.

Each cluster was mapped to one initial fuzzy rule. The cluster centers were used to initialize the centers of the Gaussian membership functions, while feature-specific fuzzy weighted standard deviations were used to initialize their spreads. For cluster (k) and feature (j), the initial spread was calculated as

$$\sigma_{kj} = \sqrt{\frac{\sum_{i=1}^{N} u_{ik}^m \left(x_{ij} - c_{kj}\right)^2}{\sum_{i=1}^{N} u_{ik}^m}}.$$

Here, x_{ij} and c_{kj} denote the j^{th} components of sample (i) and cluster center (k), respectively. The fuzzy partition coefficient was calculated as

$$FPC = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{C} u_{ik}^2.$$

For a fixed dataset and a fixed number of clusters, a higher FPC indicates a crisper partition. However, because its lower bound depends on C, FPC was treated only as a descriptive clustering diagnostic. The final number of clusters and therefore the initial number of fuzzy rules were selected exclusively from the validation results. Validation

accuracy was used as the primary selection criterion, with F1-score used as the tie-breaker. The three-rule configuration (C = 3) achieved the highest validation accuracy and was selected for the final comparison.

3.8 ANFIS training and hybrid decision

A first-order Sugeno ANFIS model was trained using the FCM initialization. Gaussian premise functions were used. The consequent coefficients were estimated using ridge-regularized least squares with a regularization coefficient of 1×10^{-3}. The Gaussian centers and widths were updated using Adam with a learning rate of 1×10^{-2}. Mean squared error was used as the loss. Rule counts 2, 3, 4, 5, and 6 were evaluated for up to 200 epochs using early stopping based on validation MSE, with a patience of 35 epochs.

Two ANFIS evaluation modes were considered. In the pure mode, the ANFIS output alone was used to determine the class. In the hybrid mode, YOLO remained the main classifier and ANFIS overrode the YOLO class only when its output fell outside a validation-selected margin around the decision threshold. The threshold and override margin were selected using validation accuracy with F1 score as the tie break. The three-rule model achieved the highest validation accuracy and was selected for the final comparison.

Table 7 FCM–ANFIS model settings selected from the training and validation procedure.

Number of Rules	FPC	Stopped epoch	ANFIS threshold	Override margin
2	0.626658	41	0.430	0.170
3	0.594166	39	0.380	0.380
4	0.563537	97	0.805	0.025
5	0.546807	104	0.750	0.255
6	0.483345	116	0.555	0.000

3.9 Evaluation measures

The YOLO validation performance was monitored using precision, recall, mAP50 and mAP50-95. For the real–synthetic comparison, class-wise recall and an overall detection-classification score were calculated from the full YOLO test-set confusion matrix. Because true negatives are not meaningfully defined for object detection, conventional classification accuracy was not used for this full-detection comparison. Class-wise recall (R_c) was calculated as

$$R_c = \frac{TP_c}{TP_c + FN_c}$$

Here TP_c and FN_c denote the true-positive and false-negative counts, respectively, for the corresponding screw-class ‘c’. The overall detection-classification score (D) was

also used as an evaluating parameter. The same evaluation procedure was applied to all eight dataset combinations, and the two-seed mean score was used as the response-surface height. The ANFIS stage was evaluated with accuracy, precision, recall and F1-score on the matched test detections.

The following ANFIS classification metrics were calculated exclusively from the matched detections and should not be confused with the object-detection metrics reported for YOLO.

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP})$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN})$$

$$F1 - score = 2 \times \text{Precision} \times \text{Recall}/\left(\text{Precision} + \text{Recall}\right)$$

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

Here, TP, TN, FP, and FN denote true-positive, true-negative, false-positive, and false-negative screw-class predictions, respectively, among the matched detections. The response surface was created by piecewise-linear interpolation of the eight measured mean detection-classification scores. It was used only as a visual description of the experimental points. It should not be interpreted as an observed detection-classification score at every point on the surface.

4 Results

4.1 Real-synthetic training results

Table 8 summarizes the results from seeds 0 and 1. The mean of the two runs was used for the comparison plot (Figure 3) and response surface (Figure 4). The detection-classification score is reported as the mean ± sample standard deviation across the two seeds. The mean class-wise recall values are reported as produced by the evaluation pipeline whereas the overall detection-classification score is a separate test measure. Because only two seeds were used, the standard deviation is treated as an indicator of run-to-run variability.

4.2 Effect of real training data

The clearest observed pattern was the increase in mean detection-classification score with increasing amounts of real training data. With 100% synthetic data, the mean detection-classification score was 19.28% with no real training data. It increased to 53.65% with 20% real data and to 83.74% with 50% real data.

The same pattern was observed at 50% synthetic data. The mean detection-classification score was 16.38% with no real training data, 56.98% with 20% real data and 82.46% with 50% real data. In this dataset, real images were associated with substantially higher mean detection-classification scores on the real-image test set.

Table 8 YOLO test results for the eight real-synthetic data combinations. The detection-classification score is reported as mean ± sample standard deviation of the runs with seeds 0 and 1.

Experiment ID	R (%)	S (%)	Mean M3×8 recall (%)	Mean M3×12 recall (%)	Detection classification score, seed 0 (%)	Detection classification score, seed 1 (%)	Mean score ± SD (%)
E1	50	100	96.15	87.79	86.90	80.57	83.74 ± 4.48
E2	20	100	69.80	78.18	58.90	48.40	53.65 ± 7.42
E3	50	0	99.35	75.63	80.40	81.90	81.15 ± 1.06
E4	20	0	81.35	70.48	61.50	64.60	63.05 ± 2.19
E5	20	50	73.70	77.54	61.60	52.36	56.98 ± 6.53
E6	0	100	46.10	66.00	11.90	26.65	19.28 ± 10.43
E7	50	50	92.93	88.40	81.30	83.62	82.46 ± 1.64
E8	0	50	45.50	38.46	10.60	22.16	16.38 ± 8.17

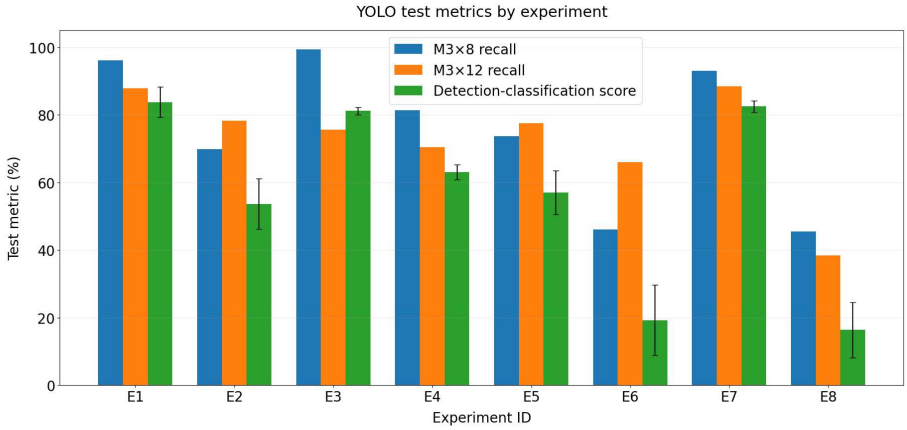


Fig. 3 Mean class-wise recall and overall detection-classification score for seeds 0 and 1. Error bars on the overall detection-classification score show ±1 sample standard deviation.

4.3 Effect of synthetic training data

The observed relationship between synthetic-data quantity and mean detection-classification score varied across real data levels. With 50% real data, the mean detection-classification score was 81.15% with no synthetic data, 82.46% with 50% synthetic data and 83.74% with 100% synthetic data. Synthetic data therefore was associated with a small increase in mean detection-classification score when 50% real training data was available.

At 20% real data, the trend was different. The mean detection-classification score was 63.05% with no synthetic data, 56.98% with 50% synthetic data and 53.65% with 100% synthetic data. In this part of the grid, adding more synthetic data was associated with a decrease in mean detection-classification score.

With no real training data, increasing the synthetic-data level from 50% to 100% increased the mean detection-classification score only slightly, from 16.38% to 19.28%. Both synthetic-only configurations nevertheless remained substantially below the configurations containing real training data and showed high run-to-run variability.

4.4 Coarse response surface

Figure 4 shows the two-seed mean detection-classification score using the percentage of the complete real dataset used for training (R) on the x-axis, the percentage of the available synthetic training pool used (S) on the y-axis and mean detection-classification score on the z-axis. The highest measured mean in the current grid was obtained at $R = 50\%$ and $S = 100\%$, reaching a mean detection-classification score of 83.74%. This was the highest measured value within the evaluated grid and should not be interpreted as a global optimum. These results suggest that the real-to-synthetic ratio should be evaluated for the actual camera and object setup instead of being transferred directly from another application. The response surface provides a visual summary of the eight measured combinations. Table 8 and Figure 3 also show that some configurations with little or no real training data exhibited substantial run-to-run variability.



Fig. 4 Piecewise-linear interpolated response surface of the mean detection-classification score from seeds 0 and 1. Only the eight marked points were measured; all intermediate surface values were obtained by linear interpolation.

4.5 FCM-ANFIS results

The FPC decreased from 0.6267 with two rules to 0.4833 with six rules (Figure 5). This trend was interpreted descriptively because raw FPC values depend on the number of clusters and are not directly comparable across different rule counts. More rules also did not lead to a monotonic increase in classification performance (Table 9 and Figure 6).

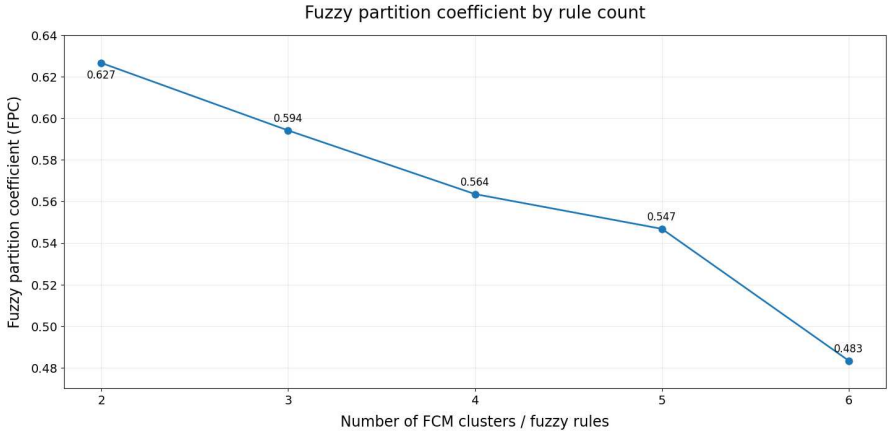


Fig. 5 Fuzzy Partition Coefficient for the five evaluated FCM cluster counts used to initialize the corresponding ANFIS rule configurations.

Table 9 Test classification results for YOLO, Random Forest, SVM and hybrid YOLO + FCM–ANFIS models on 153 matched detections.

Method	**FPC**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
YOLO baseline	N/A	80.39	81.69	77.33	79.45
Random Forest	N/A	79.08	73.63	89.33	80.72
SVM	N/A	77.12	76.32	77.33	76.82
2-rule hybrid	0.6267	76.47	74.07	80.00	76.92
3-rule hybrid	0.5942	82.35	86.36	76.00	80.85
4-rule hybrid	0.5635	66.67	67.14	62.67	64.83
5-rule hybrid	0.5468	80.39	80.82	78.67	79.73
6-rule hybrid	0.4833	77.12	74.39	81.33	77.71

The best hybrid result was obtained with three rules. Accuracy increased from 80.39% for the YOLO baseline to 82.35%, and F1 score increased from 79.45**% to 80.85%**. A McNemar test was performed on the paired test predictions of the YOLO baseline and the three-rule hybrid model. The exact two-sided test gave a p-value of 0.629. Although the hybrid model achieved higher accuracy, the difference was not

statistically significant at the 5% level. The observed improvement should therefore be interpreted as a small, non-significant difference rather than evidence that the hybrid model was superior. Precision increased from 81.69% to 86.36%, while recall

decreased from 77.33% to 76.00% (Figure 6). The three-rule pure ANFIS model also reached 82.35% accuracy. Random Forest reached 79.08% test accuracy with an F1 score of 80.72%, while SVM reached 77.12% with an F1 score of 76.82%. Two-, four-, five-, and six-rule hybrid models did not improve accuracy over the YOLO baseline.

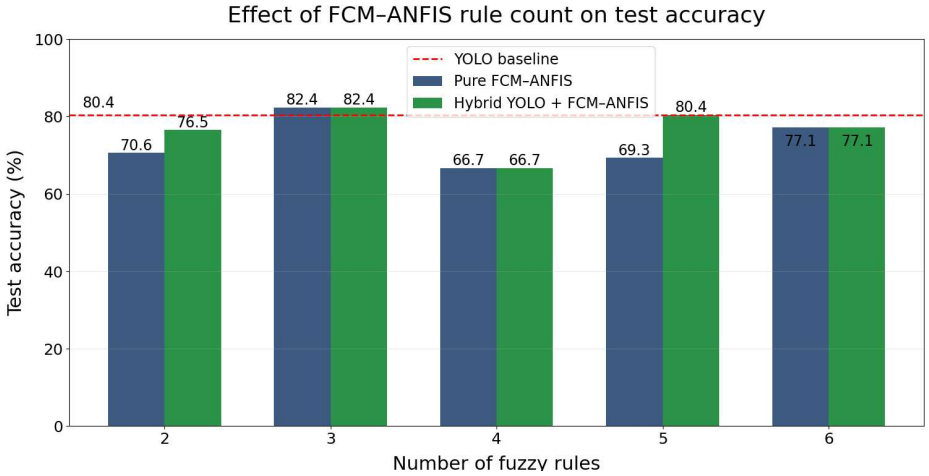


Fig. 6 Test accuracy of pure FCM-ANFIS and hybrid YOLO + FCM-ANFIS across the five evaluated rule counts. The dashed line is the YOLO baseline.

The test confusion matrix for the YOLO baseline was $\begin{bmatrix} 65 & 13 \\ 17 & 58 \end{bmatrix}$. Rows represent the ground-truth classes and columns represent the predicted classes, with Class 1 treated as the positive class. With the three-rule hybrid model, it became $\begin{bmatrix} 69 & 9 \\ 18 & 57 \end{bmatrix}$. The main change was a reduction in false positives from 13 to 9, with one additional false negative. This agrees with the increase in precision and the small reduction in recall.

5 Discussion

5.1 Real data remained necessary for this screw dataset

Real training images were the factor most strongly associated with mean detection-classification score in the evaluated grid. With 100% synthetic data, the mean detection-classification score increased from 19.28% at 0% real data to 53.65% at 20% real data and 83.74% at 50% real data. This is consistent with the review by Ljungqvist [4], which similarly indicated that real images were important when the final test domain was real.

The low synthetic-only result does not mean that synthetic data is always weak. Zhu et al. [7], Horváth et al. [8] and Niño and Gardi [13] showed that synthetic-to-real transfer can work well when the generated scenes cover relevant real-world variability. The current results suggest that a substantial synthetic-to-real domain gap remained for the screw images. The use of only two random seeds should be regarded as a limitation of the present study. Although the strong advantage of including real training data was observed in both runs, additional seeds would provide a more robust estimate of training variability and improve statistical confidence in small performance differences.

5.2 More synthetic data was not always better

The observed influence of synthetic data depended on the amount of real data. At 50% real data, the mean detection-classification score increased from 81.15% to 83.74% when the synthetic level was increased from 0% to 100%. However, this 2.59-percentage-point increase was small and was not observed consistently across both seeds. At 20% real data, the mean detection-classification score decreased from 63.05% to 53.65%. This is consistent with the conclusion in earlier ratio studies that there is no single mixing rule that works for every dataset [4, 11, 12].

5.3 ANFIS worked best with a small rule base

The validation-selected three-rule ANFIS configuration achieved the highest test accuracy among the evaluated ANFIS configurations. The hybrid accuracy increased from 80.39% for the YOLO baseline to 82.35%, and the F1-score increased from 79.45% to 80.85%. The three-rule pure ANFIS model achieved the same test accuracy of 82.35%; therefore, the hybrid model did not outperform the pure ANFIS model in terms of accuracy. Random Forest and SVM, trained on the same seven features, reached 79.08% and 77.12% test accuracy, respectively. Although Random Forest had lower accuracy than the three-rule hybrid model, its F1-score was similar (80.72% versus 80.85%). The two-, four-, five-, and six-rule hybrid models did not improve accuracy over the YOLO baseline.

The FPC decreased as the number of rules increased, but raw FPC values depend on the number of clusters and were therefore interpreted only descriptively. FPC and classification performance were therefore considered separately, and the final rule count was selected using validation performance rather than FPC alone.

The ANFIS gain is small and should be viewed as a refinement of YOLO, not a replacement. The ANFIS test used only detections that were matched to ground-truth objects at IoU ≥ 0.50, so these classification results should not be compared directly with full object-detection mAP. The McNemar test indicated that the observed improvement was not statistically significant, further supporting a cautious interpretation of the reported gain. The absolute gain was 1.96 percentage points. The goal of the ANFIS analysis was exploratory rather than to establish a definitive state-of-the-art classifier. The FCM–ANFIS analysis was conducted using detections from only one YOLO training configuration (Scenario 2, seed 0). Consequently, the observed gain may not generalize to other real–synthetic data combinations or random seeds. In

addition, the ANFIS training features were extracted from YOLO predictions on the YOLO training images. Future work should use out-of-fold predictions or an independent calibration subset to reduce the risk of optimistic second-stage training. Future work should evaluate the method across multiple real–synthetic training configurations and random seeds, using larger test sets that include additional screw categories and imaging conditions.

6 Conclusion

This work studied screw detection using real and synthetic training images and then tested an FCM–ANFIS stage for class refinement. Real-data levels of 0%, 20% and 50% and synthetic-data levels of 0%, 50% and 100% were used. The ($R = 0\%,\ S = 0\%$) combination case was omitted, leaving eight combinations. Each combination was trained with seeds 0 and 1, giving 16 YOLOv8s runs with the same fixed 30% real test set. A preliminary hyperparameter-selection study identified SGD, a learning rate of (1×10^{-3}) and a batch size of 32 before the main grid was trained.

The highest observed two-seed mean detection-classification score was 83.74% for 50% real and 100% synthetic training data. At 50% real data, the mean detection-classification score increased slightly as the amount of synthetic data increased. At 20% real data, adding more synthetic images reduced the mean detection-classification score. The scores from synthetic-only training remained substantially lower than those from training configurations that included real images. The main conclusion is that synthetic data may improve performance, but its effect depends on how much real data is already available and the remaining synthetic-to-real domain gap.

For the second stage, seven geometry, appearance and confidence features were extracted from matched YOLO detections. FCM was used to initialize ANFIS models with 2, 3, 4, 5 and 6 rules. The validation-selected three-rule hybrid achieved the highest test accuracy among the evaluated hybrid configurations, increasing matched-detection classification accuracy from 80.39% to 82.35% and F1-score from 79.45% to 80.85% on the matched test detections. However, the 1.96-percentage-point accuracy increase was not statistically significant according to the exact McNemar test (p=0.629). The three-rule pure ANFIS model achieved the same test accuracy of 82.35%, so the results did not demonstrate an accuracy advantage of the hybrid decision rule over pure ANFIS. Random Forest and SVM baselines achieved lower test accuracies of 79.08% and 77.12%, respectively. This exploratory second-stage analysis was limited to 153 matched detections obtained from one YOLO configuration (Scenario 2, seed 0) and should not be interpreted as evidence of improved end-to-end object-detection performance.

Future work should investigate reducing the synthetic-to-real gap and evaluating additional grid points around the highest-performing measured region. Additional screw geometries, head types and thread dimensions should be evaluated in future work to evaluate the generalizability of the proposed approach. The evaluation should also include real images captured under a wider range of lighting and background conditions. Additional random seeds should be evaluated to provide a more reliable estimate of training variability. The FCM–ANFIS stage should also be tested across

multiple real–synthetic training configurations using out-of-fold YOLO predictions or an independent calibration dataset.

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection, " in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, 2016.

[2] Jocher, G., Chaurasia, A. and Qiu, J. (2023) Ultralytics YOLOv8, software, version 8.0.0. https://github.com/ultralytics/ultralytics.

[3] F. E. Nowruzi, P. Kapoor, D. Kolhatkar, F. Al Hassanat, R. Laganiere, and J. Rebut, "How much real data do we actually need: Analyzing object detection performance using synthetic and real data, " arXiv preprint arXiv:1907.07061, 2019, doi: 10.48550/arXiv.1907.07061.

[4] M. G. Ljungqvist, "On synthetic and real images as training data for object detection — A brief review," in From Electrophoresis to Wikidata: Festschrift in honor of Pierre Nugues, University of Tartu Library, 2025, pp. 16-19.

[5] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 23–30, 2017, doi: 10.1109/IROS.2017.8202133.

[6] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 969–977, 2018.

[7] X. Zhu, J. Henningsson, D. Li, P. Mårtensson, L. Hanson, M. Björkman, and A. Maki, "Domain Randomization for Object Detection in Manufacturing Applications using Synthetic Data: A Comprehensive Study," in 2025 IEEE International Conference on Robotics and Automation (ICRA), pp. 16715–16721, 2025, doi: 10.1109/ICRA55743.2025.11128647.

[8] D. Horváth, G. Erdős, Z. Istenes, T. Horváth, and S. Földi, "Object Detection Using Sim2Real Domain Randomization for Robotic Applications," IEEE Transactions on Robotics, vol. 39, no. 2, pp. 1225–1243, Apr. 2023, doi: 10.1109/TRO.2022.3207619

[9] Y. Lu, B. Liu, W. Wei, B. Xiao, Z. Liu, and W. Li, "Generating synthetic images for construction machinery data augmentation utilizing context-aware object placement," Developments in the Built Environment, vol. 21, art. 100610, 2025, doi: 10.1016/j.dibe.2025.100610.

[10] M. Khan Mohammadi, T. Schneidereit, A. Mansouri Yarahmadi, and M. Breuß, "Investigating Training Datasets of Real and Synthetic Images for Outdoor Swimmer Localisation with YOLO," AI, vol. 5, no. 2, pp. 576–593, 2024. doi:10.3390/ai5020030.

[11] M. Arshad and P. Kumar, "Impact of Synthetic Data on Training and Improved YOLOv8 Models for Helmet Detection," Journal of Machine and Computing, vol. 5, no. 3, pp. 1439–1449, July 2025, doi:10.53759/7669/jmc202505114.

[12] V. B. Cecchetti, M. Rudek, and R. Z. Freire, "In the Search for the Balance Between Real and Synthetic Images in Multi-Class Detection Systems," 2024 IEEE 19th Conference on Industrial Electronics and Applications (ICIEA), pp. 1–6, 2024. doi:10.1109/ICIEA61579.2024.10664977.

[13] L. Torquato Niño and H. A. A. Gardi, "Synthetic-to-Real Object Detection using YOLOv11 and Domain Randomization Strategies," arXiv preprint arXiv:2509.15045, 2025, doi: 10.48550/arXiv.2509.15045.

[14] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-15, no. 1, pp. 116–132, 1985. doi:10.1109/TSMC.1985.6313399.

[15] D. Karaboga and E. Kaya, "Adaptive network based fuzzy inference system (ANFIS) training approaches: a comprehensive survey," Artificial Intelligence Review, vol. 52, no. 4, pp. 2263–2293, 2019. doi:10.1007/s10462-017-9610-2.

[16] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms. New York, NY, USA: Plenum Press, 1981, doi: 10.1007/978-1-4757-0450-1.

[17] A. Nemati, D. Bäcker, P. Muller, and W.-G. Drossel, "Adaptive neuro-fuzzy inference system models based on particle swarm optimization and genetic algorithms for bidirectional deep drawing," in Proceedings of the 6th International Conference on Engineering Technology and Innovation (ICETI), pp. 24–31, 2022.