

DOI: 10.5281/zenodo.23569109

CURRENT DEVELOPMENTS AND THEMATIC REVIEWS IN COMPUTER SCIENCE AND ENGINEERING

Editor

Prof. Selmin Ener Ruşen, Ph.D.

Current Developments and Thematic Reviews in Computer Science and Engineering

Editor

Prof. Selmin Ener Ruşen, Ph.D.

Publisher
Platanus Publishing®

Editor in Chief
Prof. Selmin Ener Ruşen, PhD.

Cover & Interior Design
Platanus Publishing®

The First Edition
October, 2026

ISBN
978-625-6634-93-0

©copyright
All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, or any information storage or retrieval system, without permission from the publisher.

Platanus Publishing®
Address: Natoyolu Cad. Fahri Korutürk Mah. 157/B, 06480, Mamak, Ankara, Turkey.

Phone: +90 312 390 1 118
web: www.platanuspublishing.com
e-mail: platanuskita@gmail.com



Platanus Publishing®

CONTENTS

CHAPTER 1 5

The Fundamentals of Colorectal Polyp Segmentation and an Experimental Study on YOLOv8 and YOLO11

Semih Özenç
Yavuz Selim Şahin
Süleyman Uzun

CHAPTER 2 33

A General Framework for Machine Learning Model Selection Considering Data Modalities, Computational Constraints and Interpretability

Hilal Çelik

CHAPTER 1

The Fundamentals of Colorectal Polyp Segmentation and an Experimental Study on YOLOv8 and YOLO11

Semih Özenç¹
Yavuz Selim Şahin²
Süleyman Uzun³

¹ Research Assistant, Sakarya University of Applied Sciences, Faculty of Technology, Department of Computer Engineering, Sakarya, Türkiye, ORCID: 0009-0001-5393-8495

² Sakarya University of Applied Sciences, Graduate Education Institute, Department of Computer Engineering, Sakarya, Türkiye, ORCID: 0009-0000-8283-8971

³ Assoc. Prof. Dr., Sakarya University of Applied Sciences, Faculty of Technology, Department of Computer Engineering, Sakarya, Türkiye, ORCID: 0000-0001-8246-6733

1 Introduction

Colorectal cancer is one of the major health issues where early diagnosis and preventive measures are crucial. The World Health Organization’s 2026 fact sheet reports an estimated 1.9 million new colorectal cancer cases and more than 900,000 deaths worldwide in 2022. These figures demonstrate that colorectal cancer continues to hold a significant position within the global cancer burden. The Organization emphasizes that the detection and removal of precancerous lesions through appropriate screening programs can reduce the disease burden (World Health Organization, 2026). Determining the location and visible boundaries of polyps in colonoscopic images is one of the tasks being investigated through computer-aided image analysis in this clinical context. From an image analysis perspective, polyp segmentation involves determining whether each pixel belongs to the polyp region. The appearance of polyps varies depending on size, color, texture, shape, and lighting. Distinguishing borders can be challenging in lesions that share similar colors with the mucosa or that protrude only slightly from the surface. Reflections, residual fluid, blurriness, and partial visibility also influence the algorithm’s decision. This variability means that a method that performs well on a single image does not guarantee similar success across different patients, devices, and imaging conditions. The segmentation problem must therefore be addressed as both a representation learning problem and a reliable assessment problem.

Deep learning enables image features to be learned from examples rather than being manually designed. Encoder-decoder networks, such as U-Net, have established a fundamental approach for biomedical segmentation by combining contextual information with spatial detail (Ronneberger, Fischer, & Brox, 2015). Subsequently, attention mechanisms, multiscale feature fusion, and transformer architectures have been developed for polyp segmentation. On the other hand, the YOLO family enables the extension of a single-stage approach—originally developed for object detection—to the segmentation task. These two research lines have different design priorities. Therefore, comparisons should not be based solely on the name of the architecture. Model selection should be considered in conjunction with data volume, label representation, training budget, and target hardware. Additional parameters may not translate into a useful representation within a fixed training time. A lower number of operations also does not always result in a shorter inference time due to the way the hardware executes different processing blocks. Therefore, performance and cost should be evaluated within the same experimental context.

The aim of this study is to explain the fundamental concepts of polyp segmentation and to experimentally investigate model selection under common training conditions. First, task definitions, evaluation metrics, and the main architectural approaches are presented. Next, five scale variants each from the

YOLOv8 and YOLO11 families are compared. The experimental investigation examines the effect of model scale on validation performance, the pixel and boundary performance of the weights selected via validation on the separate test set, and the contribution of computational cost to model selection. Thus, the relationship between instance segmentation and semantic segmentation metrics is demonstrated through a concrete evaluation protocol.

2 The Theoretical Foundations of Polyp Segmentation

2.1 Object Detection, Semantic Segmentation, and Instance Segmentation

Object detection typically indicates the location of a target in an image using a bounding box and a confidence score. The box may also include healthy tissue surrounding the lesion. In semantic segmentation, however, each pixel is assigned a class, and adjacent regions within the same class may not have distinct object identifiers. Instance segmentation represents each target as a separate object in addition to the pixel mask. When multiple polyps are present in a colonoscopy image, this distinction becomes important for counting and evaluating the lesions individually. YOLO segmentation models produce a bounding box, class, confidence score, and object mask. In the experiments discussed in this section, the target class is “polyp.” In the training and validation logs, bounding box metrics are denoted with the suffix “B,” while mask metrics are denoted with the suffix “M.” For example, bounding box mAP50–95 evaluates the positioning of the lesion within a rectangle, whereas mask mAP50–95 is based on the overlap between the predicted region and the reference object mask. Although both metrics are calculated from the same images, they are sensitive to different types of errors. The correct placement of the bounding box does not, by itself, prove that fine indentations and protrusions at the lesion edges are accurately represented in the mask. The unit of evaluation is also part of the task. Obtaining an accurate mask in a single image does not indicate that the lesion was detected the moment it first appeared in the video or that few false alarms were generated along the normal mucosa. Performance at the image, lesion, and procedure levels answers different questions.

2.2 Effects of Image and Label Properties

The level of detail a segmentation model can learn depends not only on the quality of the input image but also on the representation of the target mask. In regions where the reference boundary may be interpreted differently by experts, it becomes difficult to distinguish between model error and label ambiguity. A boundary shift of just a few pixels in a small polyp can alter the overlap ratio more significantly than in a large polyp. This is one of the reasons why a single average score cannot adequately account for lesions of different sizes. Analysis by size and appearance subgroups can reveal examples that perform below the average. Converting masks into a format usable for model training is also a distinct methodological step. When generating polygonal labels from a pixel

mask, the selection of contour points, the preservation of small components, and the handling of internal voids can all affect the result. When the image is rescaled, the labels must undergo the same geometric transformation. Therefore, the data preparation process should not be viewed merely as a file format conversion. Verifying the overlap of the transformed label with the original mask helps identify any systematic errors that may arise before model training. Visual similarity and source relationships should also be considered when dividing examples into training, validation, and test sets. While ensuring that the same file name does not appear in two sets is a necessary check, it does not preclude similar frames from the same patient or the same video sequence from being distributed across different sets. Such a split may overestimate model performance by including evaluation images that closely resemble training images. Fitzgerald and Matuszewski (2023) discuss the importance of video-induced data leakage in polyp segmentation from an evaluation perspective. When patient or video identifiers are available, splitting the data by these groups provides stronger independence between subsets.

2.3 Evaluation Metrics

To evaluate the accuracy of the segmentation outputs generated by the model, it is necessary to quantitatively measure the overlap between the predicted masks and the ground-truth masks. For this purpose, Intersection over Union (IoU) and the Dice similarity coefficient are commonly used in segmentation studies. IoU is defined as the ratio of the intersection area between the predicted mask (A) and the ground-truth mask (G) to the union area of these two masks, and is calculated as shown in Equation (1).

$$IoU = \frac{|A \cap G|}{|A \cup G|} \quad (1)$$

Here, $|A \cap G|$ represents the area covered in common by the predicted and ground-truth masks, while $|A \cup G|$ represents the total coverage area of the two masks. The IoU value ranges from 0 to 1; as the value approaches 1, it indicates an increase in the spatial overlap between the predicted mask and the ground-truth mask.

The Dice similarity coefficient, on the other hand, evaluates the same overlap using a different normalization approach and is defined as the ratio of twice the intersection area of the predicted and reference masks to the total area of the masks. The Dice coefficient is calculated using Equation (2):

$$Dice = \frac{2|A \cap G|}{|A| + |G|} \quad (2)$$

The Dice coefficient also takes values between 0 and 1, and higher values indicate a greater degree of overlap between the predicted region and the

reference region. There is a mathematical relationship between the Dice and IoU metrics for the same prediction–reference mask pair. This relationship is given in Equation (3).

$$Dice = \frac{2IoU}{1 + IoU} \quad (3)$$

Although Equation (3) holds for a single prediction–reference mask pair, directly applying the same transformation to the Dice and IoU values—which are calculated separately across the dataset and then averaged—is generally not valid. The main reason for this is that the transformation in Equation (3) is nonlinear, and the averages of the scores calculated on a per-example basis are not preserved under this nonlinear transformation. Therefore, the average Dice and average IoU values at the dataset level must be calculated independently of one another.

Precision is the ratio of true positive predictions to all positive predictions and is calculated using Equation (4).

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

Recall (sensitivity) is the proportion of ground-truth positives correctly identified by the model and is calculated using Equation (5).

$$Sensitivity = \frac{TP}{TP + FN} \quad (5)$$

In Equations (4) and (5), TP, FP, and FN denote true positives, false positives, and false negatives, respectively. It should also be specified whether counts are performed at the pixel level or at the matched object level. In this section, the precision and sensitivity in the validation tables are based on object matching, while their counterparts in the test tables are based on pixel counts.

Average precision (AP) summarizes the precision–recall curve generated by varying the confidence threshold. mAP50 represents the average precision at an IoU matching threshold of 0.50, while mAP50–95 represents the average of the AP values calculated at ten IoU thresholds ranging from 0.50 to 0.95. In the single-class polyp problem, the average across classes is effectively reduced to a single class. The inclusion of high IoU thresholds requires not only the approximate location of the target but also a tighter overlap of the mask. However, mAP50–95 does not measure the error of the boundaries in physical millimeters. mAP evaluates detection and instance segmentation together. Dice per image and foreground IoU examine region overlap, while boundary distances examine contour errors. In this study, validation mask AP metrics were used for model selection, while pixel and boundary metrics on the test set were used to evaluate the performance of the selected weights on held-out images. Since Dice

is equivalent to the F1 score in binary pixel classification, it was not repeated as a separate, independent metric.

3 Deep learning approaches and related research

3.1 Encoder-decoder based networks

U-Net features a symmetric encoder–decoder architecture that learns contextual features in the contracting encoder path while reconstructing spatial resolution in the expanding decoder path. The skip connections between corresponding encoder and decoder layers facilitate the integration of high-level semantic features with low-level spatial details, thereby contributing to the preservation of fine boundaries in particular. This architecture has become a fundamental approach in biomedical image segmentation due to the need for learning with a limited number of labeled examples and the requirement for detailed localization (Ronneberger et al., 2015). However, the significant variability of polyps in terms of size, shape, color, and texture, the low contrast between lesion boundaries and the surrounding mucosa, and the limited distinguishing features in small polyps can challenge the standard U-Net architecture’s capacity for representation and boundary detection. Therefore, rather than simply increasing the network depth, various design strategies have been developed, such as processing multi-scale features, utilizing attention mechanisms, enhancing boundary information, modeling contextual relationships, and improving computational efficiency. Subsequent encoder–decoder-based approaches, such as PraNet, HarDNet-MSEG, Li-SegPNet, DCRNet, ResUNet++, and SANet, aim to address these limitations in polyp segmentation by extending U-Net’s fundamental structure with different architectural components and feature processing strategies.

PraNet refines the initial map, constructed from high-level features, using reverse attention mechanisms. Fan et al. (2020) proposed the combined use of region and boundary cues as a solution to the diversity in polyp appearance. This approach demonstrates the benefit of refocusing on areas not covered or left ambiguous by the initial prediction. It highlights that, in polyp segmentation, it is necessary to investigate not only the selection of a robust backbone but also how the decoder design and boundary information are utilized.

HarDNet-MSEG balances segmentation quality and processing speed by using the HarDNet encoder and a simple decoder. Huang, Wu, and Lin (2021) reported an average Dice score above 0.90 and 86 FPS in their experimental setup. The study demonstrates that an efficiency-focused design can also be applied to the encoder–decoder family. The significance of speed measurements should be evaluated in conjunction with the resolution and hardware used.

Li-SegPNet aims for lightweight segmentation through modified attention blocks and multi-scale context processing. Sharma, Gautam, Maji, Pachori, and

Balabantaray (2023) examined its generalizability on external datasets in addition to evaluations on Kvasir-SEG and CVC-ClinicDB. DCRNet, on the other hand, utilizes both intra-image and inter-image contextual relationships. Thus, the representation of a pixel is not limited solely to the local neighborhood of the image in which it is located (Yin, Liang, Ma, & Guo, 2022).

ResUNet++ processes multi-scale features by combining residual units, Atrous Spatial Pyramid Pooling (ASPP), Squeeze-and-Excitation (SE), and attention components. Jha et al. (2019) presented comparisons between U-Net and ResUNet on Kvasir-SEG and CVC-ClinicDB within their own experimental setup. SANet, on the other hand, addresses color variations, loss of detail in small polyps, and foreground-background imbalance. The color transformation process, the attention module that reduces background noise in shallow features, and the probability adjustment strategy applied during inference are the key components of this design (Wei et al., 2021). These studies demonstrate that, in addition to network depth, feature fusion and training or inference strategies must also be evaluated.

3.2 Transformer and hybrid models

The goal of Transformer-based methods is to represent relationships between distant regions. When used in conjunction with convolutional networks, the aim is to combine local texture information with a broader context. FCB-SwinV2 is one example that examines polyp segmentation by combining convolutional and Transformer components (Fitzgerald & Matuszewski, 2023). Wang, Wang, and Sun (2025), on the other hand, proposed an architecture that incorporates a ConvNeXt-based encoder, cross-attention, and Transformer components in the decoder. These designs demonstrate that different information flows are being explored to improve segmentation quality.

Polyp-PVT combines the Pyramid Vision Transformer encoder with the Cascaded Fusion (CFM), Camouflage Identification (CIM), and Similarity Aggregation (SAM) modules. CFM aggregates semantic and spatial information from high-level features, while CIM identifies polyp information in low-level features. SAM contributes to the integration of different feature levels (Dong et al., 2023). The acronym SAM here refers to the similarity aggregation module and is not the same architecture as the Segment Anything Model. ColonFormer, on the other hand, processes long-range relationships and multi-level features using a lightweight transformer encoder, a hierarchical decoder, and a boundary-enhancing module (Duc, Oanh, Thuy, Triet, & Dinh, 2022).

Research is also being conducted on adapting pre-trained foundation models, trained on large-scale image data, for polyp segmentation. In their study Polyp-SAM, Li, Hu, and Yang (2023) compared fine-tuning only the mask decoder while keeping the encoders frozen with jointly fine-tuning the encoders and decoder of the Segment Anything Model. In the study, bounding boxes derived

from ground-truth labels were used as prompts during training and evaluation. Therefore, the results should be interpreted in conjunction with this prompt condition, which provides information about the lesion location. This setup is not directly equivalent to YOLO models that automatically generate boxes and masks from images. The need for fine-tuning, how the prompt is obtained, and the computational cost on the target hardware should be evaluated together when deploying foundation models into practice. The pre-training, input resolution, and feature fusion scheme used in hybrid models should be evaluated collectively. The convolutional and transformer branches of FCB-SwinV2 are examples of this integration.

3.3 Instance Segmentation with YOLO

YOLO is a detection approach that estimates object locations and classes in a single network evaluation (Redmon, Divvala, Girshick, & Farhadi, 2016). In the segmentation extension, object masks are added to these outputs. The idea of combining common mask prototypes with object-specific coefficients is described in YOLACT through two parallel branches (Bolya, Zhou, Xiao, & Lee, 2019). In the Ultralytics implementation, detection and mask generation are also performed within the same network (Ultralytics, 2025). The workflow is summarized in Figure 1. The backbone extracts feature maps at different resolutions from the input image. Feature fusion layers combine detailed location information with a broader context. The segmentation head generates mask prototypes and their weights for each object, in addition to bounding box and class predictions. A weighted combination of these prototypes forms the object-specific mask response. Selected bounding boxes are transformed to the output image coordinates via rescaling steps. This allows separate masks to be generated for multiple objects within the same image.

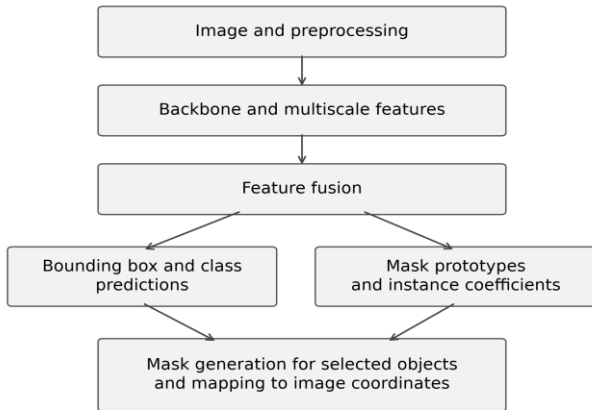


Figure 1. Conceptual workflow of YOLO instance segmentation.

An examination of the model configurations used in this study reveals that the YOLOv8 architecture employs C2f blocks as its basic feature processing units, while the YOLO11 architecture uses C3k2 blocks. YOLO11 also includes the C2PSA block, which is based on an attention mechanism (Ultralytics, n.d.). In both architectures, the SPPF module integrates multi-scale contextual information, fuses representations at different resolution levels through upsampling and feature fusion, and transfers the resulting features to the segmentation head at three different scales. In the YOLO11 configuration, these feature maps correspond to resolution levels that are downsampled by a factor of 8, 16, and 32, respectively, relative to the input image. The fact that the C2f and C3k2 blocks have different internal connection arrangements and that the YOLO11 architecture includes a C2PSA module indicates that the two model families adopt different architectural approaches in terms of feature extraction and feature reorganization (Ultralytics, 2025). However, since these components were not independently modified and compared in this study, it is not possible to directly attribute the observed performance differences to the effect of a specific architectural component. The variants n, s, m, l, and x scale the network’s depth and channel width. The same scale name does not imply an equal number of parameters across the two families. When adapting the pre-trained network to the polyp task, the number of classes is reduced to one, while the instance mask generation mechanism is preserved.

Sahafi, Koulaouzidis, and Lalinia (2024) investigated the segmentation of polypoid lesions in wireless capsule endoscopy images using YOLOv8. This study is an application of instance segmentation in gastrointestinal images. Capsule images constitute a distinct data domain compared to standard colonoscopy. As an example of a direct colonoscopic comparison, Xu, Liu, and Tang (2026) evaluated the YOLOv8-Seg and YOLOv11-Seg models alongside other segmentation networks on the Kvasir-SEG dataset in their BAR-UNet study. In this study, object masks were merged into a common foreground mask to calculate Dice and IoU. This detail clarifies under which transformation YOLO outputs can be compared with semantic segmentation networks.

3.4 Context of the Comparison in the Literature

When the studies under review are evaluated collectively, it cannot be said that a single architectural family meets all the requirements for polyp segmentation. Detailed representation of boundaries, the ability to operate on limited hardware, robustness against external data, and the production of consistent output throughout the video are different priorities. Sánchez-Peralta, Picón, Sánchez-Margallo, and Pagador (2020), who investigated the effect of data augmentation, also present an evaluation showing that method performance cannot be reduced solely to the network architecture. Examining different scales under a common protocol makes only a portion of this multidimensional choice

controllable. Table 1 summarizes the technical contributions and scope of evaluation of the studies.

Table 1. Comparison of Proposed Approaches to Polyp Segmentation

Method and source	Key contribution	Datasets	Metric	Comparison limitation
ResUNet++ Jha et al. (2019)	Residual blocks, ASPP, and channel attention modules	Kvasir-SEG and CVC-ClinicDB	Dice, mIoU	Different training and testing protocols
PraNet Fan et al. (2020)	Combining region and boundary information using reverse attention	Kvasir-SEG and CVC datasets	Dice, IoU	Different training and test splits
SANet Wei et al. (2021)	Color shift, shallow attention, and probability correction	Kvasir-SEG and CVC datasets	Dice, IoU	Different hardware and data splits
HarDNet-MSEG Huang et al. (2021)	An efficient encoder and a simple decoder	Kvasir-SEG, CVC-ClinicDB, and external datasets	Dice, IoU, FPS	Speed depends on hardware and resolution
DCRNet Yin et al. (2022)	Intra-image and inter-image contextual relationships	Kvasir-SEG, EndoScene and PICCOLO	Dice, IoU	Not retrained in this study
ColonFormer Duc et al. (2022)	Lightweight transformer encoder and hierarchical decoder	Kvasir-SEG, CVC datasets, and ETIS-Larib	Dice, IoU	Different resolution settings
FCB-SwinV2 Fitzgerald and Matuszewski (2023)	Parallel convolutional and SwinV2 branches	Kvasir-SEG and CVC-ClinicDB	Avg. Dice	Data splitting and video-level dependencies matter
Polyp-PVT Dong et al. (2023)	PVT encoder; CFM, CIM, and SAM modules	Kvasir-SEG, CVC, ETIS-Larib	Dice, IoU, MAE	Different pretraining and data splits
YOLOv8 Sahafi et al. (2024)	Instance segmentation of polypoid lesions	Wireless capsule endoscopy	Mask mAP	Different imaging modality from colonoscopy
Polyp-SAM Li et al. (2023)	Fine-tuning for the encoder and decoder	Kvasir-SEG and multicenter data	Dice, IoU	Uses box prompts derived from ground-truth labels
BAR-UNet Xu et al. (2026)	Boundary and appearance consistency; YOLO-Seg comparisons	Kvasir-SEG and external datasets	Dice, IoU, boundary	Instance masks are merged into a foreground mask

4 Experimental Design and Methods

4.1 Dataset

In this study, the open-access Kvasir-SEG dataset, developed for pixel-level segmentation of colorectal polyps, was used. The dataset consists of 1,000 endoscopic images from the “polyp” class of the Kvasir dataset—collected at the Vestre Viken healthcare facility in Norway—and their corresponding segmentation masks. The polyp boundaries were manually annotated by a team consisting of an engineer and a medical doctor, and subsequently reviewed by an experienced gastroenterologist. In the masks, the polyp regions represent the foreground, while the remaining pixels represent the background. Additionally, bounding box coordinates derived from the segmentation masks are provided (Jha et al., 2020). The dataset is divided into 700 training (70%), 150 validation (15%), and 150 test (15%) examples at the image level. Comparisons were conducted on a common data split by using the same subsets for all models. For the test evaluation, the file lists from the fixed split used in model training were used as the basis. No random resplitting was performed. Each image’s unique filename was used as the sample ID, and the corresponding mask was matched using the same ID. To ensure the traceability and reproducibility of the splitting, an index record was created containing the sample IDs, subset assignments, image and mask file paths, and SHA-256 file hashes.

As part of the data leakage audit, the sample identifiers of the training, validation, and test sets were compared, and no common file names were found across the subsets. Furthermore, a comparison based on SHA-256 hashes revealed no images with identical file contents across different subsets. However, these audits evaluate duplicates at the file level. These checks do not prove that different or visually similar frames belonging to the same patient or video recording are not present across subsets. Since the data used did not contain identifiers that allow grouping at the patient or video level, the independence of the split at the patient level could not be verified. The training set was used to learn the model parameters, while the validation set was used to monitor the training process and select the model weights. The test set was set aside for the final performance evaluation. No optimization of weights, hyperparameters, or decision thresholds was performed on this set. The target input size was set to 640 pixels for all models. Image resizing and padding to preserve the aspect ratio were performed in the Ultralytics data processing pipeline. This prevented geometric distortions that could have resulted from directly stretching the original images to a square format. During the testing phase, images were prepared to fit a 640×640 -pixel input area, while prediction masks were generated at the original image resolution for comparison with reference masks.

Data augmentation operations were applied only to the training set after data splitting. Since the transformations were generated online during training, the

augmented images were not transferred as separate instances to the validation or test sets. The correspondence between the image and the segmentation labels was preserved during geometric transformations. In the recorded training configuration, the horizontal flip probability was set to 0.5, the translation parameter to 0.1, and the scaling parameter to 0.5. The parameters used for hue, saturation, and brightness changes in the HSV color space were 0.015, 0.7, and 0.4, respectively. The probability of Mosaic data augmentation was set to 1.0, and this transformation was disabled during the last 10 epochs. MixUp and Copy-Paste transformations were not used. Blur, MedianBlur, ToGray, and CLAHE transformations were also applied with low probability during training sessions.

Random data augmentation was not applied during the validation and testing phases. No additional image enhancement or runtime data augmentation was used during testing. Using the same data splitting and augmentation configuration for all models ensured consistency in the experimental conditions. However, since the individual contributions of the augmentation methods to performance were not examined through separate ablation experiments, the results were evaluated based on the training configuration as a whole.

4.2 Model Families and Training Environment

The training was initiated using pre-trained weights from the n, s, m, l, and x segmentation variants of the YOLOv8 family and the same five scale variants of the YOLO11 family. In this section, abbreviations such as YOLOv8n and YOLO11s refer to segmentation models. The training of the ten models was performed on an NVIDIA A100-SXM4-40GB GPU using Python 3.12.12, PyTorch 2.9.0+cu126, and Ultralytics 8.3.233 (Ultralytics, 2025). Each model was trained for 70 epochs using batches of four images. Pre-trained segmentation weights, the deterministic run option, and automatic mixed-precision training were used. The optimizer = auto option in the configuration was resolved in the logs as AdamW, with an initial learning rate of 0.002 and a momentum parameter of 0.9. Table 2 lists the learning rate actually selected, rather than the default $lr_0 = 0.01$ specified in the file. The weight decay is 0.0005, the warm-up period is three epochs, and the patience value is 100. All training runs completed the planned 70 epochs and were not shortened by early stopping.

Table 2. Common Training and Evaluation Settings

Component	Recorded Value
Data split	700 for training, 150 for validation, 150 for testing
Training	70 epochs; batch = 4; imgsz = 640
GPU	NVIDIA A100-SXM4-40GB
Software	Ultralytics 8.3.233; Python 3.12.12; PyTorch 2.9.0+cu126
Initialization	Pre-trained segmentation weights
Optimization	AdamW; initial lr = 0.002; momentum = 0.9
Regularization	weight_decay = 0.0005
Reproducibility	deterministic = true
Other settings	amp = true; mask_ratio = 4; overlap_mask = true
Mosaic	Probability 1.0; disabled for the last 10 epochs
Validation summary	Highest validation mask mAP50–95

4.3 Loss Functions and Evaluation Protocol

Box, segmentation, classification, and distribution focal localization losses were tracked separately throughout training. The box and Distribution Focal Loss (dfl_loss) are aimed at learning the localization representation, the segmentation loss at learning the mask prediction, and the classification loss at learning the target class. In evaluation, task metrics calculated on the validation set were used instead of the training losses. The primary metric for comparing validation records was set as mask mAP50–95. For each model, the precision, sensitivity, and mask AP values from the epoch where this metric was highest were provided together. In the test, the best.pt checkpoint saved by each training session was used. In the Ultralytics 8.3.233 application, the checkpoint selection score for segmentation is the sum of the box and mask mAP50–95 values.

4.4 Evaluation of Computational Cost

The number of parameters and GFLOPs were taken from the fused network summaries in the logs. These represent, respectively, the model capacity and the software’s estimated computational cost. They are not measures of energy consumption or working memory. The single-class network summaries used in the experiment were used as the basis, rather than the number of general-purpose classes in the model files. Training time was calculated from the “total time” field in the last row of the CSV file. Inference time is the average forward pass time per image reported in the “best.pt” validation file at the end of training. Preprocessing, postprocessing, video streaming, and screen rendering are not

included in this value. Since no repeated speed measurements were taken, the times are presented as the recorded values for the session.

4.5 Pixel and Boundary Evaluation on the Held-Out Test Set

All models used in the study were evaluated on a separate set of 150 test images under the same inference conditions. Float32 inference was performed on a CPU using Ultralytics 8.3.233, PyTorch 2.14.0, and Python 3.12.0. Images were processed individually, with the target size set to 640 pixels, the confidence threshold set to 0.25, the bounding box IoU threshold for non-maximum suppression set to 0.70, and the maximum number of predictions per image set to 300. The original reference masks were binarized such that pixels with grayscale values above 127 were considered foreground. Prediction masks were calculated at the original image resolution. A mask logit greater than zero was considered foreground. This rule corresponds to a threshold of 0.5 for the sigmoid probability. The union of object masks that passed the confidence threshold was evaluated as a single foreground mask.

In boundary evaluation, the bidirectional Euclidean distances between the inner contours extracted using 8-neighbor erosion were combined. The 95th percentile of these distances was calculated as HD95, and the arithmetic mean was calculated as the average symmetric surface distance (ASSD). Measurements were obtained in pixels at the original resolution; values normalized to the image diagonal were also reported to minimize the effect of different image sizes. Since the physical pixel spacing was not defined, the distances were not interpreted in millimeters. All test instances contain at least one foreground pixel. In the case of an empty prediction, Dice, IoU, precision, and recall were set to zero; HD95 and ASSD were penalized by the image diagonal length. Thus, failed predictions were not excluded from the mean calculations. The 95% confidence intervals for mean Dice and IoU were calculated using 10,000 image-level bootstrap resamples, each containing 150 images sampled with replacement. Dice differences relative to YOLO11s were calculated using paired resampling of the same images.

5 Experimental findings

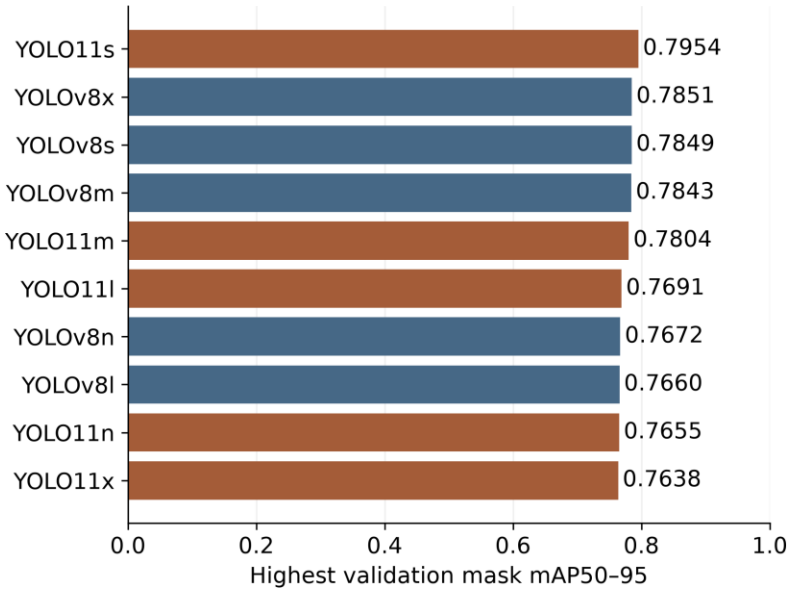
5.1 Mask performance and model ranking

The highest validation mask mAP50–95 values are shown in Table 3. YOLO11s took first place, reaching a value of 0.79535 at the 62nd epoch. At this epoch, the mask mAP50 value was 0.96761, with a precision of 0.96444 and a recall of 0.87333. YOLOv8x, in second place, achieved a value of 0.78507; YOLOv8s, in third place, achieved 0.78488; and YOLOv8m, in fourth place, achieved 0.78430. The difference between the top two models is 1.028 percentage points. The difference between YOLOv8x and YOLOv8s, however, is only 0.019 percentage points.

Table 3. Validation mask metrics for the selected epochs

Model	Epoch	Precision	Recall	mAP50	mAP50–95
YOLO11s	62	0.96444	0.87333	0.96761	0.79535
YOLOv8x	67	0.92953	0.89333	0.94855	0.78507
YOLOv8s	65	0.92472	0.90083	0.95051	0.78488
YOLOv8m	70	0.93562	0.92000	0.94873	0.78430
YOLO11m	68	0.91306	0.91333	0.95812	0.78044
YOLO11l	66	0.93657	0.91333	0.95824	0.76912
YOLOv8n	63	0.91199	0.89813	0.93879	0.76721
YOLOv8l	70	0.93544	0.88667	0.94830	0.76596
YOLO11n	53	0.94030	0.88667	0.94730	0.76548
YOLO11x	66	0.94682	0.88000	0.95746	0.76379

Figure 2 shows the ranking of the ten models. The close scores of YOLOv8x, YOLOv8s, and YOLOv8m illustrate how small the differences in a ranking based on a single training run can be. In particular, the 0.019 percentage point difference between YOLOv8x and YOLOv8s does not, on its own, constitute a practical reason for preference. YOLO11s achieved the highest validation mask mAP50–95 under this protocol.

**Figure 2.** The models' highest validation mask mAP50–95 values.

When precision and sensitivity are evaluated together, it is observed that the highest mAP value does not coincide with the highest sensitivity at every operating point. In the selected epochs, YOLOv8m’s sensitivity is 0.92000, while YOLO11s’ sensitivity is 0.87333. YOLO11s demonstrates higher precision. In the epoch selected using the mask metric, the box mAP50–95 value for YOLO11s is 0.77377, while that for YOLOv8x is 0.78008.

5.2 Training Dynamics and Sensitivity to Epoch Selection

Figure 3 shows the change in mask mAP50–95 for each model over 70 epochs. The rising, fluctuating, and plateau regions in the curves illustrate the model’s learning trajectory on this validation set. Disabling Mosaic augmentation during the last 10 epochs is a standard protocol step that alters the data distribution in the final phase of training. In the graph, this transition is indicated by the dashed line between the 60th and 61st epochs.

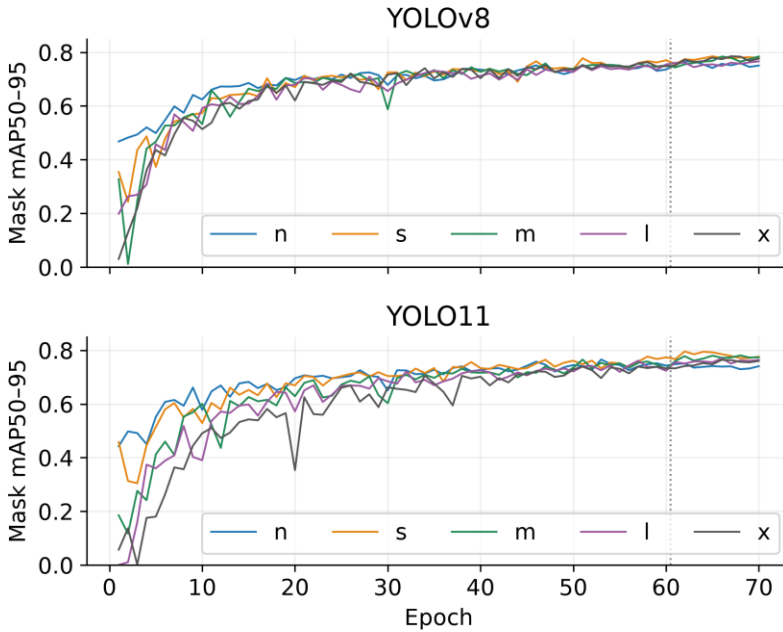


Figure 3. Validation mask mAP50–95 curves for the YOLOv8 and YOLO11 models.

The rankings change when the results from the last epoch are used. YOLOv8m ranks first with a score of 0.78430 at epoch 70. YOLOv8s has a score of 0.77879, YOLOv8x has a score of 0.77606, and YOLO11s has a score of 0.77581. The difference between YOLO11s’s highest value and its final value is 1.954 percentage points. For YOLO11n, this difference reaches 2.513 percentage points.

Table 4. Validation mask mAP50–95 by epoch selection

Model	Maximum	Final epoch	Mean of final 10 epochs
YOLOv8n	0.76721	0.75047	0.75459
YOLOv8s	0.78488	0.77879	0.77622
YOLOv8m	0.78430	0.78430	0.76785
YOLOv8l	0.76596	0.76596	0.75702
YOLOv8x	0.78507	0.77606	0.77204
YOLO11n	0.76548	0.74035	0.73930
YOLO11s	0.79535	0.77581	0.78143
YOLO11m	0.78044	0.77344	0.77181
YOLO11l	0.76912	0.76392	0.75828
YOLO11x	0.76379	0.76008	0.75150

When the averages for the last 10 epochs in Table 4 are evaluated together, it can be seen that YOLO11s performs the best.

5.3 Model Capacity and Runtime

Table 5 presents the number of parameters, computational cost, and runtime of the models. YOLO11s, with approximately 10.07 million parameters and 32.8 GFLOPs, achieved the highest mask mAP50–95 value in this comparison. Although YOLOv8x requires approximately 71.72 million parameters and 327.9 GFLOPs, it has a lower maximum mask mAP50–95 value. The difference between these two models demonstrates that increasing computational capacity does not automatically translate into improved task performance.

Table 5. Model capacity and times recorded during the A100 validation session

Model	Parameters (millions)	GFLOPs	Training (min)	Inference (ms)
YOLOv8n	3.26	11.3	17.85	1.7
YOLOv8s	11.78	39.9	18.15	1.7
YOLOv8m	27.22	104.3	21.38	3.0
YOLOv8l	45.91	210.1	24.56	3.3
YOLOv8x	71.72	327.9	25.94	4.7
YOLO11n	2.83	9.6	21.99	2.3
YOLO11s	10.07	32.8	22.36	2.3
YOLO11m	22.34	112.9	26.04	3.4
YOLO11l	27.59	131.8	34.45	3.7
YOLO11x	62.00	295.9	36.33	5.4

Figure 4 illustrates the relationship between capacity and mask performance. When considering both the number of parameters and the highest mask mAP50–95, YOLO11s achieved a higher score than larger models.

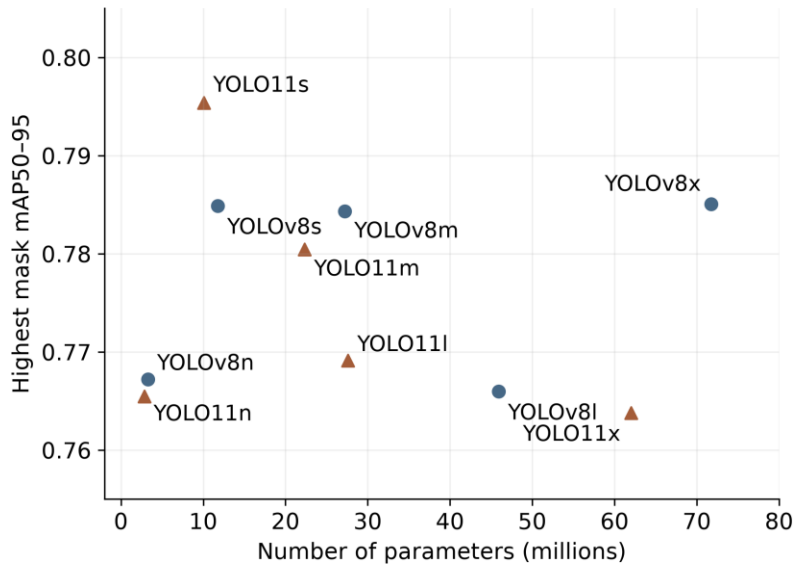


Figure 4. Relationship between the number of parameters and the highest validation mask mAP50–95.

The recorded inference times are 1.7 ms per image for YOLOv8n and YOLOv8s, and 2.3 ms for YOLO11n and YOLO11s. YOLOv8x has a value of 4.7 ms, while YOLO11x has a value of 5.4 ms. In these sessions, the YOLO11

model at the same scale did not demonstrate a shorter inference time compared to its YOLOv8 counterpart. The calculated training times range from 17.85 to 36.33 minutes. Data transfer and environment setup are not included in these times.

5.4 Results on the Held-Out Test Set

Table 6 shows the pixel metrics for each model on a separate set of 150 test images. YOLO11s, the reference candidate selected from the validation results, achieved the highest average Dice score of 0.8623 (95% confidence interval 0.8241–0.8958). The same model’s average foreground IoU value was 0.8040 (95% confidence interval 0.7634–0.8405). YOLOv8l demonstrated similar overlap performance with a Dice score of 0.8616 and an IoU of 0.8046, achieving the highest value in terms of average IoU. This slight difference in the Dice and IoU rankings reflects the impact of image-level error distributions on the mean. The test metrics correspond to binary foreground predictions compared to the original ground-truth masks. Table 7 presents boundary errors and pixel specificity as complementary metrics. For HD95 and ASSD, a smaller value indicates lower boundary error. Distances expressed as a percentage of the diagonal summarize different resolutions on a common relative scale. The diagonal penalty applied to empty predictions prevents the optimism that could result from excluding failed images from the average.

For YOLO11s, the average HD95 was calculated as 66.83 pixels and 7.98% of the diagonal. Among the models used in the study, the lowest average normalized HD95 was achieved with YOLO11s, while the lowest average normalized ASSD was achieved with YOLO11n. YOLO11s produced empty predictions in 2 out of 150 images. This demonstrates that a single high average score does not necessarily imply reliable segmentation across all images. Figure 5 shows the average Dice values along with 95% bootstrap confidence intervals at the image level. The intervals reflect the uncertainty of the fixed weights in this test sample. Model differences were also calculated in a paired manner on the same images.

Table 6. Average pixel metrics per image in the test set

Model	Dice / F1	Foreground IoU	Precision	Recall
YOLO11s	0.8623	0.8040	0.8607	0.9104
YOLOv8l	0.8616	0.8046	0.8629	0.9091
YOLOv8x	0.8580	0.7987	0.8518	0.9078
YOLOv8m	0.8533	0.7987	0.8449	0.8891
YOLO11l	0.8523	0.7952	0.8483	0.9083
YOLO11n	0.8485	0.7808	0.8233	0.9369
YOLO11m	0.8423	0.7779	0.8289	0.9144
YOLOv8n	0.8330	0.7779	0.8408	0.8660
YOLOv8s	0.8289	0.7750	0.8267	0.8836
YOLO11x	0.8251	0.7606	0.8123	0.8881

Table 7. Complementary pixel and boundary metrics in the test set

Model	Specificity	MAE	HD95 (% of diagonal)	ASSD (% of diagonal)	Empty predictions
YOLO11s	0.9680	0.0413	7.98	3.40	2
YOLOv8l	0.9692	0.0388	8.47	3.61	2
YOLOv8x	0.9696	0.0412	8.80	4.08	3
YOLOv8m	0.9691	0.0372	10.14	5.84	6
YOLO11l	0.9688	0.0446	8.26	3.51	2
YOLO11n	0.9471	0.0511	8.95	3.20	1
YOLO11m	0.9553	0.0506	9.20	3.42	1
YOLOv8n	0.9772	0.0395	10.47	6.05	6
YOLOv8s	0.9604	0.0493	11.61	6.95	7
YOLO11x	0.9585	0.0486	11.59	6.02	5

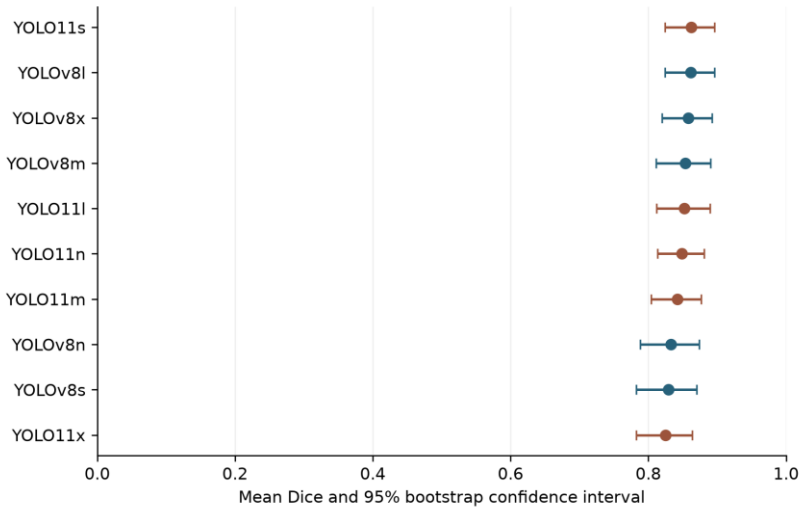


Figure 5. Average Dice score on the test set and 95% bootstrap confidence intervals at the image level.

5.5 Qualitative Error Analysis in Test Images

Figure 6 shows the original image, the reference mask, and the predictions from YOLOv8s and YOLO11s for three images in the test set. The examples were selected as the 15th, 75th, and 135th images when the test images were sorted by the image-level Dice scores of the preselected reference model, YOLO11s. This selection highlights low, medium, and high levels of overlap using the same criterion; the three examples were not used to represent the entire error distribution.

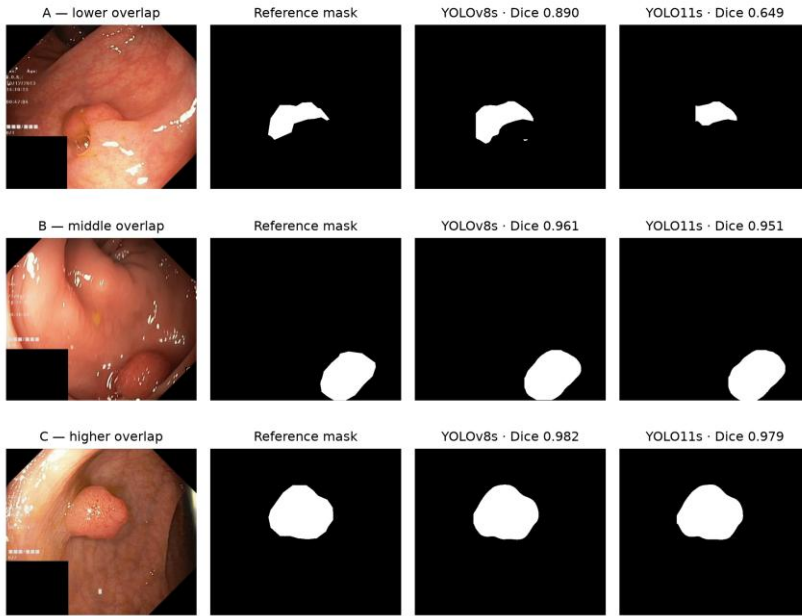


Figure 6. Examples of different overlap levels in the test set.

Samples A, B, and C represent low, medium, and high levels of overlap, respectively. Presenting the predictions of two models side by side in the same image allows us to examine how missed regions and boundary oversegmentation vary from model to model. Qualitative results should be evaluated in conjunction with aggregate pixel and boundary metrics. The predictions used in the visual analysis were generated using the same weights and thresholds as those used in the quantitative evaluation. Dice and IoU values were calculated from the original-resolution binary masks, not from scaled panels used for display.

6 Discussion

The findings of this study highlight three key points. First, increasing the model scale did not result in a consistent improvement in segmentation performance. While YOLO11s achieved the highest value for the primary validation metric—mask mAP50–95—larger m, l, and x variants from the same family yielded lower results. Similarly, in the YOLOv8 family, an increase in the number of parameters did not translate linearly into validation performance. Second, the ranking based on validation metrics did not fully align with the pixel-based ranking on the held-out test set. While YOLO11s achieved the highest average Dice score on the test set, the highest average IoU was obtained with YOLOv8l. Third, the fact that the test Dice difference between YOLO11s and YOLOv8l is quite small, and the confidence interval for the paired difference does not support a claim of statistical superiority, indicates that one should not conclude that one model family is generally superior to another based on a single

training session. When these findings are evaluated together, it becomes clear that model selection should be based not only on the architectural version or the number of parameters, but also on the validation metric, test performance, and computational cost.

These results show that higher model capacity does not automatically result in better representation on a dataset of limited size. The fact that YOLO11s, with approximately 10 million parameters, achieves higher scores on the main validation metric than larger models demonstrates that an increase in capacity alone is not sufficient in this experimental setup. However, since the architectural differences between YOLOv8 and YOLO11 were not examined through ablation experiments, the observed results cannot be directly attributed to the influence of C3k2, C2PSA, or any other component. The findings illustrate the behavior of the examined off-the-shelf model configurations under this data and training protocol, rather than the superiority of specific architectural components.

Approaches in the literature focus on different design challenges. PraNet addresses edge detail, HarDNet-MSEG addresses processing efficiency, and FCB-SwinV2 addresses the joint representation of local and global context. Rather than proposing a new segmentation architecture, the experimental section of this chapter examines the scale selection of off-the-shelf YOLO segmentation architectures under a common protocol. When these two levels are evaluated together, it becomes evident that success in polyp segmentation depends not only on architectural design but also on data preparation, training configuration, model selection criteria, and the evaluation protocol—all of which are decisive in interpreting the results.

Similar to the evaluation conducted by Xu et al. (2026), who converted object masks into a common foreground mask, in this study, test Dice and IoU values were calculated using binary foreground masks created by merging object masks that passed the confidence threshold. This transformation enables YOLO models that produce instance segmentation to be evaluated using pixel-based segmentation metrics. However, the use of the same dataset and the same metric names does not, by itself, constitute a fair comparison between studies. Differences in training and testing splits, pre-training strategies, input resolution, data augmentation, prompt usage, and decision thresholds can all influence the results. Therefore, no direct ranking of superiority has been established based on the numerical results reported in different publications.

Qualitative examples demonstrate that locating the polyp region and accurately defining its boundary exhibit distinct error characteristics. Even if a prediction correctly identifies the general location of a polyp, masks that extend into surrounding tissue or exclude part of the target can affect Dice, IoU, and boundary metrics in different ways. Extra masks can lead to the same target being represented as multiple objects, while overextended masks can result in

surrounding tissue being included in the target region. Therefore, relying on a single overlap metric alone is insufficient to fully explain the error structure. Future evaluations that are broken down into subgroups—such as small lesions, low contrast, reflections, partial visibility, and boundary ambiguity—could provide a more detailed understanding of the image conditions under which models fail.

In real-time or video-based applications, image-level segmentation performance alone is not sufficient. In addition to the targeted sensitivity level, false alarm duration, first-detection delay, and the temporal consistency of masks generated across consecutive frames must also be evaluated. In this study, the forward pass time recorded on the A100 represents only the model inference component of a system’s total latency.

This study has some limitations. The analysis is based on a single data split and a single training run for each model. Although the separate test set allowed for evaluation on images independent of the validation process, it does not demonstrate that the small differences between models would be preserved under different initial conditions and in independent training runs. Image-level bootstrap confidence intervals quantify a portion of the uncertainty arising from test sampling but do not account for the variability resulting from independent training runs. Furthermore, the models were compared on the same test set, and no multiple comparison correction was applied to the confidence intervals of the paired differences.

7 Conclusion

This chapter presents the theoretical foundations of colorectal polyp segmentation, relevant deep learning approaches, and experimental findings regarding ten scale variants of the YOLOv8 and YOLO11 families. In the common training setup, the highest validation mask mAP50–95 value was achieved with YOLO11s at 0.79535. YOLO11s, which produced the highest average Dice score on a separate set of 150 test images, achieved a Dice score of 0.8623 and a foreground IoU of 0.8040. YOLOv8l’s Dice score of 0.8616 is very close to this result.

The findings demonstrate that model selection cannot be based solely on version name or the number of parameters. Validation AP values, pixel overlap, and boundary errors in the test set reflect different dimensions of segmentation performance. Therefore, evaluation metrics, checkpoint selection rules, and computational conditions must all be considered together in model comparisons. Although the test evaluation based on original masks complements the validation findings, it should be noted that the results are limited to a single dataset and a single training run for each model.

In future studies, the stability and generalizability of the results will be evaluated using repeated training runs with different random seeds and independent external datasets at the patient or video level. Examining architectures such as U-Net, ResUNet++, PraNet, and Polyp-PVT under the same data partitioning and comparable training budgets will enable a more comprehensive comparison among architectural families.

Disclosures and Statements

Acknowledgments and Institutional Support: This study utilized a publicly available, fully anonymized, open-source biomedical dataset made accessible to researchers as part of the first author’s doctoral education and academic research activities conducted within the Department of Computer Engineering at the Institute of Graduate Education, Sakarya University of Applied Sciences. The authors thank Sakarya University of Applied Sciences for the institutional and academic resources provided for the conduct of this study.

Conflict of Interest Statement: The authors declare that there is no potential or actual conflict of interest or financial relationship with any individual, institution, or organization regarding the research, preparation, and publication of this book chapter.

Ethical Review Statement: The Kvasir-SEG dataset used in this study is a publicly available, fully anonymized, open-source biomedical dataset made accessible to researchers. No clinical interventions or experimental procedures were performed directly on human or animal subjects as part of this study.

Author Contributions: Conceptual design and problem formulation of the study: S. Ö., Y. S. Ş., S. U.; Experimental infrastructure, model training, and collection of computational logs: Y. S. Ş.; Analysis of numerical data, creation of tables and figures: S. Ö.; Establishment of the theoretical framework, literature review, drafting of the manuscript, and critical revision: S. Ö., S. U., Y. S. Ş.; Academic supervision and study coordination: S. U. All authors have read the final manuscript and approved it for publication.

References

- Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). YOLACT: Real-time instance segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 9157–9166). Retrieved from https://openaccess.thecvf.com/content_ICCV_2019/html/Bolya_YOLACT_Real-Time_Instance_Segmentation_ICCV_2019_paper.html
- Dong, B., Wang, W., Fan, D.-P., Li, J., Fu, H., & Shao, L. (2023). Polyp-PVT: Polyp segmentation with pyramid vision transformers. *CAAI Artificial Intelligence Research*, 2, 9150015. doi:10.26599/AIR.2023.9150015
- Duc, N. T., Oanh, N. T., Thuy, N. T., Triet, T. M., & Dinh, V. S. (2022). ColonFormer: An efficient transformer based method for colon polyp segmentation. *IEEE Access*, 10, 80575–80586. doi:10.1109/ACCESS.2022.3195241
- Fan, D.-P., Ji, G.-P., Zhou, T., Chen, G., Fu, H., Shen, J., & Shao, L. (2020). PraNet: Parallel reverse attention network for polyp segmentation. In A. L. Martel, P. Abolmaesumi, D. Stoyanov, D. Mateus, M. A. Zuluaga, S. K. Zhou, D. Racoceanu, & L. Joskowicz (Eds.), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020* (pp. 263–273). Cham: Springer. doi:10.1007/978-3-030-59725-2_26
- Fitzgerald, K., & Matuszewski, B. (2023). FCB-SwinV2 transformer for polyp segmentation. arXiv preprint, arXiv:2302.01027. Retrieved from <https://arxiv.org/abs/2302.01027>
- Huang, C.-H., Wu, H.-Y., & Lin, Y.-L. (2021). HarDNet-MSEG: A simple encoder-decoder polyp segmentation neural network that achieves over 0.9 mean Dice and 86 FPS. arXiv preprint, arXiv:2101.07172. Retrieved from <https://arxiv.org/abs/2101.07172>
- Jha, D., Smedsrud, P. H., Riegler, M. A., Halvorsen, P., de Lange, T., Johansen, D., & Johansen, H. D. (2020). Kvasir-SEG: A segmented polyp dataset. In Y. M. Ro, W.-H. Cheng, J. Kim, W.-T. Chu, P. Cui, J.-W. Choi, M.-C. Hu, & W. De Neve (Eds.), *MultiMedia Modeling* (pp. 451–462). Cham: Springer. doi:10.1007/978-3-030-37734-2_37
- Jha, D., Smedsrud, P. H., Riegler, M. A., Johansen, D., de Lange, T., Halvorsen, P., & Johansen, H. D. (2019). ResUNet++: An advanced architecture for medical image segmentation. *2019 IEEE International Symposium on Multimedia (ISM)* (pp. 225–230). IEEE. doi:10.1109/ISM46123.2019.00049
- Li, Y., Hu, M., & Yang, X. (2023). Polyp-SAM: Transfer SAM for polyp segmentation. arXiv preprint, arXiv:2305.00293. <https://doi.org/10.48550/arXiv.2305.00293>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 779–788). Retrieved from

https://openaccess.thecvf.com/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html

- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (pp. 234–241). Cham: Springer. doi:10.1007/978-3-319-24574-4_28
- Sahafi, A., Koulaouzidis, A., & Lalinia, M. (2024). Polypoid lesion segmentation using YOLO-V8 network in wireless video capsule endoscopy images. *Diagnostics*, 14(5), 474. doi:10.3390/diagnostics14050474
- Sánchez-Peralta, L. F., Picón, A., Sánchez-Margallo, F. M., & Pagador, J. B. (2020). Unravelling the effect of data augmentation transformations in polyp segmentation. *International Journal of Computer Assisted Radiology and Surgery*, 15(12), 1975–1988. doi:10.1007/s11548-020-02262-4
- Sharma, P., Gautam, A., Maji, P., Pachori, R. B., & Balabantaray, B. K. (2023). Li-SegPNet: Encoder-decoder mode lightweight segmentation network for colorectal polyps analysis. *IEEE Transactions on Biomedical Engineering*, 70(4), 1330–1339. doi:10.1109/TBME.2022.3216269
- Ultralytics. (n.d.). *Ultralytics YOLO11*. Retrieved September 16, 2026, from <https://docs.ultralytics.com/models/yolo11/>
- Ultralytics. (2025). *Ultralytics* (Version 8.3.233) [Computer software]. Retrieved from <https://github.com/ultralytics/ultralytics/tree/v8.3.233>
- Wang, Z., Wang, Z., & Sun, P. (2025). Deep learning model for gastrointestinal polyp segmentation. *PeerJ Computer Science*, 11, e2924. doi:10.7717/peerj-cs.2924
- Wei, J., Hu, Y., Zhang, R., Li, Z., Zhou, S. K., & Cui, S. (2021). Shallow attention network for polyp segmentation. In M. de Bruijne, P. C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, & C. Essert (Eds.), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021* (pp. 699–708). Cham: Springer. doi:10.1007/978-3-030-87193-2_66
- World Health Organization. (2026, February 13). *Colorectal cancer*. Retrieved from <https://www.who.int/news-room/fact-sheets/detail/colorectal-cancer>
- Xu, Y., Liu, L., & Tang, M. (2026). BAR-UNet: A boundary-aware and appearance-robust framework for polyp segmentation. *Frontiers in Oncology*, 16, 1873111. doi:10.3389/fonc.2026.1873111
- Yin, Z., Liang, K., Ma, Z., & Guo, J. (2022). Duplex contextual relation network for polyp segmentation. *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)* (pp. 1–5). IEEE. doi:10.1109/ISBI52829.2022.9761402



CHAPTER 2

**A General Framework for Machine
Learning Model Selection Considering
Data Modalities, Computational
Constraints and Interpretability**

Hilal Çelik¹



¹ Dr., Sivas University of Science and Technology, ORCID: 0000-0001-5428-3411

1. Introduction and Theoretical Foundations

The successful deployment of machine learning frameworks relies fundamentally on systematically aligning algorithmic capacity with the underlying data-generating distribution (Lavie, Levi, & Kahn, 2026; Massaro, Maritati, Galiano, Birardi, & Pellicani, 2018; Şeker, Şadi). Rather than treating model complexity as unbounded, a principled methodology must be grounded in theoretical limits—most notably the No Free Lunch theorem and structural inductive biases—which dictate that no single architecture possesses universal superiority across all empirical settings (Du, Zhang, Jiang, Zeng, & Lu, 2025; Goldblum, Finzi, Rowan, & Wilson, 2024; Wilson, 2025). Building upon these foundational constraints, systematic algorithmic selection is governed by a multidimensional decision framework that simultaneously evaluates dataset topology, task formulation, computational budgets, and interpretability requirements (Li, Wu, Ling, & Tu, 2026; Vanschoren, 2017). Operationalizing this framework necessitates moving beyond ad-hoc heuristics into an end-to-end procedural workflow encompassing initial data and dimensionality diagnostics, constraint formalization, hypothesis space pruning, and leakage-free validation (Du et al., 2025; Haiyun He & Goldfeld, 2025). Coupled with post-hoc explainability protocols (such as SHapley Additive exPlanations (SHAP), Partial Dependence Plots (PDP), and Gradient-weighted Class Activation Mapping (Grad-CAM)), this decision pipeline provides a repeatable methodology that reconciles computational tractability with mechanistic plausibility across diverse predictive modeling tasks (Friedman, 2001; Lundberg, Allen, & Lee; Selvaraju et al., 2017).

1.1. The Model Selection Problem in Machine Learning

The predictive ability and generalizability of any machine-learning (ML) framework are intrinsically dependent upon the degree of correspondence between the mathematical assumptions of an algorithm and the data-generating distribution. In applied computational sciences and predictive modelling, a frequent methodological error is the unthinking usage of default algorithms with high parameter counts, especially complex deep neural networks or ensemble architectures, under the assumption that higher representational capacity guarantees better performance. Essentially, this premise fundamentally contradicts the No Free Lunch theorem (Goldblum et al., 2024; Ruben, Tong, Chaudhry, & Pehlevan, 2025; Sroka, 2026), which establishes that no single learning algorithm possesses a universal performance advantage across the space of all possible data distributions and objective functions. A model attains high predictive accuracy only when its inductive biases—that is, the implicit and explicit prior assumptions incorporated in its optimization objective and functional form—match well with the true structural properties of the target system. Consequently, choosing a model arbitrarily while disregarding factors such as dataset geometry, dimensionality, signal-to-noise dynamics, and feature dependencies directly leads to severe overfitting, underfitting, prohibitive computational overhead, or uninterpretable black-box artifacts (Du et al., 2025; Lavie et al., 2026; Li et al., 2026; Wilson, 2025).

1.2. Multidimensional Criteria for Algorithmic Selection

Tackling central problems in technique rather than data lies at heart of most fundamental challenges in applied machine learning. Choosing the right algorithm requires a systematic consideration of the properties of the underlying dataset, as well as implementation trade-offs (Goldblum et al., 2024). The model selection process should not be considered in isolation, but rather formulated as a pipeline directly linking intrinsic data properties to model shapes and deployment constraints (Wilson, 2025). The interaction of the dataset topology and geometry, the nature of the target task and the form of objective, computational and latency bounds (Li et al., 2026), and interpretability constraints governs this selection pipeline (Figure 1.1). The criteria are evaluated simultaneously so that the framework determines the best choice of algorithm, and a post-hoc check as well as an XAI check establishes generalization and interpretability (Mohammed & Kora, 2023a; Tınmaz, 2025; Vanschoren, 2017).

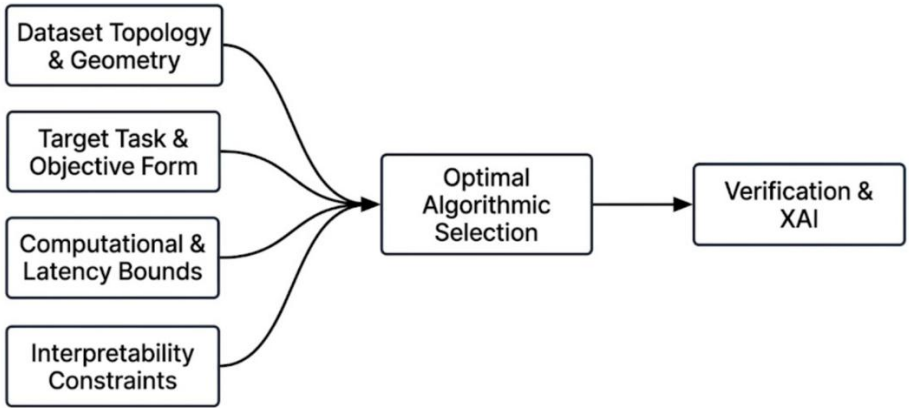


Figure 1.1. An End-to-End Framework for Data-Driven Model Selection and Explainable Verification.

Algorithm selection is fundamentally governed by measurable properties across both conceptual geometry and computational domains. Assessing architectural compatibility should not rely merely on post hoc empirical observations—namely, retrospective trial-and-error evaluations conducted only after model training. Instead, architectural compatibility must be determined a priori, guided by theoretical principles that account for data geometry, optimization objectives, computational resources, and explainability (Hollmann, Müller, Eggenberger, & Hutter; Huang et al., 2025; Qu, Holzmüller, Varoquaux, & Morvan, 2026). To structure this evaluation rigorously, the primary decision criteria governing algorithmic selection are delineated as follows:

Dataset Topology and Structural Characteristics: Determining the risk of overparameterization in sample-scarce regimes ($N < 10^3$) versus scalability requirements in large-scale domains ($N > 10^5$); managing the dimensionality ratio ($p \gg N$) through structural regularization (ℓ_1/ℓ_2 penalties) or manifold learning; and distinguishing tabular/heterogeneous matrices from grid-structured, sequential, or topological data formats (Kukačka, Golkov, & Cremers, 2017;

Lavie et al., 2026; Li et al., 2026; Ng, 2004; Xu, Zhang, Wang, Chang, & Liang, 2010) .

Task Formulation and Objective Function: Formalizing the optimization problem across continuous function approximation (regression), discrete boundary separation (classification under class imbalance), latent space projection (unsupervised dimensionality reduction), or constrained generative synthesis (Du et al., 2025; Goldblum et al., 2024; Haibo He & Garcia, 2009) .

Computational Complexity and Resource Budgets: Balancing algorithmic scaling behavior $\mathcal{O}(N)$ and $\mathcal{O}(p)$ against memory constraints during backpropagation or matrix inversion, while reconciling discrepancies between offline training capacity and low-latency inference requirements (Bottou, Curtis, & Nocedal, 2018; Chen & Guestrin, 2016; Ke et al., 2017) .

Interpretability and Domain Adherence: Navigating the intrinsic trade-off between self-explanatory models (e.g., generalized additive models, sparse linear frameworks) and high-capacity universal approximators, integrating post-hoc explainability protocols such as SHapley Additive exPlanations (SHAP) and Partial Dependence Plots (PDP) (Lundberg et al.; Molnar et al., 2023; TreeSHAP; Wilson, 2025) and ensuring strict adherence to underlying physical or mathematical principles .

Accordingly, principled machine learning model selection necessitates the construction of an end-to-end data-centric taxonomy and decision-theoretic framework (Du et al., 2025; Vanschoren, 2017). We can systematically reconcile prescriptive modeling paradigms with empirical problem settings by formalizing the mathematical relationships between raw data modalities, statistical constraints and algorithmic inductive biases (Goldblum et al., 2024; Wilson, 2025) . Instead of using ad hoc empirical benchmarks, algorithmic selection should be performed on a common methodological decision framework that allows practitioners to characterize dataset geometry and rationally decide on the optimal model architecture, validation protocol and explainability pipeline (Friedman, 2001; Li et al., 2026; Lundberg et al.) .

2. Taxonomy and Mathematical Characterization of Data Modalities

The performance and stability of machine learning algorithms are conditioned by the algebraic and statistical properties of the input representation space. Prior to formulating the optimization objective or selecting an algorithmic architecture, the input domain must be classified according to its mathematical topology, degree of continuity, dependency structure and dimensionality constraints (Coecke, Sadrzadeh, & Clark, 2010; Hong, 2022; Huang et al., 2025) .

2.1. Tabular and Heterogeneous Data Matrices

Tabular data represents the most ubiquitous modality in experimental and observational sciences, typically formalized as an input design matrix $\mathbf{X} \in \mathbb{R}^{N \times p}$, accompanied by a target response vector $\mathbf{y} \in \mathcal{Y}^N$, where N denotes the number

of observations and p denotes the cardinality of the feature space.(Borisov et al., 2024; Grinsztajn, Oyallon, & Varoquaux; Hastie, Tibshirani, & Friedman, 2009)

Unlike continuous signals or spatial grids, tabular matrices exhibit critical intrinsic challenges:

Heterogeneous Measurement Scales and Semantic Discontinuity: The feature space of tabular data comprises various statistical variable types, such as continuous numerical measurements (R), integer counts (N), ranked ordinal scales and unordered categories (C) (Borisov et al., 2024) . Unlike image or audio signals, adjacent entries in a tabular matrix do not indicate any inherent physical organisation or temporal sequence (Grinsztajn et al.) . Consequently, rearranging the order of rows or columns leaves the true underlying relationships completely unchanged. This means that continuous geometric proximity cannot be reliably assumed across tabular coordinates (Borisov et al., 2024; Hastie et al., 2009) .

Complex Nonlinear Interactions without Local Smoothness: Tabular feature manifolds often exhibit sharp local decision boundaries and interaction terms of varying orders (e.g., $x_j \times x_k$). Continuous neural networks with smooth activation functions (such as Gaussian error linear units or Rectified Linear Units) frequently underperform on unnormalized tabular structures due to rotational invariance issues, whereas axis-aligned recursive partitioning methods (e.g., decision trees) naturally isolate discrete hypercubes in the feature domain (Grinsztajn et al.) .

2.2. Sequential and Temporal Dynamics

When the assumption of independent and identically distributed observations is violated by temporal or sequence-dependent generation, the dataset is formalized as a discrete-time sequence in Equation (2.1):

$$\mathcal{S} = \{(\mathbf{x}_t, y_t)\}_{t=1}^T, \quad \mathbf{x}_t \in \mathbb{R}^p \quad (2.1)$$

As formulated in Equation (2.2), the conditional probability of the state at time t depends strictly on the preceding historical states:

$$P(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \dots, \mathbf{x}_1) \quad (2.2)$$

The mathematical characteristics governing sequential modalities include:

Autoregressive Dependencies and Stationarity: Real-world time series and process data frequently display stochastic trends, cyclical patterns, and non-constant variance (heteroscedasticity). Establishing weak stationarity—constant mean $\mathbb{E}[\mathbf{x}_t] = \mu$ and autocovariance $\gamma(t, t - \tau) = \gamma(\tau)$ independent of time t —is a prerequisite for classical linear formulations (e.g., ARIMA), whereas

nonlinear dynamical architectures (e.g., gated recurrent units, temporal convolutional networks) learn non-stationary transition operators directly (Box, Jenkins, Reinsel, & Ljung, 2015; Cho, Bahdanau, Bougares, Schwenk, & Bengio, 2014; Hamilton, 1994) .

Varying Memory Horizons: Dependencies may be short-range (Markovian transitions) or long-range (power-law decay in autocorrelation functions). The primary modeling challenge involves circumventing exponential vanishing or exploding gradients across extended time steps T during optimization (Bengio, Simard, Frasconi, & Member, 1994; Cho et al., 2014) .

Temporal Leakage in Validation Schemes: Because observations preserve a strict chronological arrow of time, standard uniform cross-validation induces severe forward-looking information leakage. Sequential data dictates strictly causal validation strategies, such as rolling-origin splits or forward-chaining backtesting (Bergmeir & Benítez, 2012; Hamilton, 1994; Sasse et al., 2025) .

2.3. Dimensionality Regimes and the $p \gg N$ Paradigm

The ratio between the number of available training observations N and the dimensionality of the feature space p fundamentally dictates the geometric conditioning of empirical risk minimization and governs algorithmic generalization bounds. In canonical large-sample regimes ($N \gg p$), data points densely populate the underlying continuous manifold, satisfying standard asymptotic consistency conditions and enabling high-capacity learners—such as deep neural networks and deep gradient-boosted ensembles—to resolve intricate, nonlinear decision boundaries without collapsing into empirical variance (Hastie et al., 2009; Lavie et al., 2026). Conversely, when the parameter space far exceeds the sample size, the learning dynamics enter a severely unwell-posed optimisation state, which is characterised by extreme geometric sparsity and the curse of dimensionality (Beyer, Goldstein, Ramakrishnan, & Shaft, 1997). High-dimensional representations can produce rank-deficient or computationally burdensome matrix structures, motivating low-rank approximations and dimensionality-reduction strategies (Li et al., 2026). Consequently, tractable learning in this regime often benefits from models endowed with explicit structural priors—such as parameter shrinkage via regularization, sparse subspace projections, or margin-maximizing kernels—to constrain the effective hypothesis space (Katırcı, ÇELİK, & Oğuz, 2025; Kukačka et al., 2017; Ng, 2004; Xu et al., 2010) .

2.4. Data Incompleteness and Missingness Regimes

From an algorithmic selection perspective, these mechanisms impose strict operational constraints. While certain tree-based ensembles natively manage missing states via default split routing in tabular matrices, continuous architectures across temporal, spatial, and tabular domains require rigorous multivariate imputation protocols—such as low-rank matrix completion, iterative chained equations, or spatio-temporal interpolation—to avoid catastrophic

estimation bias and maintain geometric consistency prior to model fitting (Grinsztajn et al.; Roderick, Little, & Rubin, 2020; Sasse et al., 2025) .

Missing Completely at Random (MCAR): The probability of missingness is strictly independent of both observed and unobserved data coordinates (e.g., stochastic sensor transmission loss or purely random dropouts). Under MCAR, complete-case analysis remains unbiased, though statistical efficiency is substantially diminished.

Missing at Random (MAR): Missingness depends systematically on observable covariates but is conditionally independent of the unobserved missing values themselves. Valid inference under MAR necessitates conditioning on all predictive covariates.

Missing Not at Random (MNAR): Missingness is non-ignorable and directly coupled with the latent value itself or system failure states (e.g., extreme values falling outside instrument detection thresholds or equipment failure under critical stress). In such regimes, the absence of data itself constitutes a predictive signal, frequently introducing structural null states or degenerate matrix representations.

3. Model Selection Criteria and Machine Learning Models

Algorithmic model selection fundamentally hinges on reconciling the geometric and statistical properties of an empirical dataset with the inductive biases inherent in candidate hypothesis spaces. An inductive bias represents the foundational set of mathematical assumptions an algorithm leverages to prioritize certain predictive functions over others in the absence of complete empirical data (Goldblum et al., 2024; Wilson, 2025). Deploying high-capacity learning machines without validating whether their operational mechanics align with the data-generating mechanism results in severe empirical degradation, numerical instability, or catastrophic overfitting (Grinsztajn et al.; Suh & Cheng, 2024) . To establish a rigorous decision framework, this section categorizes the major machine learning paradigms across five critical structural regimes, delineating their mathematical formulations, geometric properties, governing constraints, and precise selection boundaries (Hastie et al., 2009; Vanschoren, 2017) .

3.1. Sample-to-Feature Geometry and Regularized Formulations

The ratio between the number of training observations and covariate dimensionality dictates the numerical conditioning of parameter estimation and the stability of optimization trajectories. In canonical linear regression, standard ordinary least squares relies on an unconstrained squared error loss. However, when the feature space approaches or exceeds the sample size, or when severe multicollinearity is present, the empirical Gram matrix becomes singular and rank-deficient. Under these conditions, estimator variance diverges toward infinity, producing severe coefficient oscillation, non-unique interpolating solutions, and catastrophic generalization error on unseen test distributions (Du et al., 2025; Hastie et al., 2009) .

To stabilize this underdetermined optimization space, structural regularization introduces explicit inductive biases by constraining parameter magnitudes (Wilson, 2025). Ridge regression incorporates a continuous quadratic penalty on coefficient magnitudes, which bounds the condition number of the system, stabilizes matrix inversion, and suppresses estimation variance while retaining all predictors. Conversely, the Least Absolute Shrinkage and Selection Operator (Lasso) imposes an absolute magnitude penalty, generating sharp geometric corners that drive non-informative coefficients exactly to zero, thereby executing simultaneous shrinkage and sparse variable selection (Li et al., 2026). When high-dimensional datasets exhibit groups of strongly correlated predictors, the Elastic Net formulation linearly synthesizes both penalty terms, overcoming the rank saturation limits of Lasso and ensuring group-wise coefficient stability across ill-posed empirical regimes (Katırcı et al., 2025; Zou & Hastie, 2005).

In small-to-moderate sample settings ($N < 5,000$), Support Vector Machines (SVM) and Support Vector Regression (SVR) model complex, non-linear relationships by finding an optimal decision margin. Instead of manual feature engineering, these methods project data into higher dimensions using kernel functions (such as radial basis functions) to separate patterns cleanly. By ignoring errors below a specified tolerance threshold, the model focuses only on key data points along the boundary (the support vectors). Because this optimization guarantees a single best solution and resists outlier noise, kernel SVM/SVR provides a dependable choice for non-linear tasks where sample size is too limited for deep neural networks (Dedieu, Mazumder, & Wang, 2022; Drucker, Surges, Kaufman, Smola, & Vapnik, 1997; Hawkins, 2012).

3.2. Tabular Complexity, Nonlinear Decision Boundaries and Tree Ensembles

When the underlying response surface exhibits severe non-convexities, step-like local discontinuities, or high-order predictor interactions, linear formulations suffer from substantial structural bias. In structured tabular domains—where continuous coordinates lack spatial continuity or temporal ordering—decision-tree ensembles provide an optimal inductive bias through axis-aligned recursive partitioning. By recursively segmenting the feature space into orthogonal hypercubes, tree architectures capture complex non-linear thresholds without requiring numerical feature normalization, monotonic transformations, or explicit interaction modeling (Borisov et al., 2024; Ganaie, Hu, Malik, Tanveer, & Suganthan, 2022; Grinsztajn et al.; Mohammed & Kora, 2023b).

The algorithmic selection of tree ensembles bifurcates into two distinct learning paradigms based on dataset cardinality and noise characteristics:

Bagging Ensembles (Random Forest): Recommended for moderate sample sizes contaminated by heavy outlier noise or measurement variance. By constructing deep, low-bias decision trees over bootstrap samples and decorrelating them via random feature subspace selection, Random Forests systematically suppress estimator variance. Because individual trees train independently, the architecture demonstrates remarkable resistance to overfitting

and insensitivity to extreme covariate values, serving as a reliable default baseline across unnormalized tabular matrices (Breiman, 1999; Lujan-Moreno, Howard, Rojas, & Montgomery, 2018) .

Gradient Boosted Decision Trees (GBDT: XGBoost, LightGBM, CatBoost): Indispensable for large-scale tabular regimes requiring maximum predictive accuracy across complex interaction manifolds. Gradient boosting builds shallow, high-bias trees additively to minimize arbitrary differentiable loss functions via stage-wise optimization. Specific algorithmic variants resolve distinct operational bottlenecks:

XGBoost is an efficient gradient-boosting framework that constructs ensembles of decision trees through regularized objective optimization, incorporating second-order gradient information and tree-complexity control (Chen & Guestrin, 2016; Guang-Yu & Geng, 2019) .

LightGBM minimizes memory footprints and computational overhead across massive sample sizes through histogram-based feature binning and gradient-based one-side sampling (Corporation, 2026; Ke et al., 2017) .

CatBoost eliminates prediction shift and target leakage in datasets with high-cardinality nominal predictors through ordered target statistics and symmetric oblivious trees. Furthermore, gradient boosted ensembles natively accommodate missing feature coordinates via default split routing, bypassing the estimation bias frequently induced by external imputation protocols (Breiman, 1999; Lujan-moreno, Howard, Rojas, & Montgomery, 2018; Prokhorenkova, Gusev, Vorobev, Dorogush, & Gulin, 2018) .

3.3. Sequential, Temporal and Autoregressive Architectures

Standard static estimators suffer from structural failure when empirical observations violate the assumption of independent and identically distributed variables through sequential, temporal, or process-dependent generation (Hamilton, 1994) . Sequential data modalities require models that can capture directional dependencies, temporal autocorrelation and varying memory horizons without causing informational leakage (Du et al., 2025; Sasse et al., 2025). To reconcile these operational constraints, sequential learning architectures evolve from basic recurrent formulations to more advanced gated and non-recurrent temporal approaches (Bai, Kolter, & Koltun, 2018; Hochreiter, 2016) .

Recurrent Neural Networks (Standard RNNs):

Standard recurrent networks introduce feedback loops where the current hidden state is continuously conditioned on both the immediate input vector and the immediate historical hidden state. This design creates a cyclical parameter-sharing mechanism suitable for short, low-order autoregressive dependencies (Hamilton, 1994) . However, their practical hypothesis space is strictly bounded: as the sequence length expands, gradient signals backpropagated through time decay or explode exponentially. Consequently, basic RNNs exhibit severe memory decay, rendering them incapable of modeling long-range temporal

horizons or delayed response kinetics (Bengio et al., 1994; Çelik, Katırcı, & İşlek, 2024; Hochreiter, 2016) .

Gated Recurrent Architectures (LSTM and GRU):

To bypass the vanishing and exploding gradient dilemma of basic RNNs, gated recurrent mechanisms introduce regulated linear channels and continuous memory storage:

Long Short-Term Memory (LSTM): Regulates information flow across extended time horizons through three distinct gating structures: a forget gate that discards redundant historical states, an input gate that incorporates novel sequential perturbations, and an output gate that modulates latent activations. This explicit cell state preservation ensures stable gradient flow across prolonged experimental trajectories, dynamic sensor streams, and longitudinal degradation curves (CAN, 2019; Hochreiter, 2016) .

Gated Recurrent Units (GRU): Condenses the LSTM mechanics by coupling the cell state and hidden state into a single representation and merging the gating dynamics into reset and update gates. By reducing structural parameter volume, GRUs mitigate overfitting risks across moderate sample sizes and provide faster optimization convergence with negligible loss in modeling capacity (Chung; İşlek, Katırcı, & Celik, 2024) .

Temporal Convolutional Networks (TCNs)

As sequence length increases substantially, recurrent architectures encounter sequential execution bottlenecks that prohibit layer-wise parallelization (Du et al., 2025) . Temporal Convolutional Networks circumvent this operational limit by deploying 1D dilated causal convolutions (Bai et al., 2018) . Causal masking strictly preserves the chronological arrow of time to prevent forward-looking information leakage, while exponential dilation factors expand the receptive field without an explosive growth in parameter density, outperforming classical recurrent models on complex, long-range univariate and multivariate time series (Bai et al., 2018; Sasse et al., 2025) .

Sequential Attention and Temporal Foundation Models

When multivariate temporal spaces exhibit intricate inter-variable dependencies alongside variable-length delays, attention-based sequence models displace recurrent routing altogether (Du et al., 2025) . By computing pairwise dot-product affinities across the entire temporal window, attention mechanisms dynamically assign focus to critical transition thresholds, overcoming the sequential step-by-step memory bottleneck and excelling in high-dimensional, long-context forecasting regimes (Vaswani et al., 2017) .

3.4. Deep Representation Learning and Large-Scale Regimes

Deep neural networks approximate highly non-linear operators through hierarchical compositions of parameterized affine transformations and non-linear activation functions. While the Universal Approximation Theorem guarantees

that a single hidden layer with sufficient width can approximate any continuous function on compact subsets of Euclidean space, deep architectures leverage depth to achieve exponential expressiveness with polynomially fewer parameters (Du et al., 2025) . Despite their dominance in unstructured vision, audio, and language domains, standard Multi-Layer Perceptrons (MLPs) and naive transformer architectures frequently underperform tree-based ensembles on structured tabular data. This empirical degradation stems primarily from rotational invariance: rotating the coordinate system of the input matrix does not alter the hypothesis class or the optimization trajectory of fully connected networks. However, tabular features represent distinct physical semantics (e.g., temperature, pressure, concentration), where arbitrary coordinate rotations destroy the coordinate-aligned inductive structures that tree ensembles naturally preserve (Borisov et al., 2024; Grinsztajn et al.).

Consequently, deep neural architectures become statistically viable and methodologically superior under specific empirical configurations:

Extensive Sample Volume ($N > 10^5$): Large-scale datasets provide the dense manifold coverage necessary for stochastic gradient descent to converge without succumbing to sample memorization or variance collapse (Wilson, 2025).

Multi-Modal Data Integration: Predictive settings requiring joint, end-to-end representation learning across heterogeneous sources—such as combining numerical metadata, temporal sensor streams, and spatial image tensors—rely on modular deep architectures (Borisov et al., 2024) .

Self-Supervised Pre-Training and Transfer Learning: Settings where large uncured corpora can be leveraged to pre-train expressive latent representations that are subsequently fine-tuned on sample-scarce downstream tasks (Lan et al., 2020) .

3.5. Variational Autoencoders and Generative Adversarial Networks

Beyond discriminative mapping, deep architectures excel in continuous density estimation and manifold sampling through generative paradigms. Variational Autoencoders (VAEs) map high-dimensional covariates into smooth, probabilistic latent spaces via variational inference, enforcing continuity through Kullback-Leibler divergence regularizers to enable structured interpolation and property-guided inverse optimization (Kingma & Welling, 2019, 2022) . Concurrently, Generative Adversarial Networks (GANs) formulate synthesis as a minimax game between generator and discriminator networks, capturing complex, high-order empirical distributions without explicit likelihood tractability (Eckerli & Osterrieder, 2021; Goodfellow et al.) . In experimental and engineering disciplines, these generative frameworks provide critical utility for conditional synthetic data augmentation, candidate screening, and inverse material/process design under resource constraints.

3.6. Domain-Constrained, Physics-Informed, and Symbolic Syntheses

In scientific, chemical, and physical engineering applications, experimental data generation is frequently constrained by prohibitive operational costs, yielding extreme sample scarcity ($N < 100$). Fitting unconstrained, high-capacity black-box models in this data-sparse regime leads to severe empirical failure, unstable extrapolation, and physically implausible predictions—such as negative concentrations, non-monotonic rate laws, or violations of conservation principles (Du et al., 2025; Wilson, 2025). To enforce mechanistic consistency, domain knowledge must be integrated directly into the learning objective through specialized paradigms:

Physics-Informed Neural Networks (PINNs): PINNs resolve severe sample scarcity by embedding known governing differential equations directly into the empirical risk optimization objective via automatic differentiation. By penalizing the residual norms of the governing equations across domain collocation points, PINNs ensure that model trajectories satisfy underlying conservation laws and boundary conditions, even when trained on highly sparse observational measurements (Raissi, Perdikaris, & Karniadakis, 2018).

Symbolic Regression and Gene Expression Programming (GEP): When domain practitioners require transparent, closed-form algebraic formulations rather than opaque parameterizations, symbolic learning applies evolutionary optimization to discover parsimonious mathematical equations. By constructing and mutating expression trees over primitive arithmetic operators, GEP yields fully interpretable, compact analytical laws that combine predictive accuracy with explicit scientific explainability (Ferreira, 2021).

4. Model Validation Strategies, Hyperparameter Optimization, and Explainable AI (XAI)

Model selection must be coupled with rigorous validation schemes, systematic hyperparameter optimization, and transparent post-hoc interpretability protocols to ensure generalization and guard against empirical artifacts (Hastie et al., 2009).

4.1. Validation Protocols and Mitigation of Data Leakage

Data leakage—the inadvertent incorporation of out-of-sample information into the model training pipeline—artificially inflates performance metrics and leads to generalization failures on unseen data (Sasse et al., 2025). Validation protocols must strictly mirror the underlying data topology:

***k*-Fold Cross-Validation:**

For independent and identically distributed (i.i.d.) tabular data, K-Fold cross-validation partitions the dataset into K mutually exclusive subsets, or folds, and iteratively uses each fold as a hold-out validation set while training the model on the remaining folds. When the target variable is class-imbalanced, Stratified K-Fold cross-validation preserves the empirical class proportions across folds,

thereby maintaining a more representative class distribution during model validation (Hastie et al., 2009) .

Rolling-Origin and Causal Temporal Splits:

Conventional randomised data splitting can violate temporal causality for dynamic time series, sensor streams and degradation trajectories by allowing information from future observations to influence model training, thereby introducing forward-looking leakage. Forward-chaining validation addresses this issue by ensuring that each training set comprises observations from historical intervals preceding the corresponding evaluation window. This preserves the temporal ordering between the training and evaluation data (Bergmeir & Benítez, 2012).

Encapsulation of Data Preprocessing:

Feature transformations—including scaling, mean/median imputation, and dimensionality reduction—must be computed exclusively on the active training split and then mapped onto the validation fold. Fitting transformation parameters globally across the entire corpus causes direct information leakage (Hastie et al., 2009; Sasse et al., 2025) .

4.2. Hyperparameter Optimization Frameworks

Exhaustive grid searches and unguided random searches scale poorly across high-dimensional parameter spaces. Probabilistic sequential optimization mitigates this computational burden:

Bayesian Optimization and Gaussian Process Surrogates:

Bayesian Optimization constructs a probabilistic surrogate model of the objective function to map hyperparameters to expected cross-validation performance and employs acquisition functions, such as Expected Improvement, to balance exploration of uncertain regions with exploitation of known high-performing parameter subspaces(Snoek, Larochelle, & Adams, 2012).

Modern Adaptive Frameworks:

Within automated hyperparameter optimization, Optuna combines Tree-structured Parzen Estimators (TPE) with automated pruning mechanisms, such as the Successive Halving Algorithm, to efficiently explore promising configurations while terminating underperforming trials at an early stage (Watanabe & Irie, 2026). For larger-scale optimization tasks, Keras Tuner and distributed frameworks extend this process through scalable Hyperband-based scheduling and neural architecture search, enabling efficient exploration of deep neural network architectures (Liao, Li, Shang, & Ma, 2022).

4.3. Model Interpretability and Explainable AI (XAI)

Non-linear ensemble methods and deep neural networks can achieve strong predictive performance, but their internal decision-making processes often remain opaque and difficult to interpret (Du et al., 2025) . This lack of

transparency creates a need for post-hoc, model-agnostic explainability methods that can provide mechanistic insight into model behavior and assess whether the learned relationships are consistent with underlying physical and empirical principles (Sasse et al., 2025) .

Shapley Additive exPlanations (SHAP):

SHAP is grounded in cooperative game theory and quantifies the marginal additive contribution of each predictor to individual model predictions. By decomposing predictions into feature-level contributions, SHAP provides both global feature-importance rankings and local instance-level attribution profiles while satisfying key properties such as local accuracy, missingness, and consistency (Lundberg et al.; TreeSHAP) .

Partial Dependence Plots (PDP) and Individual Conditional Expectation (ICE):

Partial dependence plots (PDPs) are used to interpret global models. They estimate the marginal average effect of one or two selected features on model predictions across the population, providing an overall view of the relationship between feature values and predicted outcomes (Beyer et al., 1997; Friedman, 2001) . However, averaging across observations can mask significant differences in individual responses. Individual Conditional Expectation (ICE) overcomes this issue by breaking down the population-level relationship into observation-specific response curves. This allows for the identification of heterogeneous feature effects and potential interaction patterns at the individual level (Goldstein, Kapelner, Bleich, & Pitkin, 2015).

Spatial Attribution and Tensor Visualizations:

In regard to convolutional neural networks (CNNs), the Grad-CAM method highlights decisive pixel regions and spatial boundary activations directly within the input tensors (Selvaraju et al., 2017) .

5. Summary and Model Selection Framework

Navigating the landscape of statistical learning requires reconciling the mathematical assumptions of candidate algorithms with the empirical geometry and constraints of the data domain. Rather than relying on unguided heuristics or assuming the universal superiority of high-capacity models, practitioners must adopt an integrated decision pipeline that links data modality, sample volume, parameter ratios, and domain knowledge to inductive biases. The following framework synthesizes these principles into an actionable decision taxonomy and a step-by-step procedural workflow, establishing a systematic pathway for robust and generalizable model deployment.

In order to guide practitioners in identifying optimal model-data pairings, Table 1 provides a systematised classification that maps empirical dataset properties, dimensionality regimes, and task constraints to appropriate algorithmic families and their associated validation protocols. Aligning the

geometric characteristics of the feature space with the correct inductive biases can reduce the risk of overfitting and significant empirical decline.

Table 1. Decision Framework for machine learning model selection across data modalities, dimensionality regimes, and constraints.

Data Modality	Sample Volume (N)	Dimensionality (p)	Governing Constraint Challenge	Recommended Algorithmic Family	Validation & XAI Protocols
Tabular	Small ($N < 500$)	High ($p \gg N$, Spectral)	Multicollinearity, severe overfitting risk	Elastic Net, Lasso, Ridge	k -Fold CV, penalty trace (ℓ_1/ℓ_2) diagnostics
Tabular	Small to Moderate ($N < 5,000$)	Low to Moderate	Non-linear margin separation	SVR/SVM	Bayesian optimization, SHAP
Tabular	Moderate ($10^2 < N < 10^5$)	Moderate to High	Noisy feature space, high outlier density	Random Forest	Out-of-bag (OOB) error estimation, SHAP, PDP
Tabular	Large ($N > 10^3$)	High categorical cardinality	High-accuracy non-linear mapping	CatBoost, LightGBM, XGBoost	Stratified k -Fold, SHAP, ICE
Sequential	Dynamic / Longitudinal	Ordered sequence	Autocorrelation, vanishing temporal gradients	LSTM, GRU	Rolling-origin / causal temporal split
Spatial / Grid	Moderate to Large ($N > 1,000$)	2D/3D tensor pixels	Structural translation equivariance	CNN	Tensor augmentations, Grad-CAM, feature maps
Generative / Latent	Moderate to Large ($N > 1,000$)	High / Multi-modal	Inverse design, distribution matching, synthetic data augmentation	VAEs, GANs	Latent space interpolation, reconstruction loss, Fréchet Inception Distance (FID)
Physical / Kinetic	Sparse ($N < 100$)	Differential equations	Governing conservation and physical laws	PINNs	Residual PDE loss monitoring, $\mathcal{L}_2$ error norms
Symbolic Target	Small ($N < 500$)	Low to Moderate	Closed-form analytical formula requirement	GEP Regression	Parsimony validation, expression tree analysis

A higher parameter count does not ensure superior out-of-sample performance. Lasting predictive reliability is achieved when algorithmic inductive bias closely corresponds to the underlying data structure. By evaluating

sample scale, feature geometry, and structural constraints together, researchers minimize the risk of empirical failure while building interpretable, reproducible machine learning workflows.

6. Future Directions and Open Challenges

As data-driven modeling paradigms continue to mature, emerging methodologies introduce both novel opportunities and rigorous technical challenges to systematic model selection:

Tabular Foundation Models and In-Context Learning: The success of large-scale pre-trained transformer architectures in natural language processing and computer vision has recently extended to tabular domains. Emerging tabular foundation models (e.g., TabPFN, TabNet) exhibit remarkable capabilities in zero-shot and few-shot prediction across sample-scarce regimes. Nevertheless, determining whether these pre-trained representations can consistently surpass highly optimized gradient boosted decision trees (GBDTs) across heterogeneous, high-cardinality, and heavy-tailed industrial datasets remains an open research question.

Limitations of Automated Machine Learning (AutoML) vs. Domain Expertise: While AutoML pipelines dramatically streamline hyperparameter search and baseline algorithm screening, they frequently operate as unconstrained empirical optimizers. Consequently, they remain vulnerable to latent data leakage, the propagation of spurious collinearities, and the failure to enforce mandatory physical or mathematical invariants. Advancing beyond naive search heuristics necessitates the development of constraint-aware AutoML frameworks that natively incorporate domain-specific priors into the objective function.

Optimization and Scalability in Physics-Informed Architectures: Physics-Informed Neural Networks (PINNs) provide a robust inductive bias for sample-scarce engineering problems governed by differential equations. However, scaling PINNs to stiff, chaotic, or multiscale dynamical systems presents substantial optimization bottlenecks. Resolving gradient pathologies, balancing multi-objective loss landscapes, and handling non-convex boundary value problems without prohibitive computational latency remain key computational frontiers.

Causal Robustness in Post-Hoc Explainability (XAI): Established post-hoc attribution frameworks—principally Shapley Additive exPlanations (SHAP) and Partial Dependence Plots (PDP)—rely heavily on observational marginal distributions. Under high multicollinearity, these methods may evaluate permutations in physically impermissible feature spaces, producing misleading attribution metrics. Bridging model-agnostic explainability with formal causal inference and structural equation modeling is imperative to ensure that explanatory outputs reflect genuine mechanistic dependencies rather than statistical artifacts.

References

- Bai, S., Kolter, J. Z., & Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. <https://doi.org/10.48550/arXiv.1803.01271>
- Bengio, Y., Simard, P., Frasconi, P., & Member, S. (1994). Learning Long-Term Dependencies with Gradient Descent is Difficult, 5(2). <https://doi.org/10.1109/72.279181>
- Bergmeir, C., & Benítez, J. M. (2012). On the Use of Cross-validation for Time Series Predictor Evaluation. *Information Sciences*. <https://doi.org/10.1016/j.ins.2011.12.028>
- Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1997). When Is “Nearest Neighbor” Meaningful?
- Borisov, V., Leemann, T., Sebler, K., Haug, J., Pawelczyk, M., & Kasneci, G. (2024). Deep Neural Networks and Tabular Data: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6), 7499–7519. <https://doi.org/10.1109/TNNLS.2022.3229161>
- Bottou, L., Curtis, F. E., & Nocedal, J. (2018). Optimization Methods for Large-Scale Machine Learning. <https://doi.org/10.48550/arXiv.1606.04838>
- Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time Series Analysis: Forecasting and Control. *Journal of Time Series Analysis*, 37(5), 709–711. <https://doi.org/10.1111/jtsa.12194>
- Breiman, L. (1999). Random Forests - Random Features, 1–29. <https://doi.org/10.1023/A:1010950718922>
- CAN, B. (2019). LSTM Ağları ile Türkçe Kök Bulma. *Bilişim Teknolojileri Dergisi*, 183–193. <https://doi.org/10.17671/gazibtd.486042>
- Çelik, H., Katırcı, R., & İşlek, B. (2024). Effect Of Parameters On Performance In Question-Answer Model With Simple Rnn Deep Learning Method. <https://doi.org/10.5281/zenodo.11320381>
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 13-17-Aug*, 785–794. <https://doi.org/10.1145/2939672.2939785>
- Cho, K., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *Journal of Clinical Microbiology*, 28(4), 828–829. <https://doi.org/10.1128/jcm.28.4.828-829.1990>
- Chung, J. Gated Recurrent Neural Networks on Sequence Modeling, 1–9.

- Coecke, B., Sadrzadeh, M., & Clark, S. (2010). Mathematical Foundations for a Compositional Distributional Model of Meaning. <https://doi.org/10.48550/arXiv.1003.4394>
- Corporation, M. (2026). *Welcome to LightGBM's documentation*. Retrieved 09/12/2026 from <https://lightgbm.readthedocs.io/en/stable/>
- Dedieu, A., Mazumder, R., & Wang, H. (2022). Solving L1-regularized SVMs and Related Linear Programs: Revisiting the Effectiveness of Column and Constraint Generation. *Journal of Machine Learning Research*, 23, 1–41.
- Drucker, H., Surges, C. J. C., Kaufman, L., Smola, A., & Vapnik, V. (1997). Support vector regression machines. *Advances in Neural Information Processing Systems*, 779–784.
- Du, K. L., Zhang, R., Jiang, B., Zeng, J., & Lu, J. (2025, February 1). Understanding Machine Learning Principles: Learning, Inference, Generalization, and Computational Learning Theory. *Mathematics*. Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/math13030451>
- Eckerli, F., & Osterrieder, J. (2021). Generative Adversarial Networks In Finance An Overview, 825215(825215).
- Ferreira, C. (2021). Gene Expression Programming: a New Adaptive Algorithm for Solving Problems. <https://doi.org/10.48550/arXiv.cs/0102027>
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232. <https://doi.org/10.1214/aos/1013203451>
- Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., & Suganthan, P. N. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*. <https://doi.org/10.1016/j.engappai.2022.105151>
- Goldblum, M., Finzi, M., Rowan, K., & Wilson, A. G. (2024). The No Free Lunch Theorem, Kolmogorov Complexity, and the Role of Inductive Biases in Machine Learning. <https://doi.org/10.48550/arXiv.2304.05366>
- Goldstein, A., Kapelner, A., Bleich, J., & Pitkin, E. (2015). Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics*, 24(1), 44–65. <https://doi.org/10.1080/10618600.2014.907095>
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. *Generative Adversarial Nets*. <https://doi.org/10.48550/arXiv.1406.2661>
- Grinsztajn, L., Oyallon, E., & Varoquaux, G. *Why do tree-based models still outperform deep learning on typical tabular data?* <https://doi.org/10.52202/068431-0037>

- Guang-Yu, L., & Geng, H. (2019). The behavior analysis and achievement prediction research of college students based on XGBoost gradient lifting decision tree algorithm. In *ACM International Conference Proceeding Series* (Vol. Part F148391, pp. 289–294). Association for Computing Machinery. <https://doi.org/10.1145/3323771.3323803>
- Hamilton, J. D. (1994). *Time Series Analysis*. <https://doi.org/10.2307/j.ctv14jx6sm>
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning Data Mining, Inference, and Prediction*.
- Hawkins, J. (2012). *Support Vector Regression*.
- He, Haibo, & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263–1284. <https://doi.org/10.1109/TKDE.2008.239>
- He, Haiyun, & Goldfeld, Z. (2025). Information-Theoretic Generalization Bounds for Deep Neural Networks. Retrieved from <http://arxiv.org/abs/2404.03176>
- Hochreiter, S. (2016). Long Short-Term Memory, (November 1997). <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hollmann, N., Müller, S., Eggensperger, K., & Hutter, F. *TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second*. <https://doi.org/10.48550/arXiv.2207.01848>
- Hong, I. K. (2022). Smooth Mathematical Function from Compact Neural Networks, 1–22. <https://doi.org/10.48550/arXiv.2301.00181>
- Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., ... Liu, T. (2025). A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Transactions on Information Systems*, 43(2). <https://doi.org/10.1145/3703155>
- İşlek, B., Katırcı, R., & Celik, H. (2024). Enhancing Question Answering Systems Through Optimal Hyperparameter Tuning in GRU. *8th International Artificial Intelligence and Data Processing Symposium, IDAP 2024*, (September). <https://doi.org/10.1109/IDAP64064.2024.10710732>
- Katırcı, R., ÇELİK, H., & Oğuz, T. (2025). conversational ai and question answering systems: architectures, training challenges and solutions. In A. ŞEKER & A. G. YÜKSEK (Eds.), *Artificial Intelligence Future Frontiers of Generative AI in Engineering* (pp. 47–68). <https://doi.org/10.5281/zenodo.17449635>
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems, 2017-Decem(Nips)*, 3147–3155.
- Kingma, D. P., & Welling, M. (2019). An Introduction to Variational Autoencoders. <https://doi.org/10.1561/22000000056>

- Kingma, D. P., & Welling, M. (2022). Auto-Encoding Variational Bayes. <https://doi.org/10.48550/arXiv.1312.6114>
- Kukačka, J., Golkov, V., & Cremers, D. (2017). Regularization for Deep Learning: A Taxonomy, 1–23. <https://doi.org/10.48550/arXiv.1710.10686>
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). Albert: a Lite Bert for Self-Supervised Learning of Language Representations. *8th International Conference on Learning Representations, ICLR 2020*, 1–17.
- Lavie, I., Levi, N., & Kahn, Y. (2026). Statistical Properties of Training & Generalization. <https://doi.org/10.48550/arXiv.2606.20299>
- Li, Z., Wu, X. Y., Ling, X., & Tu, C. (2026). Compressing complexity: a critical synthesis of structural, analytical, and data-driven dimensionality reduction in dynamical networks. *Advances in Physics: X*. Taylor and Francis Ltd. <https://doi.org/10.1080/23746149.2026.2645192>
- Liao, L., Li, H., Shang, W., & Ma, L. (2022). An Empirical Study of the Impact of Hyperparameter Tuning and Model Optimization on the Performance Properties of Deep Neural Networks. *ACM Transactions on Software Engineering and Methodology*, 31(3). <https://doi.org/10.1145/3506695>
- Lujan-moreno, G. A., Howard, P. R., Rojas, O. G., & Montgomery, D. C. (2018). Design of experiments and response surface methodology to tune machine learning hyperparameters , with a random forest case-study. *Expert Systems With Applications*, 109, 195–205. <https://doi.org/10.1016/j.eswa.2018.05.024>
- Lujan-Moreno, G. A., Howard, P. R., Rojas, O. G., & Montgomery, D. C. (2018). Design of experiments and response surface methodology to tune machine learning hyperparameters, with a random forest case-study. *Expert Systems with Applications*, 109, 195–205. <https://doi.org/10.1016/j.eswa.2018.05.024>
- Lundberg, S. M., Allen, P. G., & Lee, S.-I. *A Unified Approach to Interpreting Model Predictions*. <https://doi.org/10.48550/arXiv.1705.07874>
- Massaro, A., Maritati, V., Galiano, A., Birardi, V., & Pellicani, L. (2018). ESB Platform Integrating Knime Data Mining Tool Oriented on Industry 4.0 Based on Artificial Neural Network Predictive Maintenance. *International Journal of Artificial Intelligence & Applications*, 9(3), 01–17. <https://doi.org/10.5121/ijia.2018.9301>
- Mohammed, A., & Kora, R. (2023a, February 1). A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University - Computer and Information Sciences*. King Saud bin Abdulaziz University. <https://doi.org/10.1016/j.jksuci.2023.01.014>

- Mohammed, A., & Kora, R. (2023b, February 1). A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University - Computer and Information Sciences*. King Saud bin Abdulaziz University. <https://doi.org/10.1016/j.jksuci.2023.01.014>
- Molnar, C., Freiesleben, T., König, G., Herbinger, J., Reisinger, T., Casalicchio, G., ... Bischl, B. (2023). Relating the Partial Dependence Plot and Permutation Feature Importance to the Data Generating Process. In *Communications in Computer and Information Science* (Vol. 1901 CCIS, pp. 456–479). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-3-031-44064-9_24
- Ng, A. Y. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. *Proceedings of the 21 St International Conference on Machine Learning*. <https://doi.org/10.1145/1015330.101543>
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). Catboost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems, 2018-Decem*(Section 4), 6638–6648.
- Qu, J., Holzmüller, D., Varoquaux, G., & Morvan, M. Le. (2026). TabICLv2: A better, faster, scalable, and open tabular foundation model. <https://doi.org/10.48550/arXiv.2602.11139>
- Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2018). *Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations*. <https://doi.org/10.1016/j.jcp.2018.10.045>
- Roderick, N., Little, J. A., & Rubin, D. B. (2020). *Statistical Analysis with Missing Data*. (D. J. Balding, N. A. C. Cressie, G. M. Fitzmaurice, G. H. Givens, H. Goldstein, G. Molenberghs, ... R. S. Tsay, Eds.) (Vol. 6). <https://doi.org/10.1002/9781119013563>
- Ruben, B. S., Tong, W. L., Chaudhry, H. T., & Pehlevan, C. (2025). No Free Lunch From Random Feature Ensembles: Scaling Laws and Near-Optimality Conditions. <https://doi.org/10.48550/arXiv.2412.05418>
- Sasse, L., Nicolaisen-Sobesky, E., Dukart, J., Eickhoff, S. B., Götz, M., Hamdan, S., ... Patil, K. R. (2025). Overview of leakage scenarios in supervised machine learning. *Journal of Big Data*, 12(1). <https://doi.org/10.1186/s40537-025-01193-8>
- Şeker, Şadi, E. Knime ile uçtan uca veri bilimi.
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *Proceedings of the IEEE International Conference on Computer Vision* (Vol. 2017-October, pp. 618–626). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ICCV.2017.74>

- Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. <https://doi.org/10.48550/arXiv.1206.2944>
- Sroka, G. (2026). Empirical Evaluation of No Free Lunch Violations in Permutation-Based Optimization. <https://doi.org/10.48550/arXiv.2603.03613>
- Suh, N., & Cheng, G. (2024). A Survey on Statistical Theory of Deep Learning: Approximation, Training Dynamics, and Generative Models. <https://doi.org/10.1146/annurev-statistics-040522-013920>
- Tinmaz, H. (2025). From Transparency to Trust: A Literature Review on Explainable AI in Educational Systems. *Mersin Üniversitesi Eğitim Fakültesi Dergisi*, 21(3), 999–1018. <https://doi.org/10.17860/mersinefd.1739347>
- TreeSHAP. *shap.TreeExplainer*. Retrieved 09/07/2026 from <https://shap.readthedocs.io/en/latest/generated/shap.TreeExplainer.html>
- Vanschoren, J. (2017). *Automated Machine Learning. Interdisciplinary Mathematical Sciences* (Vol. 19). https://doi.org/10.1142/9789814630146_0014
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 2017-Decem(Nips)*, 5999–6009. <https://doi.org/10.48550/arXiv.1706.03762>
- Watanabe, S., & Irie, K. (2026). Optuna Constrained Tree-Structured Parzen Estimator Is a Joint Density Generalization of c-TPE. <https://doi.org/10.48550/arXiv.2606.09889>
- Wilson, A. G. (2025). Deep Learning is Not So Mysterious or Different. <https://doi.org/10.48550/arXiv.2503.02113>
- Xu, Z. Ben, Zhang, H., Wang, Y., Chang, X. Y., & Liang, Y. (2010). L1/2 regularization. *Science in China, Series F: Information Sciences*, 53(6), 1159–1169. <https://doi.org/10.1007/s11432-010-0090-0>
- Zou, H., & Hastie, T. (2005). *Regularization and variable selection via the elastic net. J. R. Statist. Soc. B* (Vol. 67). <https://doi.org/10.1111/j.1467-9868.2005.00503.x>