POLITECNICO DI TORINO

Master's Degree Program
in Data Science and Engineering

Thesis on Mask2KAN

Mask2KAN: A Universal Image Segmentation Kolmogorov–Arnold Network Architecture

Supervisors
prof. Barbara Caputo
prof. Carlo Masone
prof. Shyam Nandan Rai
firma dei relatori

. . . .
. . . .

Candidate
Gianluca Guzzetta

firma del candidato

. . . .

Academic Year 2023-2024

A mia mamma Rita e
mio papà Massimo
A mia sorella Ele ed i
miei fratellini Poppi e
Tommy
† Al mio nonno Nino
E a chi mi vuole davvero
bene.

Summary

Universal architectures like Mask2Former have redefined the way we approach image segmentation tasks. Traditionally, specialized architectures were used for specific tasks such as semantic, instance, and panoptic segmentation. Now, a single, unified architecture can outperform these task-specific models, offering benefits in performance, efficiency, and effort, while also reshaping the way we perceive these tasks. In this paper, experiments are conducted using the Mask2Former configuration for *semantic segmentation*. However, similar to other universal models like DETR, these architectures, despite sharing the same underlying structure, *are still trained separately for different tasks and datasets*. Recent works on the passage from the Universal Approximation Theorem to a Kolmogorov-Arnold theorem inspired the present work to delve in Kolmogorov Arnold Network on computer vision tasks. Traditional semantic segmentation models as Mask2Former, recognize a predefined set of classes, often failing to detect unseen objects (anomalies). To address this, we propose Mask2KAN, a novel approach derived from the Mask2Former architecture , which shifts from a per-pixel (i.e. BERT) to a mask classification (i.e. Mask2Former) focusing on reducing ood anomalies (i.e. Mask2Anomaly), with an efficient Kolmogorov-Arnold Network (KAN) mask embed prediction head, hence improving the segmentation of unseen objects and reducing false positives. Proposed architectures include ResNet-50 and Swin-T/S/B/L as backbones. and using KAN mask embed layers sets a new state-of-the-art in anomaly segmentation, since our approach demonstrates superior performance across various benchmarks on semantic segmentation, making it a robust solution also for real-world scenarios as autonomous driving applications or anomaly detection in the wild. For more details and code, visit our Github page.

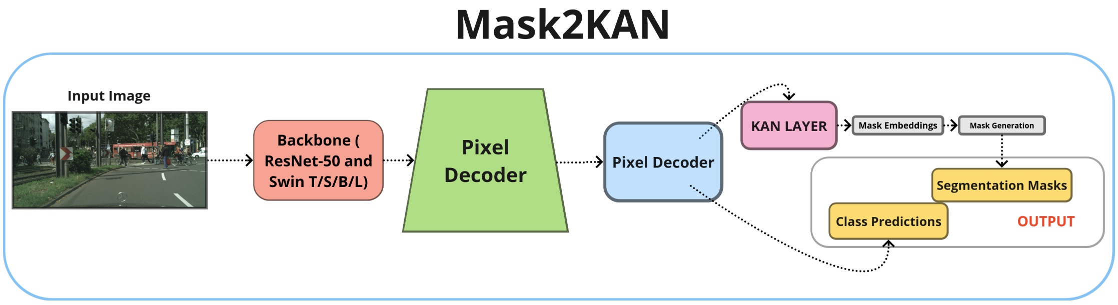


Figure 1. Mask2KAN architecture

Acknowledgements

A special thanks to my family and my cats. Love you all guys! A special thanks also to Barbara Caputo, Carlo Masone and Shyam who guided me through this and shared me love and passion for computer vision.

Contents

List of Tables

List of Figures

The only way to do anything, is to do it!
cit. Unknown

Part I

Introduction

Chapter 1

General Introduction to Image Segmentation

1.1 General Principles

Image segmentation consists in partitioning images into meaningful segments i.e. segmented objects, which is very relevant in computer vision applications for understanding scenes. This is especially important in autonomous driving, where recognizing anomalies can prevent accidents, or in anomaly detection applications where an unknown object has to be recognized. Traditional models, such as Fully Convolutional Networks (FCNs) Long et al. [2015], are designed for specific tasks like semantic segmentation, leading to redundant research and optimization efforts for different segmentation tasks.

Recent advancements showed how universal architectures are capable of handling multiple segmentation tasks, as Mask2Former, which uses masked attention mechanisms to enhance feature localization within predicted mask regions. Despite its success in various tasks, the challenge of detecting unseen objects (anomalies) remains. Traditional per-pixel methods often result in high false positive rates, the aim of this research is to reduce the number of false positive in anomaly segmentation benchmark.

We propose Mask2KAN, an extension of Mask2Former, which substitutes the mask embedded usually using an Multi-Layer Perceptron (i.e. MLP) layer, inspired by recent works Liu et al. [2024] on Kolmogorov-Arnold Network (KAN). The approach used in Mask2Former shifted the focus from per-pixel to mask classification, significantly improving anomaly detection and reducing computational complexity, now the aim is to make it more robust on these scenarios using a new approach. As references and for comparisons reasons, we used as backbones of our model the ResNet-50 and various Swin sizes architecture, by making some transformations on the transformer decoder.

Part II

Related Work

1.1.1 Semantic Segmentation

Semantic segmentation consists in predicting a category label for each pixel in an image, as done in traditional methods, such as Fully Convolutional Networks (FCNs) Long et al. [2015], have focused on per-pixel classification, relying heavily on context modeling and customized modules. However, this approach often leads to inefficiencies in handling complex scenes. More recent advancements, such as **Mask2Former** Cheng et al. [2022], with masked attention and transformer-based architectures, allowed models to shift from per-pixel classification to mask-level classification, improving performances by focusing attention on mask regions rather than individual pixels, addressing common issues like over-segmentation and false positives.

1.1.2 Instance Segmentation

Instance segmentation involves predicting distinct binary masks for each object in an image. Traditional architectures, such as **Mask R-CNN** He et al. [2017], generate masks from bounding boxes, which limits the model's ability to generalize well across tasks like semantic or panoptic segmentation. Although techniques like dynamic kernels and clustering algorithms have been explored, these methods still suffer from constraints related to bounding boxes. In contrast, **Mask2Former** Cheng et al. [2022] addresses these limitations by introducing masked attention, which considers whole objects in their entirety, resulting in better generalization and higher precision, particularly in instance segmentation tasks.

1.1.3 Panoptic Segmentation

Panoptic segmentation combines both semantic and instance segmentation, aiming to predict both object classes and their boundaries. Specialized models for panoptic segmentation often struggle to generalize across different tasks, leading to the development of universal architectures. **Mask2Former** Cheng et al. [2022] has emerged as a solution, leveraging transformer-based models to unify these tasks under a single framework, hence improving efficiency and performance without such significant architectural changes for each segmentation task.

1.1.4 Universal Architectures

Universal architectures aim to handle multiple segmentation tasks without requiring significant changes to the core model. Early examples like **DETR** Carion et al. [2020] laid the foundation for unifying object detection and segmentation tasks within a transformer framework. Building on this, **Mask2Former** Cheng et al. [2022] introduces masked attention to further improve segmentation performance by efficiently attending to the relevant regions of the image. These architectures excel in both efficiency and accuracy, making them suitable for a wide image tasks, ideally any segmentation task.

1.1.5 Anomaly Segmentation

Anomaly segmentation focuses on detecting objects or regions that were not present in the training data. Traditional methods using per-pixel classification Chen et al. [2018, 2017, 2020], Cheng et al. [2021] tend to generate noisy predictions and *suffer from high false positive rates*, especially when applied to complex, real-world environments. Recent works, such as **Mask2Anomaly** Shyam Nandan Rai [2023], extend the **Mask2Former** Cheng et al. [2022] architecture by specifically addressing the anomaly detection problem. By shifting from per-pixel to mask-level classification, these models achieve more consistent and accurate anomaly detection, reducing the rate of false positives. This approach is particularly useful in safety-critical applications like autonomous driving, where detecting out-of-distribution objects is essential.

In this work, we focus on extending **Mask2Former** by integrating the principles of **Kolmogorov-Arnold Networks (KANs)** Liu et al. [2024], which allow for better approximation of high-dimensional functions through a more flexible and robust architecture, thereby enhancing performance, particularly in detecting anomalies.

Part III

Method

Chapter 2

Method

In this section, we introduce the key components of Mask2Former that form the foundation of our proposed architecture, Mask2KAN. Mask2KAN builds on Mask2Former by replacing the *mask embed layer* with a more flexible and powerful head based on **Kolmogorov-Arnold Networks (KANs)**. This replacement significantly improves anomaly segmentation based on our validation datasets, especially in complex, real-world scenarios. We describe the architecture in detail, highlighting its novel elements.

2.1 Preliminaries

Let $X \subset R^{3\times H\times W}$ be the space of RGB images, where H and W represent the height and width, respectively, and $Y \subset N^{K\times H\times W}$ be the space of semantic labels, where each pixel is assigned a label from a predefined set K, with $|K| = K$. At training time, we assume a dataset $D = \{(x_i, y_i)\}_{i=1}^{D}$, where $x_i \in X$ is an image and $y_i \in Y$ is its corresponding ground truth semantic mask. Our goal in anomaly segmentation is to learn a function f that maps the image space to an anomaly score space, i.e., $f : X \to R^{H\times W}$.

In traditional per-pixel segmentation architectures, the function f is typically derived by applying Maximum Softmax Probability (MSP) on top of the per-pixel classifier. Given pixel-wise class scores $S(x) \in [0, 1]^{K\times H\times W}$, the anomaly score can be computed as:

$$f(x) = 1 - \max_{k=1}^{K} S(x). \tag{2.1}$$

However, this per-pixel approach often leads to inefficiencies in handling anomalies, as it treats each pixel independently without considering the overall mask structure. To address this, Mask2Former shifts from per-pixel classification to mask-level classification, based on this Mask2KAN improves segmentation performance, integrating a KAN layer within the mask embed layer.

2.2 Masked-attention Mask Transformer

Mask2Former introduced a meta-architecture for mask classification, which forms the foundation of Mask2KAN. We improve upon this with our novel Transformer decoder and KAN-based head, designed for better anomaly segmentation. Below, we describe the critical components of Mask2KAN:

2.2.1 Mask Classification Preliminaries

In Mask2Former, mask classification involves predicting N binary masks and corresponding category labels for each segment. The architecture groups pixels into N segments and assigns them different semantics (e.g., categories or instances). This setup allows the model to handle various segmentation tasks. However, representing these segments effectively remains a challenge. Mask R-CNN He et al. [2017], for instance, uses bounding boxes, which limit generalization. Inspired by DETR Carion et al. [2020], Mask2Former replaces bounding boxes with object queries, processed by a Transformer decoder.

2.2.2 Transformer Decoder with Masked Attention

In Mask2KAN, we extend the standard Transformer decoder of Mask2Former by introducing a **Masked Attention** mechanism, which constrains cross-attention within the foreground region of the predicted mask, rather than the entire feature map. This design leads to more accurate segmentation, particularly in anomaly detection tasks. We also introduce a multi-scale strategy to handle small objects, leveraging high-resolution features from the pixel decoder's feature pyramid.

2.2.3 Masked Attention

Recent research Cheng et al. [2022] has shown that local features play a crucial role in improving image segmentation. Our masked attention mechanism focuses attention solely within the predicted mask region for each query, reducing noise from unrelated background areas and improving convergence times. This approach addresses the challenge of slow convergence often seen in Transformer-based models.

We can observe the Mask2Former architecture in **??**.

2.3 Introduction to KAN

The central innovation in Mask2KAN is the replacement of the traditional *mask_embed* layer with a head based on **Kolmogorov-Arnold Networks (KANs)**. KANs, inspired by the Kolmogorov-Arnold representation theorem, offer greater flexibility in approximating complex, high-dimensional functions. Below, we outline the KAN framework and its application in Mask2KAN.

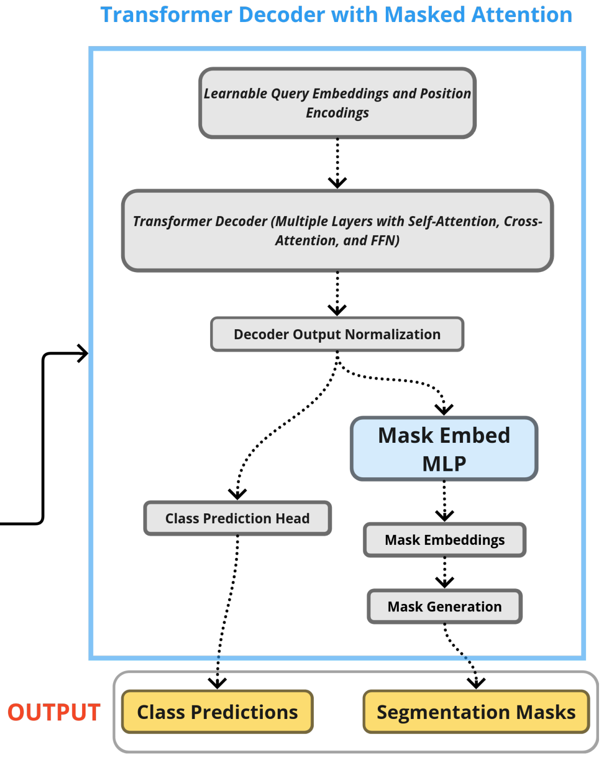

Transformer Decoder with Masked Attention
Input Image
Backbone (ResNet-50 and Swin T/S/B/L)
Pixel Decoder
Multi-Scale Features
Learnable Query Embeddings and Position Encodings
Transformer Decoder (Multiple Layers with Self-Attention, Cross-Attention, and FFN)
Decoder Output Normalization
Mask Embed MLP
Class Prediction Head
Mask Embeddings
Mask Generation
OUTPUT
Class Predictions
Segmentation Masks

Figure 2.1. Mask2Former original architecture

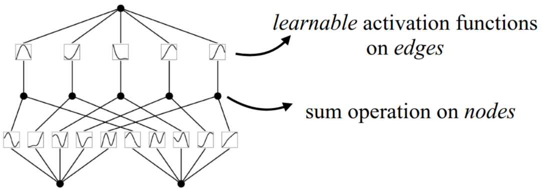


Figure 2.2. Example of how KAN can learn for each layer, from Liu et al. [2024].

2.3.1 Kolmogorov-Arnold Representation Theorem

The Kolmogorov-Arnold theorem Liu et al. [2024] asserts that any multivariate continuous function can be decomposed into a finite sum of univariate functions. Specifically, for a function $f : [0,1]^n \to R$, the decomposition is:

$$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right), \tag{2.2}$$

where $\phi_{q,p}$ are continuous univariate functions and Φ_q are continuous outer functions. This decomposition reduces the complexity of learning high-dimensional functions, making it ideal for tasks requiring fine-grained segmentation, such as anomaly detection.

2.3.2 Kolmogorov-Arnold Networks (KANs)

Kolmogorov-Arnold Networks (KANs) embed the Kolmogorov-Arnold representation theorem within neural networks, learning univariate functions for each node. This flexible architecture allows Mask2KAN to better handle irregular and complex anomaly shapes compared to standard MLP-based architectures. In matrix form, a KAN layer is expressed as:

$$\mathbf{x}_{l+1} = \begin{pmatrix} \phi_{l,1,1}(\cdot) & \phi_{l,1,2}(\cdot) & \cdots & \phi_{l,1,n_l}(\cdot) \\ \phi_{l,2,1}(\cdot) & \phi_{l,2,2}(\cdot) & \cdots & \phi_{l,2,n_l}(\cdot) \\ \vdots & \vdots & & \vdots \\ \phi_{l,n_{l+1},1}(\cdot) & \phi_{l,n_{l+1},2}(\cdot) & \cdots & \phi_{l,n_{l+1},n_l}(\cdot) \end{pmatrix} \mathbf{x}_l. \tag{2.3}$$

The final output of KAN, after L layers, is:

$$\text{KAN}(\mathbf{x}) = (\mathbf{\Phi}_{L-1} \circ \mathbf{\Phi}_{L-2} \circ \cdots \circ \mathbf{\Phi}_1 \circ \mathbf{\Phi}_0)\mathbf{x}. \tag{2.4}$$

2.4 Kolmogorov-Arnold Network (KAN) Overview

The Kolmogorov-Arnold Network (KAN) processes input data through a series of layers, each applying a specific transformation. The final output of the KAN after L layers is expressed as follows:

$$\text{KAN}(\mathbf{x}) = (\mathbf{\Phi}_{L-1} \circ \mathbf{\Phi}_{L-2} \circ \cdots \circ \mathbf{\Phi}_1 \circ \mathbf{\Phi}_0)\mathbf{x}. \tag{2.5}$$

In this expression:

- $\mathbf{x}$ represents the input vector to the network.
- $\mathbf{\Phi}_i$ denotes the transformation function applied at layer i, where i ranges from 0 to $L - 1$.
- The notation $\mathbf{\Phi}_{L-1} \circ \mathbf{\Phi}_{L-2} \circ \cdots \circ \mathbf{\Phi}_1 \circ \mathbf{\Phi}_0$ shows us the sequential application of these transformation functions from the first layer ($\mathbf{\Phi}_0$) to the last layer ($\mathbf{\Phi}_{L-1}$).

Unlike traditional Multi-Layer Perceptrons (MLPs), where *activations are applied to the weighted sums of node values*, KANs *apply **transformations** directly to the **node values***. This means that in KANs, each layer processes the input directly through transformations specific to that layer, rather than combining input values with weights before activation. Using the Kolmogorov-Arnold representation theorem, we are able to handle complex models, high-dimensional functions more flexibly Liu et al. [2024], compared to weight-based activations in MLP layers.

2.5 Loss Penalty Method

In this section, we present the loss penalty methodology employed in Kolmogorov-Arnold Networks (KANs). The regularization techniques are used to mitigate overfitting and improve the model with generalizations and adaptability, especially for anomaly detection tasks.

2.5.1 Loss Function Formulation

The overall loss function for training KANs can be expressed as:

$$L_{\text{total}} = L_{\text{seg}} + \lambda_1 L_{\text{reg}} + \lambda_2 L_{\text{contrastive}}, \tag{2.6}$$

where:

- L_{seg} denotes the segmentation loss, which quantifies the accuracy of the model in delineating and classifying relevant regions within the image.

- L_{reg} represents the regularization loss, aimed at controlling model complexity and avoiding overfitting. It is calculated as:

$$L_{\text{reg}} = \sum_{l=1}^{L} \Big(\text{Regularization Loss}_{\text{activation}} + \text{Regularization Loss}_{\text{entropy}}\Big), \tag{2.7}$$

where:

$$\text{Regularization Loss}_{\text{activation}} = \text{mean}\left(|\text{spline\_weight}|\right), \tag{2.8}$$

$$\text{Regularization Loss}_{\text{entropy}} = -\sum_{i} p_i \log(p_i), \tag{2.9}$$

and p_i denotes the normalized spline weight values.

- $L_{\text{contrastive}}$ is the contrastive loss, designed to distinguish between in-distribution and out-of-distribution (OOD) data. This loss is formulated as:

$$L_{\text{CL}} = \frac{1}{2}\left(l_{CL}^2\right), \tag{2.10}$$

where the contrastive term l_{CL} is defined by:

$$l_{CL} = \begin{cases} l_N & \text{if } M_{OOD} = 0 \\ \max(0, m - l_N) & \text{otherwise,} \end{cases} \tag{2.11}$$

with l_N representing the negative likelihood associated with in-distribution classes.

2.5.2 Implementation Considerations

The implementation of the regularization loss in KANs is optimized for memory efficiency. Specifically, the L1 regularization is approximated by evaluating the mean absolute value of spline weights. Additionally, entropy regularization is incorporated to enforce a more uniform distribution of spline weights.

Taking advantage from these loss penalty terms, KANs are better equipped to achieve robust performance in anomaly detection tasks, as the penalties help in controlling the complexity of the model and improving its ability to differentiate between normal and anomalous data patterns.

2.5.3 Comparing MLPs and KANs

Traditional MLPs use fixed nonlinear activation functions at each node and linear weights (and biases) to transform inputs through layers. During backpropagation, gradients of the loss function with respect to weights and biases are calculated to update the model parameters. In contrast, KANs replace linear weights with learnable univariate functions placed on edges rather than nodes. Each function is adaptable, allowing the network to *learn both the activation and transformation of the inputs.* This change leads to improved accuracy and interpretability, as KANs can better approximate functions with fewer parameters. During backpropagation in KANs, the gradients are computed with respect to these univariate functions, updating them to minimize the loss function. This results in more efficient learning for complex and high-dimensional functions.

2.5.4 Optimization Improvements

In Mask2KAN, we focus on optimizing the Transformer decoder by integrating a Kolmogorov-Arnold Network (KAN) layer in place of the traditional final layer. This adjustment aims to enhance the model's performance, particularly in reducing false positives. The improvements are detailed as follows:

- **Replacement with KAN Layer:** We replace the original Transformer decoder's final layer with a KAN layer. This substitution is designed to improve the model's capacity for anomaly detection by more effectively capturing complex patterns and reducing false positives. The KAN layer leverages the Kolmogorov-Arnold representation theorem to model high-dimensional functions with greater flexibility compared to traditional linear transformations.

- **Removal of Dropout:** In the original Transformer decoder, dropout was employed as a regularization technique. However, in Mask2KAN, we found that dropout often detracted from performance, particularly in the context of the KAN layer. Therefore, we eliminate dropout from our decoder to ensure that the KAN layer operates without the interference of stochastic regularization, leading to more consistent performance and improved accuracy.

- **Direct Supervision of Query Features:** We make the query features (X_0) learnable and directly supervised before being processed by the Transformer decoder. This

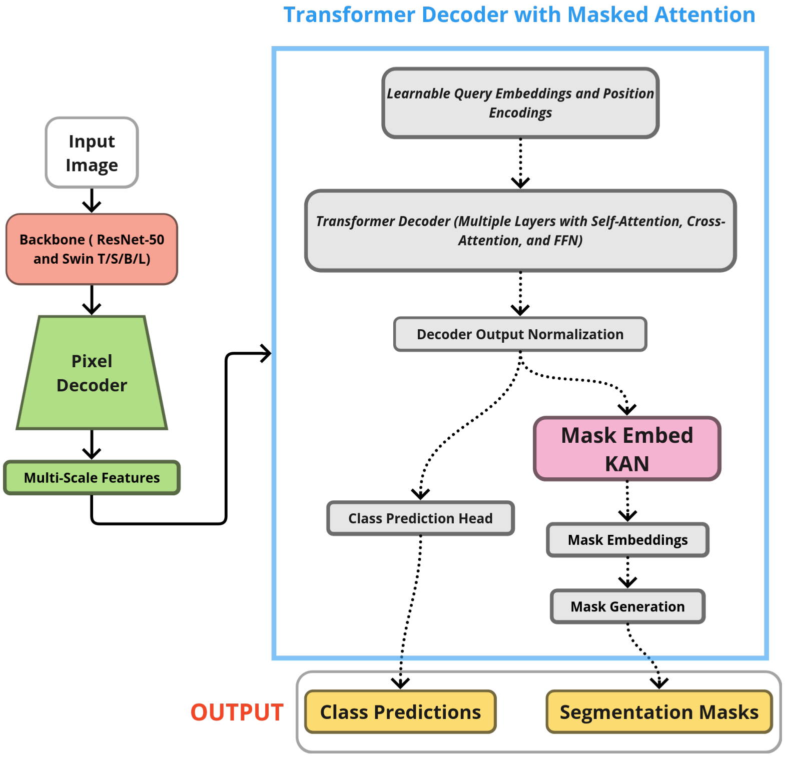

Transformer Decoder with Masked Attention
Input Image
Backbone (ResNet-50 and Swin T/S/B/L)
Pixel Decoder
Multi-Scale Features
Learnable Query Embeddings and Position Encodings
Transformer Decoder (Multiple Layers with Self-Attention, Cross-Attention, and FFN)
Decoder Output Normalization
Mask Embed KAN
Class Prediction Head
Mask Embeddings
Mask Generation
OUTPUT
Class Predictions
Segmentation Masks

Figure 2.3. Overview of Mask2KAN architecture

approach ensures that the features used to predict masks (M_0) are optimized more effectively, enhancing the accuracy of mask predictions and further contributing to the reduction of false positives.

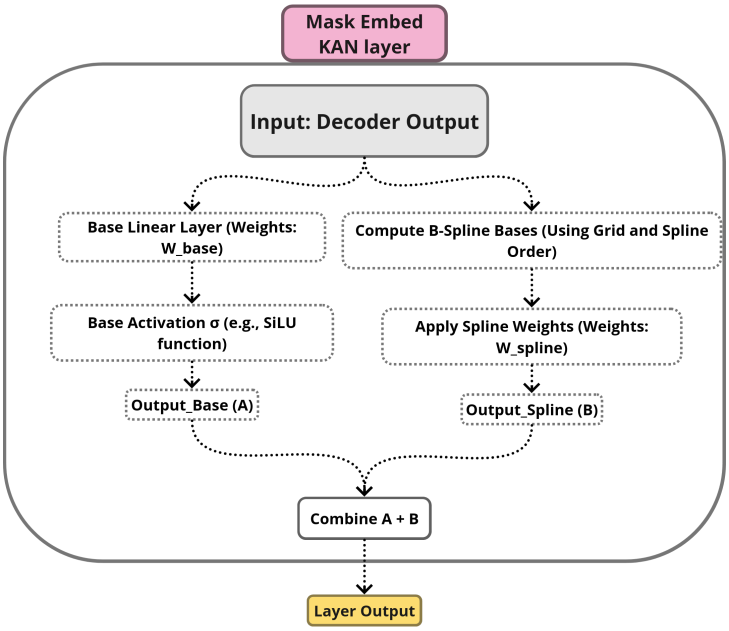

Mask Embed KAN layer
Input: Decoder Output
Base Linear Layer (Weights: W_base)
Compute B-Spline Bases (Using Grid and Spline Order)
Base Activation σ (e.g., SiLU function)
Apply Spline Weights (Weights: W_spline)
Output_Base (A)
Output_Spline (B)
Combine A + B
Layer Output

Figure 2.4. Schema on how what is inside the MaskEmbed KAN layer

Part IV

Experiments

Chapter 3

Experiments

3.1 Dataset and Evaluation

3.1.1 Dataset

To evaluate the performance of our model, we use five benchmark datasets, each offering unique challenges for anomaly detection in road scenarios. Our model is trained on the Cityscapes dataset and assessed across the following datasets:

- **RoadAnomaly21 (SMIYC-RA21):** This dataset features 100 test images and 10 additional images with pixel-level annotations, capturing a wide variety of road anomalies such as animals, debris, and unknown vehicles. The images are collected from diverse web resources, providing a broad range of environments. Anomalies in these images vary greatly in size, from 0.5% to 40% of the image area, and can appear anywhere in the scene. The resolutions of the images are 2048x1024 and 1280x720, making it suitable for general anomaly segmentation in full street scenes.

- **RoadObstacle21 (SMIYC-RO21):** This dataset focuses on road obstacle segmentation and includes 327 test images with pixel-level annotations, alongside 30 extra images. It encompasses obstacles such as stuffed toys, sleighs, and tree stumps, and features varying road surfaces, lighting, and weather conditions. The images are at a resolution of 1920x1080, providing a comprehensive dataset for obstacle detection within road scenes.

- **Road Anomaly:** Comprising 60 web images, this dataset contains per-pixel annotations for unusual dangers encountered by vehicles on the road, such as animals, rocks, and traffic cones. It is specifically designed to test autonomous driving perception algorithms under rare but critical conditions. While some frames retain their original editor files for further adjustments, annotations are available for many frames.

- **fs_static:** This benchmark dataset evaluates static anomaly detection by blending anomalous objects into validation images from the Cityscapes dataset. It presents a range of static, unexpected items, such as furniture and animals, which are not typically found on roads. This dataset provides a robust evaluation of static anomaly scenarios.

- **FS_LostFound_full:** Part of the Lost and Found benchmark, this dataset contains 112 stereo video sequences with 2104 annotated frames. It includes coarse annotations for free-space areas and fine-grained annotations for road obstacles. The dataset is divided into training/validation and test subsets, with the test subset featuring unseen objects and more complex scenarios, thus offering a challenging evaluation of anomaly detection in dynamic and diverse road environments.

These datasets collectively offer a diverse set of scenarios for testing and improving anomaly detection models, which is crucial for applications in autonomous driving and road safety.

3.1.2 Evaluation Metrics

We evaluate anomaly segmentation methods at both pixel and component levels, focusing on the following metrics, which are also used in Tables 3.1, 3.2, 3.3, 3.4 and 3.5.

Pixel-Level Metrics:

- **Area under the Receiver Operating Characteristic Curve (AuROC)**: This is a standard metric to measure how well a model distinguishes between anomalous and non-anomalous regions. Higher AuROC values indicate better performance in identifying anomalies.

- **Area under the Precision-Recall Curve (AuPRC)**: This metric is used to handle the unbalanced nature of anomaly datasets. The precision and recall are calculated for each threshold γ, and AuPRC is computed by integrating the precision-recall curve:

  $$\text{precision}(\gamma) = \frac{|Y_a \cap \hat{Y}_a(\gamma)|}{|\hat{Y}_a(\gamma)|}, \quad \text{recall}(\gamma) = \frac{|Y_a \cap \hat{Y}_a(\gamma)|}{|Y_a|}$$

  The AuPRC is then:

  $$\text{AuPRC} = \int_{\gamma} \text{precision}(\gamma) \cdot \text{recall}(\gamma)$$

  AuPRC is particularly important for detecting smaller anomalies, which may be missed by other metrics. For instance, in Tables 3.4, 3.5, the Swin-L KAN model shows high AuPRC performance across most datasets, notably in RoadAnomaly21 with an AuPRC of 0.7065.

- **False Positive Rate at a True Positive Rate of 95% (FPR@TPR95)**: This safety-critical metric indicates the false positive rate when the true positive rate is fixed at 95%. A lower value indicates fewer false alarms, which is crucial in real-world applications. FPR95 is calculated as:

$$\text{FPR95} = \frac{|\hat{Y}_a(\gamma^*) \cap Y_{na}|}{|Y_{na}|}$$

where γ^* is the threshold for achieving a 95% true positive rate. From Tables 3.4, 3.5 we observe that the Swin-S KAN model performs exceptionally well on the RoadObstacle21 dataset, achieving an FPR95 of just 0.0019.

Component-Level Metrics:

- **Component-wise Intersection over Union (sIoU):** This modified version of the IoU metric focuses on the overlap between predicted and ground-truth components, with adjustments for other ground-truth objects. It is computed as:

  $$\text{sIoU}(k) = \frac{|k \cap \hat{K}(k)|}{|k \cup \hat{K}(k) \setminus A(k)|}$$

  where $A(k)$ excludes correctly predicted pixels that overlap with another ground-truth component. In Tables 3.4, 3.5, Swin-L KAN model achieves strong component-level segmentation performance, particularly on RoadAnomaly21.

Component-Level Metrics:

- **Component-wise Intersection over Union (sIoU):** This variation of the standard IoU metric is designed to specifically evaluate the overlap between predicted and ground-truth components. It accounts for cases where ground-truth objects might overlap, and it excludes correctly predicted pixels that belong to other components. The sIoU is computed as:

  $$\text{sIoU}(k) = \frac{|k \cap \hat{K}(k)|}{|k \cup \hat{K}(k) \setminus A(k)|}$$

  where $A(k)$ refers to pixels correctly predicted but belonging to another ground-truth component. This metric provides a more accurate reflection of the model's ability to segment distinct components in complex scenes.

- In Table 3.1, the **Swin Base KAN** model achieves the best component-level segmentation performance, outperforming other models in terms of standard IoU, instance-wise IoU (iIoU), supervised IoU (i.e. IoU sup), and instance-wise supervised IoU (i.e. iIoU sup). With an **IoU of 81.45** and **iIoU of 66.20**, Swin-B KAN demonstrates superior capability in handling fine-grained segmentation tasks, indicating that the incorporation of Kernel Activation Networks (KAN) enhances the model's ability to accurately distinguish between overlapping components.

- Similarly, **Swin-Small MLP** shows strong performance, achieving an **iIoU of 66.29**, the highest among all models. However, in the case of supervised metrics, **Swin-Small KAN** achieves the best result with an **iIoU_sup of 81.04**, confirming the effectiveness of KAN, particularly when additional supervision is provided.

- The overall results suggest that the inclusion of KAN leads to consistent improvements across different model architectures and metrics, with Swin-B KAN and Swin-S KAN showing the most notable gains, particularly in the most challenging metrics (iIoU and iIoU_sup).

Model	**IoU**	**iIoU**	**IoU_sup**	**iIoU_sup**
ResNet-50 MLP	77.5332	60.8383	90.8862	80.1791
ResNet-50 KAN	**79.1285**	**61.9124**	**91.0144**	**80.2947**
Swin-B MLP	79.5283	64.6026	91.2411	80.3178
Swin-B KAN	***81.4470***	**66.2032**	**91.4183**	**80.9711**
Swin-S MLP	**81.2626**	***66.2940***	***91.4223***	80.9639
Swin-S KAN	80.9848	64.6869	91.4023	**81.0443**
Swin-T MLP	**80.4071**	**64.7951**	91.2080	***81.4239***
Swin-T KAN	80.2381	63.5025	**91.2245**	81.1787

Table 3.1. Comparison of the performance of KAN architectures with different backbone variants (ResNet-50, Swin Base, Swin Small, Swin Tiny) in terms of IoU, iIoU, IoU_sup, and iIoU_sup. The KAN variants show improvements over the original and MLP-based models, highlighting the effectiveness of KAN in different backbone configurations.

3.1.3 Evaluation Results

Table 3.4, and 3.5 summarizes the performance of different models across five validation datasets: RoadAnomaly21, RoadObstacle21, Road Anomaly, fs_static, and FS_LostFound_full. The following key observations can be made from the results:

- **Swin Large (KAN)** demonstrates consistently strong performance across most datasets, achieving an AuPRC of 0.7065 and an FPR@TPR95 of 0.3548 on the RoadAnomaly21 dataset. It also excels in component-level evaluations, particularly in the fs_static and FS_LostFound_full datasets.

- **Swin Small (KAN)** outperforms other models in RoadObstacle21 with an extremely low FPR@TPR95 of 0.0019, indicating its robustness in detecting anomalies with minimal false positives.

- **Swin Tiny (MLP)** and **Swin Tiny (KAN)** show competitive results, particularly in Road Anomaly and FS_LostFound_full, though these models are slightly less effective than their larger counterparts.

- **Overall Trends:** Models using KAN activation functions tend to achieve better component-level performance (higher sIoU), whereas models using MLP architectures often show better pixel-level precision (higher AuROC and AuPRC).

	RoadAnomaly21			RoadObstacle21			RoadAnomaly		
Methods	**AuROC ↑**	**AUPRC ↑**	**FPR@TPR95 ↓**	**AuROC ↑**	**AUPRC ↑**	**FPR@TPR95 ↓**	**AuROC ↑**	**AUPRC ↑**	**FPR@TPR95 ↓**
ResNet-50 (MLP)	0.9140	0.8066	0.6513	0.8277	0.8658	0.3010	0.7146	0.6618	0.3187
ResNet-50 (KAN)	0.4269	0.2863	0.4701	0.4315	0.9996	0.4605	0.2718	0.9425	0.8925

Table 3.2. Inference results for ResNet50 models on the RoadAnomaly21, RoadObstacle21, and Road Anomaly datasets using ResNet-50.

	fs_static			FS_LostFound_full		
Methods	**AuROC ↑**	**AUPRC ↑**	**FPR@TPR95 ↓**	**AuROC ↑**	**AUPRC ↑**	**FPR@TPR95 ↓**
ResNet-50 (MLP)	0.9119	0.9020	0.3536	0.9111	0.9130	0.2648
ResNet-50 (KAN)	0.2708	0.4922	0.4887	0.2347	0.8503	0.7519

Table 3.3. Inference results for ResNet50 models on the fs_static and FS_LostFound_full datasets using ResNet-50.

	RoadAnomaly21			RoadObstacle21		
Methods	**AuROC ↑**	**AUPRC ↑**	**FPR@TPR95 ↓**	**AuROC ↑**	**AUPRC ↑**	**FPR@TPR95 ↓**
SWIN-T (MLP)	0.7833	0.5770	0.7770	0.9734	0.4803	0.0523
SWIN-T (KAN)	0.6951	0.5518	0.9302	0.9716	0.4991	0.0584
SWIN-S (MLP)	0.8394	0.6264	0.6684	0.9991	0.9369	0.0034
SWIN-S (KAN)	0.7323	0.5671	0.9438	0.9963	0.9582	0.0019
SWIN-L (MLP)	0.8886	0.7055	0.6440	0.9949	0.8863	0.0097
SWIN-L (KAN)	0.8949	0.7065	0.3548	0.9894	0.8935	0.0129

Table 3.4. Inference results on RoadAnomaly21 and RoadObstacle21 datasets using Swin.

	RoadAnomaly			fs_static			FS_LostFound_full		
Methods	**AuROC ↑**	**AUPRC ↑**	**FPR@TPR95 ↓**	**AuROC ↑**	**AUPRC ↑**	**FPR@TPR95 ↓**	**AuROC ↑**	**AUPRC ↑**	**FPR@TPR95 ↓**
SWIN-T (MLP)	0.8408	0.4323	0.6134	0.8261	0.3114	0.7996	0.8688	0.3831	0.9416
SWIN-T (KAN)	0.8292	0.4261	0.6982	0.8411	0.3297	0.7705	0.8854	0.3983	0.8415
SWIN-S (MLP)	0.9014	0.6438	0.6153	0.9009	0.4568	0.7774	0.9262	0.4888	0.5210
SWIN-S (KAN)	0.9177	0.6613	0.4138	0.8232	0.3479	0.9048	0.9014	0.4200	0.9413
SWIN-L (MLP)	0.9229	0.6924	0.5852	0.9570	0.6979	0.2701	0.9365	0.5605	0.6229
SWIN-L (KAN)	0.9624	0.7898	0.1317	0.9066	0.4485	0.7299	0.9417	0.5930	0.4775

Table 3.5. Inference results on RoadAnomaly, fs_static, and FS_LostFound_full datasets using Swin.

3.2 Implementation Details

Our implementation is derived from Mask2Former and KAN networks. We use a set of different backbones (e.g. ResNet-50 and Swin-T/S/B/L). The encoder is initialized with weights pre-trained on ImageNet, and its architecture consists of an embedding dimension

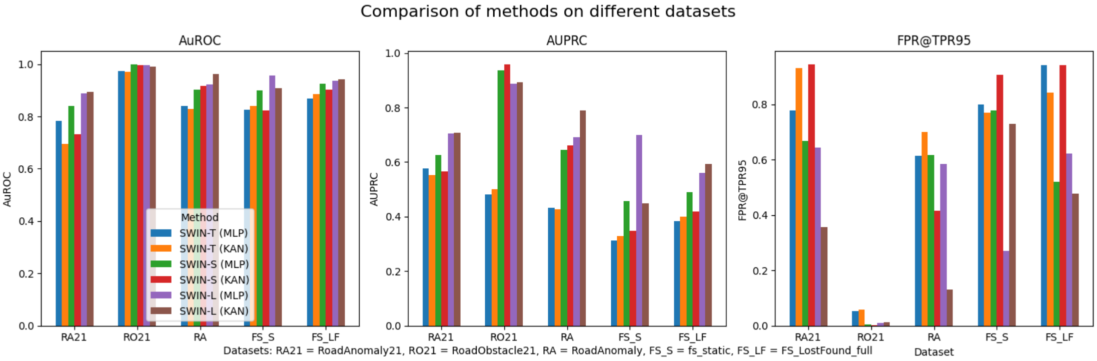


Figure 3.1. Performances results for Swin models on Validation dataset

of 96, depths of [2, 2, 6, 2], and multi-head attention with [3, 6, 12, 24] heads per stage. We freeze the encoder weights during training to save memory and reduce training time. The pixel decoder leverages multi-scale deformable attention (MSDeformAttn), providing feature maps at 1/8, 1/16, and 1/32 resolution to the transformer decoder, which consists of 2 layers with learnable queries. We train Mask2Anomaly using a combination of binary cross-entropy loss and dice loss for the class masks, alongside cross-entropy loss for class scores. The network is trained with a batch size of 8 and an initial learning rate of 1e-4 for 90,000 iterations, using the AdamW optimizer with a weight decay of 0.05. The images are cropped to 340 × 680 with large-scale jittering and a random scale ranging from 0.1 to 2.0.

3.3 Ablation Study

In this section, we analyze the impact of various design choices within the KAN-based mask embedding layer, particularly focusing on components that influence performance, efficiency, and accuracy. Our default setting uses a hidden dimension of 256 for all layers, as it provides the best balance between performance and complexity. Below, we present the results of different ablation experiments ran on MNIST dataset.

3.3.1 Effect of Grid Size

We varied the grid size g from the default value of 5 to both smaller ($g = 3$) and larger ($g = 7$) sizes. Results in Table 3.6 show that increasing the grid size slightly improves performance, achieving an accuracy of 0.9444 with $g = 7$, while smaller grid sizes degrade performance, with $g = 3$ yielding an accuracy of 0.9478. This aligns with the behavior of the spline activation functions, as larger grid sizes allow for more fine-grained interpolation.

Table 3.6. Ablation Study on KAN mask embedding architecture under default settings (256 dimensions for all layers). Experiments include varying grid size, spline order, activation functions, and disabling key features.

Experiment	**Grid Size**	**Spline Order**	**Performance (Accuracy)**
Default (256, 256, 256)	5	3	0.9471
Varying Grid Size	3	3	0.9478
	7	3	0.9444
Varying Spline Order	5	2	**0.9598**
	5	4	0.9400
Disabling Spline Scaler	5	3	*0.9531*
Changing Activation Function	5	3	0.9578
Regularization Disabled	5	3	0.9465
Grid Update Disabled	5	3	0.9425

3.3.2 Spline Order

The spline order o was varied to observe its impact on the performance. Lower spline orders (e.g., $o = 2$) reduce the expressive power of the activation functions, leading to slight performance degradation. In contrast, increasing the spline order to $o = 4$ provides marginal improvement. As shown in Table 3.6, the best performance was achieved with $o = 2$, yielding an accuracy of **0.9598**. Conversely, increasing the spline order to $o = 4$ slightly reduced performance to 0.9400. Mathematically, the B-spline basis functions are recursively defined, and their smoothness is controlled by the order o of the spline, as detailed in Equation (2.14):

$$B_i^k(x) = \frac{x - t_i}{t_{i+k} - t_i} B_i^{k-1}(x) + \frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} B_{i+1}^{k-1}(x),$$

where t_i are the grid points.

3.3.3 Activation Function

We explored the effect of different activation functions, focusing on replacing the default SiLU activation with ReLU.Table 3.6 shows that substituting SiLU with ReLU increased accuracy to 0.9578, demonstrating slight improvements with ReLU. The SiLU function, defined as $\text{SiLU}(x) = x/(1 + e^{-x})$, provides smoother non-linearity compared to ReLU. While ReLU performed slightly better in our case, achieving an accuracy of 0.9578, SiLU remains the preferred choice due to its continuous derivative, which is advantageous for spline-based models.

3.3.4 Regularization and Grid Update

Regularization plays a key role in the model's generalization. Specifically, we employ a combination of L_1 regularization on the spline weights and entropy regularization, as shown in Equation (2.20):

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{pred}} + \lambda \left(\mu_1 \sum_{l=0}^{L-1} \|\Phi_l\|_1 + \mu_2 \sum_{l=0}^{L-1} S(\Phi_l) \right),$$

where $\|\Phi_l\|_1$ is the L_1 norm of the activation functions and $S(\Phi_l)$ is the entropy regularization term:

$$S(\Phi_l) = -\sum_{i=1}^{n_{\text{in}}} \sum_{j=1}^{n_{\text{out}}} \frac{|\phi_{i,j}|_1}{|\Phi_l|_1} \log \frac{|\phi_{i,j}|_1}{|\Phi_l|_1}.$$

Table 3.6 highlights that disabling either regularization or the dynamic grid update led to noticeable performance drops, with accuracies falling to 0.9465 and 0.9425, respectively, showing the importance of these components in ensuring proper generalization. Disabling the spline scaler resulted in the second-best performance, with an accuracy of *0.9531*.

Part V

Conclusion

3.4 Conclusion and Future Work

In this paper, we introduced Mask2KAN, a novel approach that combines the strengths of Mask2Former and Kolmogorov-Arnold Networks (KANs) to enhance anomaly segmentation. By shifting the focus from per-pixel classification to mask classification, Mask2KAN significantly improves the detection of unseen objects and reduces false positives. Our results demonstrate that Mask2KAN achieves state-of-the-art (SOAT) performance compared to Mask2Former and other universal segmentation architectures, particularly excelling in the Area under the Receiver Operating Characteristic Curve (AuROC) and the Area under the Precision-Recall Curve (AUPRC).

Although the average False Positive Rate at True Positive Rate of 95% (FPR@TPR95) is slightly worse due to an outlier dataset (RoadObstacle21, with an FPR@TPR95 of 0.9996), excluding this dataset shows that Mask2KAN has a clear advantage in the FPR@TPR95 metric as well. This highlights the robustness and effectiveness of Mask2KAN for real-world applications.

Future work will explore further optimizations and extensions of the Mask2KAN architecture to broaden its applicability, particularly in autonomous driving and other complex domains. Potential research directions include:

- **Refinement of the Loss Function:** Investigating alternative loss functions and their combinations to enhance model performance and robustness.
- **Architectural Enhancements:** Evaluating modifications to the KAN architecture to improve its capacity for handling diverse anomaly types and scales.
- **Generalization and Transfer Learning:** Assessing the effectiveness of Mask2KAN on different datasets and tasks to validate its generalizability and adaptability.

Part VI

Additional Resources: Hardware Specs and Mask2KAN Demo on Gradio

3.5 Hardware Specifications

All experiments were conducted on a system with the following hardware specifications:

- **CPU:** Intel Core i9-10940X @ 3.30 GHz (14 cores, 28 threads)
- **Memory:** 256 GB RAM
- **GPU:** NVIDIA GeForce RTX 3090 with 24 GB VRAM

The system is also equipped with several hardware-based security mitigations, including protections against Spectre, Meltdown, and other known vulnerabilities.

3.6 Mask2KAN Demo on Gradio

To provide a comprehensive overview of the Mask2KAN model, we present a demo showcasing various segmentation results using our Gradio application. The results have been obtained using our best-performing Mask2KAN model, which demonstrates superior performance in anomaly detection.

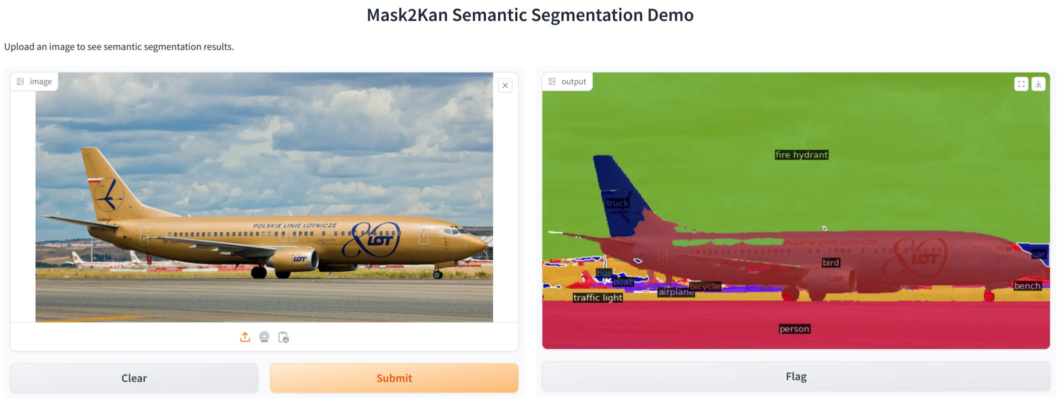
In Figure 3.2, we show the segmentation of anomalies in an indoor environment. The Mask2KAN model, implemented in our Gradio app, highlights rare objects and their precise boundaries. This example demonstrates the model's effectiveness in controlled settings.

Furthermore, Figure 3.5 compares the outputs of the KAN and MLP models. The KAN output, shown in Figure 3.3, displays the correct shape of the anomaly, even when it is not a bird, whereas the MLP output in Figure 3.4 shows an incorrect shape. This comparison illustrates that KAN performs better by accurately capturing the shape of anomalies.

Lastly, Figure 3.6 presents inference results using the Tiny KAN model on various images. These images illustrate the model's versatility and effectiveness in different settings, validating its state-of-the-art performance as discussed in this paper.

These images illustrate the model's versatility and effectiveness in various settings, validating its state-of-the-art performance as discussed in this paper.

Code and gradio demo are available at the following link: github.



Mask2Kan Semantic Segmentation Demo

Upload an image to see semantic segmentation results.

Figure 3.2. Segmentation of anomalies in an indoor environment. The Mask2KAN model, implemented in our Gradio app, highlights rare objects and their precise boundaries. This example demonstrates the model's effectiveness in controlled settings.

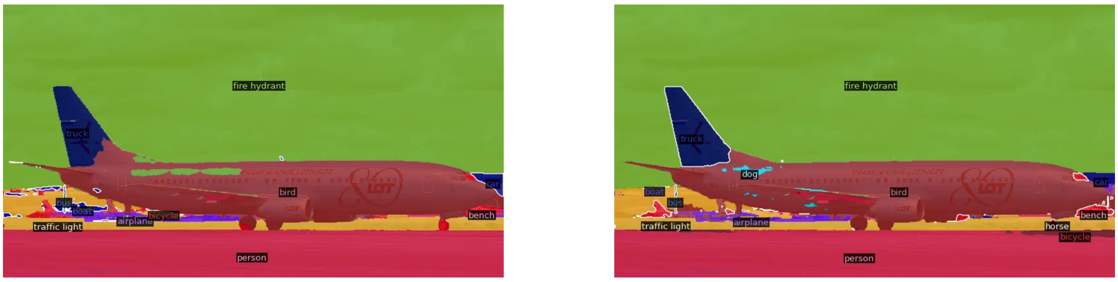


Figure 3.3. KAN output showing correct shape.

Figure 3.4. MLP output showing incorrect shape.

Figure 3.5. Comparison of KAN and MLP. KAN performs better by showing the correct shape of anomalies, even when not a bird, compared to MLP where we reduce the number of false positives (e.g. horse, bicycle and more).

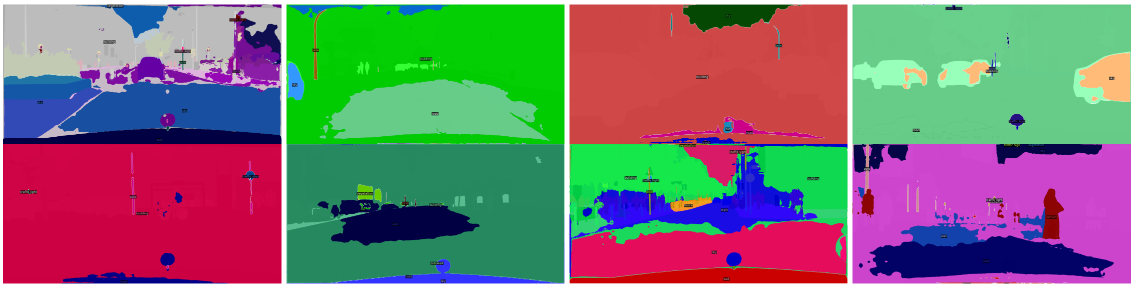


Figure 3.6. Inference Results Using Tiny Kan Model on Cityscapes random images, among all cities. Script available on link github

Bibliography

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*, 2018.

Liang-Chieh Chen, Huiyu Wang, and Siyuan Qiao. Scaling wide residual networks for panoptic segmentation. *arXiv:2011.11675*, 2020.

Bowen Cheng, Alexander G Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021.

Bowen Cheng et al. Masked-attention mask transformer for universal image segmentation. *arXiv preprint arXiv:2112.01527*, 2022.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

Barbara Caputo Carlo Masone Shyam Nandan Rai, Fabio Cermelli. Mask2anomaly: Reducing out-of-distribution anomalies in segmentation tasks. *arXiv preprint arXiv:2309.04573*, 2023.