

Article

Not peer-reviewed version

HKAN: A Hybrid Kolmogorov-Arnold Network for Robust Fabric Defect Segmentation

Min Li , [Pei Ye](#) * , Shuqin Cui , Ping Zhu , [Junping Liu](#)

Posted Date: 28 November 2024

doi: 10.20944/preprints202411.2185.v1

Keywords: Fabric Defect Detection; CNN; Transformer; KANs; Semantic Segmentation




Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

HKAN: A Hybrid Kolmogorov-Arnold Network for Robust Fabric Defect Segmentation

Min Li, Pei Ye , Shuqin Cui [†], Ping Zhu [†] and Junping Liu [†]

School of Computer and Artificial Intelligence, Wuhan Textile University, Wuhan, 430200, China; 2008031@wtu.edu.cn (M.L.); csq@wtu.edu.cn (S.C.); zhuping@wtu.edu.cn (P.Z.); jpliu@wtu.edu.cn (J.L.);

* Correspondence: ye-pei@outlook.com;

[†] These authors contributed equally to this work.

Abstract: Currently, fabric defect detection methods predominantly rely on CNN models. However, due to the inherent limitations of CNNs, such models struggle to capture long-distance dependencies in images and fail to accurately detect complex defect features. While Transformers excel at modeling long-range dependencies, their quadratic computational complexity poses significant challenges. To address these issues, we propose combining CNNs with Transformers and introduce Kolmogorov-Arnold Networks (KANs) to enhance feature extraction capabilities. Specifically, we designed a novel network for fabric defect segmentation, named HKAN, consisting of three components: encoder, bottleneck, and decoder. First, we developed a simple yet effective KANConv Block using KAN convolutions. Next, we replaced the MLP in PoolFormer with KAN, creating a lightweight KANTransformer Block. Finally, we unified the KANConv Block and the KANTransformer Block into a Hybrid KAN Block, which serves as both the encoder and bottleneck of HKAN. Extensive experiments on three fabric datasets demonstrate that HKAN outperforms mainstream semantic segmentation models, achieving superior segmentation performance and delivering prominent results across diverse fabric images.

Keywords: fabric defect detection; CNN; transformer; KANs; semantic segmentation

1. Introduction

Fabric defect detection is a quality control process that employs various technical methods to inspect textiles either during production or after the completion of finished products [1]. Common defects include issues such as uneven coloration, inconsistent texture, structural damages (e.g., yarn breakage, holes, and cracks), surface imperfections (e.g., oil stains and wrinkles), and edge problems like bad selvedge and color discrepancies [2]. The goal of fabric defect detection is to ensure that textiles meet quality standards, minimize the presence of defective products in the market, and enhance the overall competitiveness of the products.

In the early textile manufacturing industry, fabric defects were primarily identified through manual visual inspection [3]. However, with advancements in industrial automation, this labor-intensive and time-consuming approach has been gradually replaced. In recent years, deep learning, particularly convolutional neural networks (CNNs), has revolutionized computer vision and provided new solutions for fabric defect detection. For instance, Zhao et al. [4] proposed MSCNN, which achieves efficient and accurate real-time defect detection by capturing multi-scale feature maps and incorporating K-means clustering analysis. Similarly, Huang et al. [5] combined segmentation and decision networks to deliver high-precision, real-time defect detection. Beyond standard CNN applications, some researchers have explored integrating domain-specific knowledge to improve detection. Koulali et al. [6] introduced an unsupervised learning framework with dynamic feature selection to simplify training and enhance detection accuracy. Chen et al. [7] improved Faster R-CNN [8] by incorporating Gabor filters into its first convolutional layer, effectively reducing interference from background textures, and used a genetic algorithm to optimize filter parameters. Lu et al. [9] proposed TADet, a texture-aware, one-stage detection network that leverages adaptive feature fusion and multi-task training, addressing the limitations of traditional one-stage approaches by incorporating crucial texture information. While these CNN-based methods demonstrate strong performance in detecting simple, localized defects (e.g.,

Figure 1 (a)), they struggle with modeling long-range dependencies. This limitation becomes evident when defects span multiple regions or involve complex background textures, as shown in Figure 1 (c), where performance significantly degrades.

Another notable model in deep learning is Transformer [10], based on self-attention mechanism, which effectively captures long-range dependencies between sequences and has achieved remarkable results in natural language processing. Vision Transformer (ViT) [11] is the first to apply the Transformer architecture to the computer vision, yielding impressive results in image classification tasks. Following this, Swin Transformer [12] introduced a hierarchical structure and a more efficient self-attention mechanism, achieving state-of-the-art (SOTA) results across various downstream computer vision tasks, marking a significant milestone in the field. Inspired by these advancements, some researchers have also begun to explore the use of Transformers for fabric defect detection. Qu et al. [13] proposed a novel network called U-SMR for fabric defect detection. U-SMR integrates ResNet-50 [14] with Swin Transformer modules, combining global contextual features, defect detail features, and high-level semantic features, achieving excellent performance on the ZJU-Leaper dataset [3]. Xu et al. [15] introduced a fabric defect segmentation model based on an improved Swin-Unet [16] and Gabor filters. Specifically, the model incorporates Gabor filters to extract low-level texture features, reducing texture interference and accelerating convergence during the early training stages. Additionally, a non-fully symmetric U-shaped architecture is designed to simplify model training, while multi-stage result fusion is employed to achieve precise defect localization.

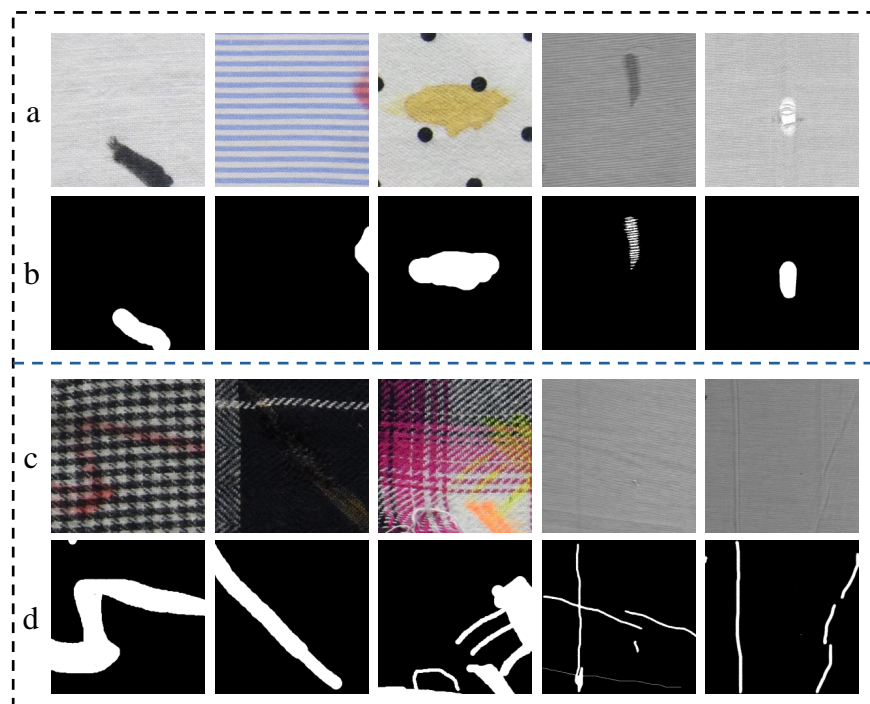


Figure 1. Example of fabric defects. (a) and (c) are defects in simple and complex textured fabrics, respectively, while (b) and (d) are their corresponding ground truth.

Regardless of whether they are CNNs or Transformers, an indispensable component of these models is the Multi-Layer Perceptrons (MLPs) [17]. Serving as a core building block of neural networks, MLPs play a crucial role in various deep learning architectures due to their strong capability for nonlinear fitting. Recently, the emergence of Kolmogorov-Arnold Networks (KANs) [18] has demonstrated the potential to address this issue. KANs enhance model interpretability by introducing learnable activation functions on the edges rather than at the nodes, providing a novel perspective. Moreover,

by replacing weight parameters with learnable functions on the edges, KANs utilize model parameters more efficiently, thereby reducing the computational burden of the model. In this paper, we aim to explore the potential of integrating KANs with CNNs and Transformers, applying this combination to the field of fabric defect segmentation. Our main contributions are as follows:

- We constructed a Hybrid KAN model that integrates CNNs, Transformers, and KANs into a unified framework, leveraging the strengths of each component. CNNs are used to extract local features, Transformers capture global dependencies, while KANs accurately capture complex patterns and subtle differences in fabric defect images through their flexible nonlinear feature extraction capabilities.
- We proposed a simple KAN Conv Block (KCB) using KANConv to extract local features from images. Compared to traditional convolutional modules, KCB reduces model complexity and parameter count while maintaining similar accuracy.
- We replaced the traditional MLP in the Transformer architecture with KAN, constructing a new KAN Transformer Block (KTB) for extracting global features from images. KTB leverages the characteristics of KAN to enhance the Transformer's ability to capture global contextual information in images, making it more adaptable when dealing with complex fabric texture images.

2. Related Work

2.1. Semantic Segmentation

The foundational work in semantic segmentation, FCN [19], achieved pixel-level prediction by replacing fully connected layers with convolutional layers. U-Net [20] further advanced this field by introducing an encoder-decoder structure that effectively combines low-level and high-level features through skip connections, making it one of the most widely used architectures across diverse applications. Subsequently, CNN-based methods [21–25] have pushed the boundaries of semantic segmentation by expanding receptive fields, integrating multi-scale information, and employing feature fusion techniques. Building on the success of Transformers in natural language processing, their powerful global modeling capabilities have been adapted for semantic segmentation [26–31]. Transformer-based models have introduced self-attention mechanisms and global context modeling, significantly improving segmentation accuracy and adaptability to complex scenes. The method proposed in this paper combines the complementary strengths of CNNs and Transformers for feature extraction, aiming to achieve more comprehensive and precise segmentation of fabric defects by leveraging the local feature modeling of CNNs and the global context capture of Transformers.

2.2. Kolmogorov–Arnold Networks

Recently, Kolmogorov-Arnold Networks (KANs) have emerged as a novel architecture that improves interpretability and accuracy over traditional multilayer perceptrons (MLPs). Researchers have increasingly applied KANs to various computer vision tasks. For example, U-KAN [32] enhances the encoder-decoder structure of U-Net [20] by integrating KAN layers with nonlinear activation functions, boosting feature extraction and improving model accuracy, efficiency, and interpretability. Cambrin et al. [33] applied U-KAN to crop area segmentation and used post-hoc explanation techniques to evaluate saliency maps, highlighting the distinct visual regions emphasized by U-KAN and U-Net during predictions. Similarly, Bodner et al. [34] incorporated KANs' nonlinear activation functions into convolutional layers, creating Convolutional KANs that achieve superior parameter efficiency and offer a novel approach to neural network optimization. In addition, Yang et al. [35] proposed the Kolmogorov-Arnold Transformer (KAT), replacing B-spline functions with rational functions to accelerate computations and substituting MLP layers with Group-Rational KANs within the Transformer architecture. Experiments demonstrate that KAT outperforms MLP-based Transformers across various vision tasks while maintaining similar computational complexity. Building on this, He et al. [36]

introduced MLP-KAN, which combines MLPs and KANs under the Mixture-of-Experts (MoE) framework. This design unifies deep representation learning and function learning, allowing MLP-KAN to dynamically adapt to task characteristics. As a result, MLP-KAN demonstrates superior flexibility and efficiency across diverse domains, optimizing performance based on task-specific needs.

3. Methodology

In this section, we first give a brief introduction to KAN, then present an overview of our proposed approach, and finally the proposed hybrid KAN block will be presented in detail.

3.1. Preliminary

MLPs are one of the fundamental components of current deep learning models, renowned for their ability to approximate complex nonlinear functions through the universal approximation theorem. However, MLPs are not perfect nonlinear regressors. As the number of layers in the network deepens, the internal workings of MLPs become less interpretable, failing to meet the demands for transparency and interpretability in practical applications. Additionally, MLPs can be parameter-heavy, especially in models like Transformers where they dominate the parameter count, leading to efficiency concerns.

As a novel neural network architecture, KANs demonstrate the potential to replace MLPs. KANs are rooted in the Kolmogorov-Arnold representation theorem, which allows for the representation of multivariate functions as compositions of univariate functions and additions. Unlike MLPs, which employ fixed activation functions at the nodes, KANs place learnable activation functions on the edges, parameterized as splines. This innovative design eschews linear weights altogether, with each weight replaced by a learnable univariate function. Compared to MLPs, KANs have higher transparency and interpretability, and they exhibit faster neural scaling laws, allowing them to achieve greater accuracy with fewer parameters. KANs are defined by their core equation, which is an embodiment of the Kolmogorov-Arnold representation theorem. For a function $f(x_1, \dots, x_n)$ defined on a bounded domain, it can be expressed as:

$$f(x) = f(x_1, \dots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^n \phi_{q,p}(x_p) \right) \quad (1)$$

where $\phi_{q,p}$ and Φ_q are continuous univariate functions. This equation is the linchpin of KANs, allowing them to decompose complex functions into a series of simpler, learnable components. This decomposition is not only mathematically elegant but also provides a foundation for KANs' enhanced accuracy and interpretability, as it leverages the compositional nature of functions more effectively than MLPs.

For an input vector \mathbf{x} , the output of the KAN is given by:

$$\text{KAN}(\mathbf{x}) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \dots \circ \Phi_1 \circ \Phi_0)\mathbf{x} \quad (2)$$

where Φ_i represents the set of univariate functions (splines) that are applied to the inputs from the previous layer. The composition \circ indicates that the output of one function set is used as the input to the next.

In summary, while MLPs use fixed activation functions and linear combinations of inputs, the KANs use learnable activation functions that are applied to each input separately and are composed in a multi-layer structure. This allows KANs to capture more complex and compositional structures in the data.

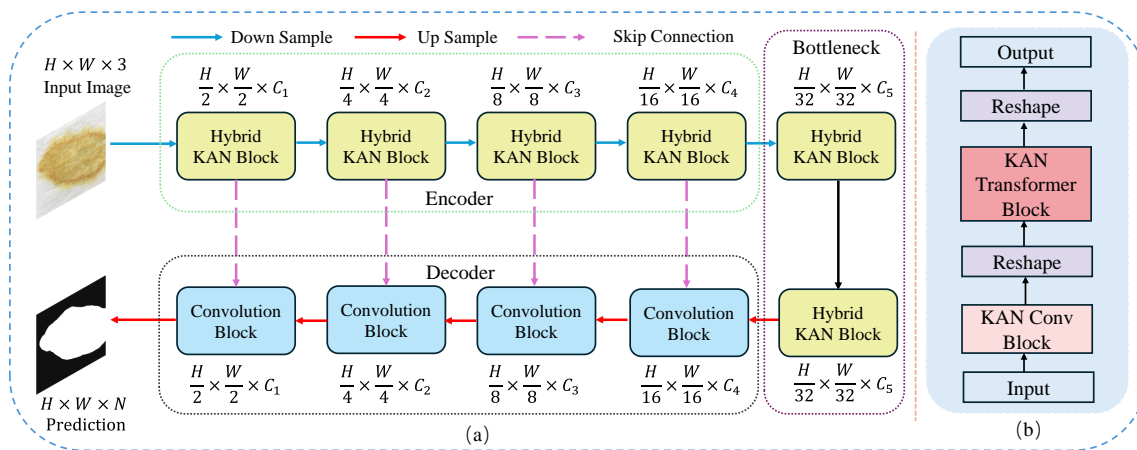


Figure 2. Overview of the HAKN framework. (a) The overall architecture of HKAN consists of three parts: encoder, bottleneck, and decoder. The encoder and bottleneck are composed of Hybrid KAN Blocks, while the decoder utilizes a Convolution Block. (b) Detailed illustration of the Hybrid KAN Block, primarily consisting of KAN Conv Block and KAN Transformer Block.

3.2. Overview of the Proposed Method

The framework of the method proposed in this paper is shown in Figure 2 (a). It is designed similarly to U-Net, consisting of encoder, bottleneck, and decoder. The encoder and bottleneck are composed of multiple cascaded Hybrid KAN Blocks, while the decoder comprises several convolutional blocks. Notably, the Hybrid KAN Block is a novel module we designed, which integrates KAN with CNN and Transformer into a unified framework, as illustrated in Figure 2 (b).

Specifically, for an input image $I \in \mathbb{R}^{H \times W \times C}$ where H , W , and C represent the height, width, and number of channels of the image, respectively. Initially, the input image I is fed into a four-stage encoder to extract features at different levels. During this process, the size of the feature maps sequentially reduces to $\{1/2, 1/4, 1/8, 1/16\}$ of the original image dimensions. The encoder captures progressively more abstract features of the input as the dimensions decrease. In addition, the network incorporates a skip-connection mechanism that helps retain spatial information from higher resolution feature maps by directly connecting layers of similar dimensions. After the encoding phase, the bottleneck layer aggregates the features in order to capture the most critical information, setting the stage for effective feature reconstruction. The upsampling process then initiates, where each upsampling step is paired with convolutional layers to meticulously refine the spatial details of the reconstructed output.

3.3. Hybrid KAN Block

Our proposed Hybrid KAN Block is a brand-new module that integrates KAN into the CNN and Transformer frameworks, forming a unified whole, as shown in Figure 2 (b). Let the input feature map be denoted as $X \in \mathbb{R}^{H \times W \times C}$, the input feature map X first passes through the KAN Convolutional Block:

$$X_{conv} = \text{KANConv}(X) \quad (3)$$

where $X_{conv} \in \mathbb{R}^{H \times W \times 2C}$. The output from the KAN Convolutional Block, X_{conv} is then reshaped to fit the input dimensions required by the Transformer block:

$$X_{reshaped} = \text{Reshape}(X_{conv}) \quad (4)$$

where $X_{reshaped} \in \mathbb{R}^{N \times 2C}$ and $N = H \times W$. This reshaped output is then fed into the KAN Transformer Block to capture long-range dependencies within the feature map:

$$X_{transformed} = \text{KANTransformer}(X_{reshaped}) \quad (5)$$

The output from the Transformer block, $X_{transformed}$, is reshaped back to spatial dimensions:

$$X_{out} = \text{Reshape}(X_{transformed}) \quad (6)$$

where $X_{out} \in \mathbb{R}^{H \times W \times 2C}$. The final output X_{out} is the processed feature map from the Hybrid KAN Block, which integrates both CNN-based local feature extraction and Transformer-based global feature modeling.

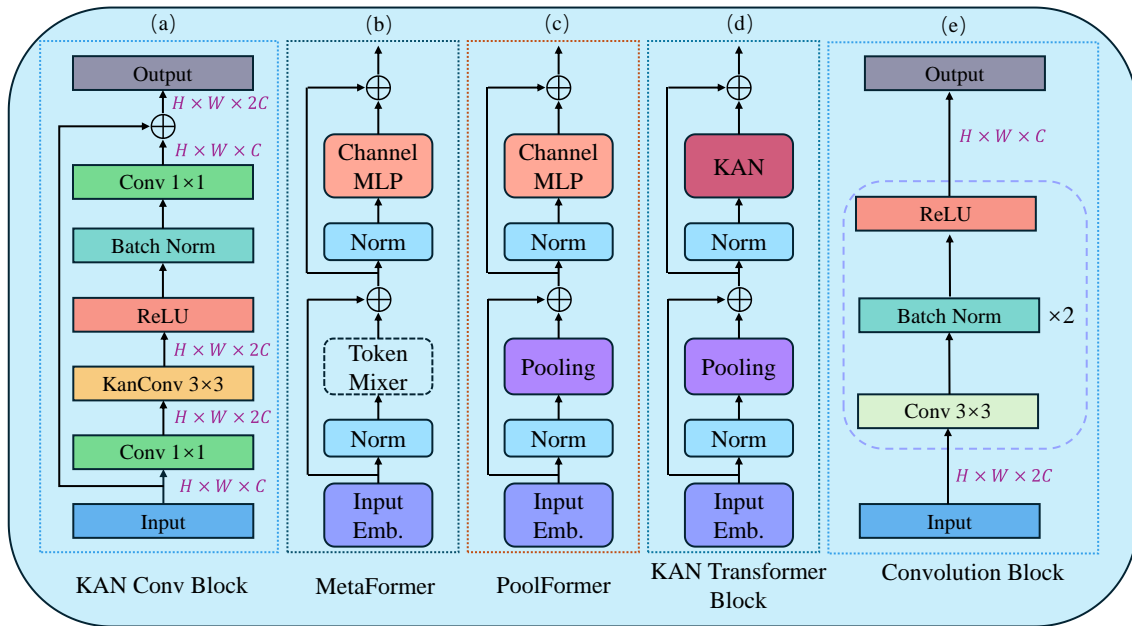


Figure 3. Specific architecture of each module. The KAN Conv Block and KAN Transformer Block are components of the Hybrid KAN Block, and the Convolution Block serves as the decoder.

3.3.1. KAN Conv Block

KAN Convolutions [34] represent an innovative approach to traditional convolutional operations within neural networks, drawing inspiration from the Kolmogorov-Arnold theorem. Unlike typical convolutional networks that utilize fixed activation functions, KAN Convolution employs B-Spline functions which are defined as a linear combination of basis splines. Each element within the convolutional kernel is a B-Spline function:

$$\phi(x) = \sum_i c_i B_i(x) \quad (7)$$

here, $B_i(x)$ are basis spline functions, and c_i are trainable coefficients. These coefficients are adjusted during training, allowing the convolutional layer to mold its activation functions according to the learned data patterns. In KAN Convolution, the application of the kernel to the input feature map involves a unique adaptive process where each kernel element has its own B-Spline activation function. The output feature map at each location is computed as:

$$(\text{Image} * K)_{i,j} = \sum_{k=1}^N \sum_{l=1}^M \phi_{kl}(\text{Image}_{i+k,j+l}) \quad (8)$$

here, K is a KAN convolution kernel of size $N \times M$, ϕ_{kl} are the B-Spline activation functions applied uniquely at each kernel position (k, l) , enriching the convolution operation with the ability to capture a wider range of features through these adaptive nonlinear transformations.

To handle inputs outside the predefined grid range, the grid is dynamically extended. This is formulated as an optimization problem to adjust the control points c'_j of the B-Splines:

$$\{c'_j\} = \arg \min_{c'_j} \mathbb{E}_{x \sim p(x)} \left[\sum_{j=0}^{G2+k-1} c'_j B'_j(x') - \sum_{j=0}^{G1+k-1} c_j B_j(x) \right] \quad (9)$$

where $G1$ is the previous grid size, $G2$ is the new grid size, and k is the degree of the B-Spline.

The overview of our proposed KAN Conv Block is illustrated in Figure. 3 (a). The input feature map F_{in} first undergoes a 1×1 convolution to double the depth of the feature map while maintaining its spatial dimension. After the initial convolution, the feature map F_1 is normalized using batch normalization to stabilize the learning process. The batch-normalized feature map F_2 is then passed through a Rectified Linear Unit (ReLU) activation function to mitigate the gradient vanishing problem. The activated feature map F_3 is subjected to a specialized KAN convolution with a 3×3 kernel. The feature map F_4 is once again processed through a 1×1 convolution to refine the feature map. Finally, the output feature map F_{out} is element-wise added with the original input F_{in} through a skip connection, ensuring that the block contributes enhancements without losing original features. The whole process of KAN Conv Block can be expressed as follows:

$$F_1 = \text{Conv}_{1 \times 1}(F_{in}) \quad (10)$$

$$F_2 = \text{BatchNorm}(F_1) \quad (11)$$

$$F_3 = \text{ReLU}(F_2) \quad (12)$$

$$F_4 = \text{KanConv}_{3 \times 3}(F_3) \quad (13)$$

$$F_5 = \text{Conv}_{1 \times 1}(F_4) \quad (14)$$

$$F_{out} = F_{in} + F_5 \quad (15)$$

where $F_{in}, F_5 \in \mathbb{R}^{H \times W \times C}$ and $F_1, F_2, F_3, F_4, F_{out} \in \mathbb{R}^{H \times W \times 2C}$.

3.3.2. KAN Transformer Block

The Transformer architecture is highly regarded in the field of deep learning due to its ability to capture long-range dependencies. The emergence of Vision Transformer [11] and Swin Transformer [12] has also highlighted the potential for Transformer technology in the field of computer vision. PoolFormer [37] builds on the Transformer by distilling a general architecture called MetaFormer, and it utilizes simple pooling operations as a token mixer, which improves computational efficiency while maintaining strong performance. As seen in Figure. 3, the MLP is an indispensable component of the Transformer architecture, used to enhance the model's capability to process non-linearities. Although the MLP provides the necessary non-linear capabilities, its fixed activation functions and relatively simple network structure may show limitations when dealing with complex or highly non-linear data patterns. Additionally, the standard MLP offers limited flexibility in parameter tuning and model scaling, which may restrict its effectiveness in various tasks and broader applications. Inspired by these insights, this paper replaces the MLP part of the PoolFormer architecture with a KAN. By incorporating KAN, we expect the model to maintain computational efficiency while more accurately capturing and representing complex data features.

The entire process of the KAN Transformer Block is illustrated in Fig. 3 (d). The input image $I \in \mathbb{R}^{H \times W \times C}$ first undergoes an Input embedding:

$$X = \text{InputEmb}(I) \quad (16)$$

where $X \in \mathbb{R}^{N \times C}$ and $N = H \times W$. This step serves to convert the input image I from its original spatial dimensions $\mathbb{R}^{H \times W \times C}$ into a sequence of tokens X . Next, X is fed into repeated KAN Transformer Blocks, each of which contains two residual sub-blocks. Specifically, the first sub-block uses a pooling operation to act as a token mixer, and this process can be represented as:

$$Y = \text{Pooling}(\text{Norm}(X)) + X \quad (17)$$

where $\text{Norm}(\cdot)$ normalization operations, such as Layer Normalization or Batch Normalization. The second residual sub-block replaces the original MLP with a two-layer structured KAN:

$$Z = (\text{Norm}(Y)\Phi_1)\Phi_2 + Y \quad (18)$$

where Φ_1 and Φ_2 denote two layers of learnable activation functions within the KAN.

4. Experiments

In this section, we evaluated the proposed method on three fabric datasets. We began by introducing the datasets, selecting evaluation metrics, and describing the experimental details. Next, we conducted comparative experiments against several advanced models. Additionally, we performed ablation studies to further verify the effectiveness of our method. Finally, we conducted visualization experiments to more intuitively perceive the differences between the various methods.

4.1. Datasets

- **Four Fabric Defects Dataset** : Currently, there is no multi-class semantic segmentation dataset specifically for fabric defects. Therefore, we collect and annotate four types of fabric defects: Hole, Oil, Stain, and ThreadError. The four types of fabric defects consist of a total of 325 images, and image enhancement techniques such as rotation, scaling, and random cropping are used to expand the dataset to 1625 images.
- **Fabric Dataset [37]**: This dataset contains a total of 1600 fabric defect images, which only include defects and the background. The defects in the dataset have weak features and contain false defects, making the detection task quite challenging.
- **ZJU-Leaper Dataset [3]**: This is currently the largest fabric dataset. It consists of 15 types of fabric texture images, which are divided into 4 groups based on the complexity of the background texture. We randomly selected 500 defect images for each of the 15 types of fabric textures, resulting in a total of 7,500 images to be used as the dataset.

The three datasets mentioned above all use images with a size of 512×512 pixels. The ratio of the training set to the test set is 4:1.

4.2. Implementation Details and Evaluation Metrics

- **Implementation Details**: The experiments in this paper are deployed on mmsegmentation. The learning rate is set at 0.01, momentum is at 0.9, using the stochastic gradient descent algorithm, with a weight decay of 0.0005. The maximum number of iterations is set at 40,000, with a batch size of 8, on an RTX 4060Ti GPU.
- **Evaluation Metrics**: We use Pixel Accuracy (PA), Mean Intersection over Union (mIoU), and Mean Dice Coefficient (mDice) as evaluation metrics.

4.3. Comparative Experiments

To validate the effectiveness of the method proposed in this paper, we conducted comparative experiments with six advanced models: U-Net [20], PSPNet [21], DeepLabV3+ [23], Segformer [28], KNet [29], and TransNeXt [38]. The results and analysis of these experiments will be discussed in subsequent sections of the paper.

4.3.1. Results on Four Fabric Defects Dataset

The quantitative results on the Four Fabric Defects Dataset are presented in Table 1. Our method achieves 99.33% PA, 91.60% mIoU, and 95.46% mDice, outperforming all other models. Among the CNN-based methods, U-Net performs the worst, with only 70.59% mIoU, indicating its limitations in handling multi-class fabric defect images. DeepLabV3+ is the best-performing CNN-based model, but our approach surpasses it by 0.81% in mIoU and 1.14% in mDice. For Transformer-based architectures, KNet achieves the highest performance among its peers, outperforming all CNN-based models in PA, mIoU, and mDice. However, our method still exceeds KNet, with a 0.74% improvement in mIoU. These results demonstrate the superior capability of our approach in addressing complex multi-class fabric defect images, leveraging the combined strengths of CNNs and Transformers to outperform models based purely on either architecture.

Table 1. Quantitative Comparisons to SOTA methods on the Four Fabric Defects Dataset

Method	PA (%)	mIoU (%)	mDice (%)
UNet [20]	96.57	70.59	83.08
PSPNet [21]	98.94	87.60	93.20
DeepLabV3+ [23]	99.08	89.59	94.32
Segformer [28]	99.21	90.79	94.99
KNet [29]	99.26	90.86	95.01
TransNeXt [38]	99.16	89.87	94.46
HKAN	99.33	91.60	95.46

4.3.2. Results on Fabric Dataset

The quantitative results on the Fabric Dataset are shown in Table 2. Our method achieves the highest performance in terms of mIoU and mDice, reaching 70.41% and 79.08%, respectively. For PA, our method achieves 99.68%, slightly trailing PSPNet and Segformer by only 0.02%. The weak defect features in this dataset, coupled with their low contrast against the background texture, result in generally low mIoU scores across all models. Moreover, the performance gap between CNN-based and Transformer-based methods is minimal, indicating that neither architecture alone is sufficient to fully address the challenges posed by this dataset. These results suggest that there is still considerable room for improvement in fabric defect segmentation, particularly in enhancing the ability to distinguish subtle defects from complex background textures.

Table 2. Quantitative Comparisons to SOTA methods on the Fabric Dataset

Method	PA (%)	mIoU (%)	mDice (%)
UNet [20]	99.67	64.14	72.16
PSPNet [21]	99.70	67.33	75.83
DeepLabV3+ [23]	99.08	68.24	76.82
Segformer [28]	99.70	69.74	78.38
KNet [29]	99.69	69.35	77.99
TransNeXt [38]	99.69	69.57	78.21
HKAN	99.68	70.41	79.08

4.3.3. Results on ZJU-Leaper Dataset

The quantitative results for Group 1 and Group 2 in the ZJU-Leaper Dataset are shown in Table 3. Our method achieved the best results in both tissue datasets. In Group 1, the fabric background texture is single and clear, and the fabric defects are not complex, so the differences between the models are not easily noticeable, and all achieve good detection results. As the background texture becomes slightly more complex and the fabric defects become more diverse, the detection performance of different models shows significant differences. Our method achieved a mIoU in Group 2 that was 4.42% higher than UNet and 0.14% higher than the best-performing model, DeepLabV3+. In Groups 3 and 4, the fabric images contain complex fabric textures and diverse fabric defects. Our method achieved a PA in Group 3 that was only 0.01% lower than PSPNet, and the mDice reached 93.36% and 94.01% in Groups 3 and 4, outperforming other models. The experimental results indicate that our method can accurately capture image details when facing simple fabric defect images and still maintains strong competitiveness when dealing with complex fabric defect images. This is attributed to the KAN Transformer Block proposed in our method, which has the ability to handle long-distance dependencies.

Table 3. Quantitative comparison with the SOTA method on Group1 and Group2 of the ZJU-Leaper dataset

Model	Group1			Group2		
	PA (%)	mIoU (%)	mDice (%)	PA (%)	mIoU (%)	mDice (%)
U-Net [20]	97.26	83.17	90.18	96.69	81.46	89.02
PSPNet [21]	97.32	84.81	91.30	97.11	83.32	90.30
DeepLabV3+ [23]	97.65	85.73	91.88	97.53	85.74	91.90
Segformer [28]	97.75	86.40	92.32	97.45	85.45	91.71
KNet [29]	97.73	86.49	92.38	97.47	85.42	91.69
TransNeXt [38]	97.57	85.45	91.71	97.18	83.92	90.71
HKAN	97.77	86.63	92.46	97.53	85.88	91.99

Table 4. Quantitative comparison with the SOTA method on Group3 and Group4 of the ZJU-Leaper dataset

Model	Group3			Group4		
	PA (%)	mIoU (%)	mDice (%)	PA (%)	mIoU (%)	mDice (%)
U-Net [20]	97.60	82.57	89.72	96.21	80.47	88.36
PSPNet [21]	98.44	87.99	93.28	97.96	88.81	93.84
DeepLabV3+ [23]	98.33	87.43	92.94	97.91	88.67	93.75
Segformer [28]	98.39	87.67	93.09	97.89	88.44	93.61
KNet [29]	98.19	86.45	92.31	97.68	87.81	93.23
TransNeXt [38]	98.21	85.80	92.54	97.61	87.27	92.89
HKAN	98.43	88.10	93.36	97.98	89.10	94.01

4.3.4. Visualization Results

We displayed intuitive detection results on the Four Fabric Defects Dataset in Figure 4. From Figure 4, it is evident that for detecting simple Holes, most models perform well except for UNet. When detecting Oil, both KNet and TransNeXt suffer from false positives. In the case of Stain detection, DeepLabV3+, Sgeformer, and KNet all exhibit minor false positives in specific areas, whereas our method does not. Finally, the detection results of ThreadError show that our method provides clearer detection outcomes.

Figure 5 shows the intuitive detection results on the Fabric Dataset. As can be seen from Figure 5, the detection performance of the three CNN-based models is weaker than that of the Transformer-

based models. TransNeXt, which is a Transformer model, can detect the creases on the right side as shown in the third row of Figure 5, whereas Segformer and KNet cannot. Our method demonstrates superior detection performance compared to TransNeXt, with clearer and more complete boundaries. This can be observed more clearly in the second and third rows of Figure 5.

Figure 6 presents the intuitive detection results from our experiments on the ZJU-Leaper dataset. It is apparent from the image that our method significantly outperforms the compared models. Our model is able to accurately capture the boundaries of the targets, demonstrating excellent detail depiction capabilities even in complex scenes. In the presence of significant background interference, our model exhibits stronger robustness, effectively reducing areas of incorrect classification, while other models are somewhat lacking in this aspect. Additionally, our model performs more consistently in segmenting large-scale targets, mitigating issues such as fragmentation or insufficient coherence that are commonly observed in models like Segformer and KNet.

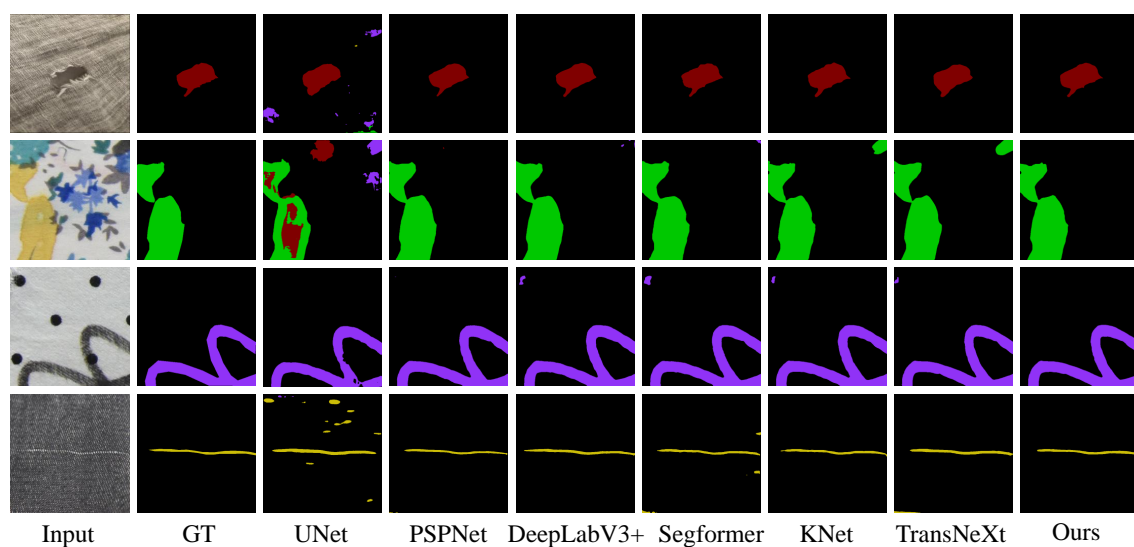


Figure 4. Intuitive visualization results on the Four Fabric Defects Dataset.

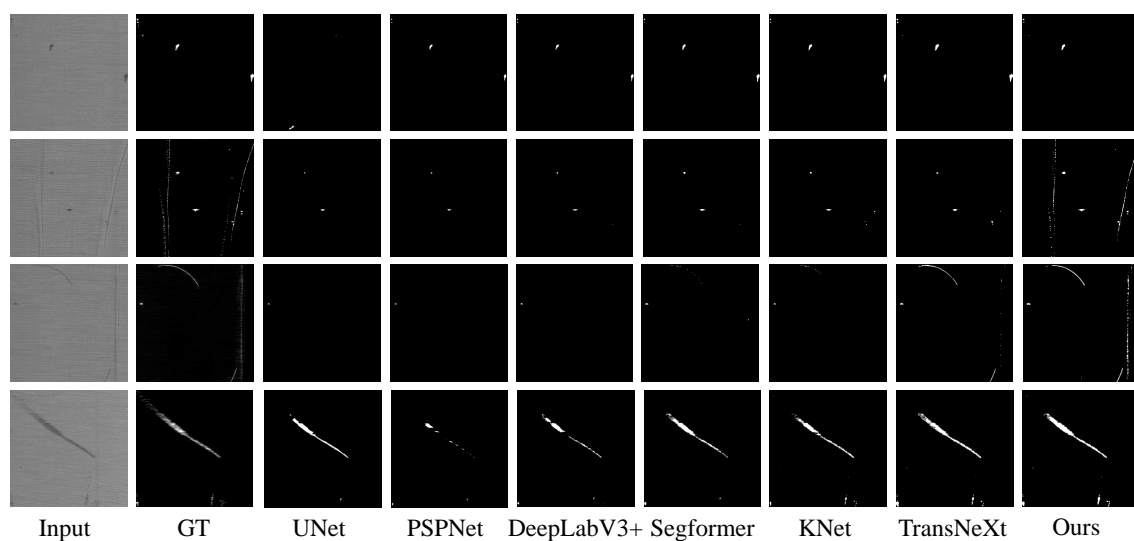


Figure 5. Intuitive visualization results on the Fabric Dataset.

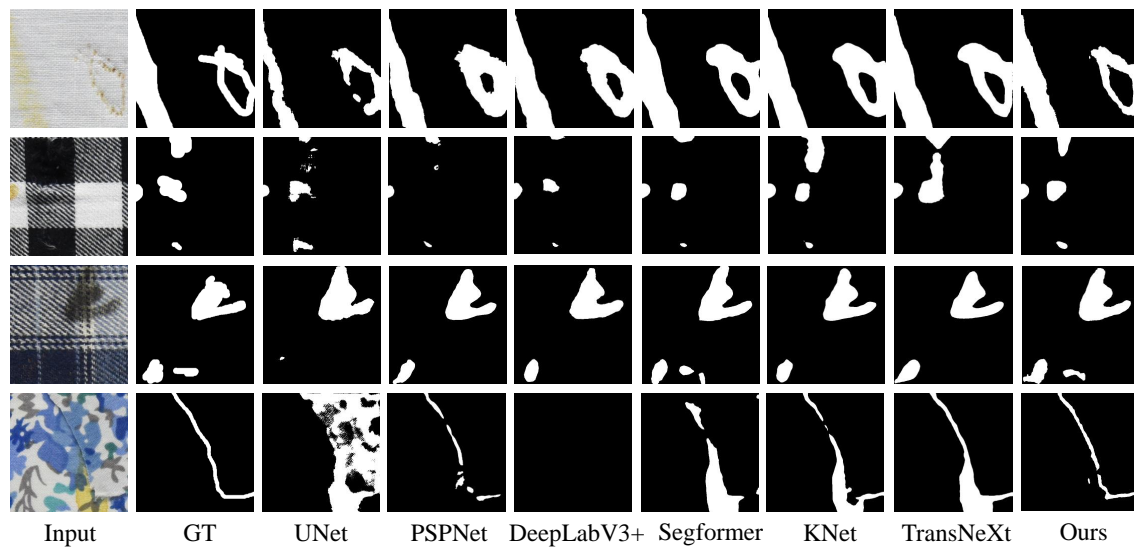


Figure 6. Intuitive visualization results on the ZJU-Leaper Dataset.

Across diverse scenarios, our approach adapts well to various inputs, consistently producing high-quality segmentation results.

4.4. Ablation Experiments

To further verify the effectiveness of the proposed module in this paper, we conducted ablation experiments on the Four Fabric Defects Dataset. We selected the UNet based on pure CNN as the baseline method and sequentially chose PoolFormer Block, KANConv Block, KANTransformer Block, and Hybrid KAN Block as the Encoders. The quantitative experimental results are shown in Table 5. From Table 5, it can be seen that replacing the convolutional modules in UNet with KANConv can significantly improve the detection performance, but there is still a certain gap with mainstream models. Using the PoolFormer Block and KAN Transformer Block as Encoders can further improve the detection performance of the model, but it still does not reach the level of SOTA models. Using the entire Hybrid KAN Block as the Encoder, the detection performance is close to that of Segformer and KNet, which perform better. From the above experimental results and analysis, it can be seen that the KANConv Block, KANTransformer Block, and Hybrid KAN Block proposed in this paper play different roles to collectively enhance the model's feature extraction ability, fully combining the advantages of each model.

Table 5. Ablation experiment results on the Four Fabric Defects Dataset.

Method	PA (%)	mIoU (%)	mDice (%)
CNN Encoder	96.57	70.59	83.08
Transformer Encoder	98.69	85.28	91.66
KANConv Encoder	98.09	81.60	89.15
KANTransformer Encoder	98.93	87.52	93.10
Hybrid KAN Encoder	99.26	90.84	95.02
HKAN	99.33	91.60	95.46

5. Conclusions

This paper proposes a fabric defect segmentation network called HKAN, which integrates CNNs, Transformers, and KANs into a unified framework, demonstrating significant potential for the fabric defect segmentation task. Extensive experiments were conducted on three fabric datasets, and the

results indicate that HKAN can achieve excellent segmentation outcomes for both simple and complex fabric images, outperforming models based solely on CNNs or Transformers. In the future, the authors plan to further explore the possibilities of combining these three types of models to leverage their respective strengths in different scenarios.

Author Contributions: Conceptualization, M.L. and P.Y.; methodology, M.L.; software, M.L. and P.Y.; validation, M.L., P.Y.; formal analysis, S.C. and P.Z.; investigation, M.L., P.Y.; resources, J.L.; data curation, S.C. and P.Z.; writing—original draft preparation, P.Y.; writing—review and editing, M.L.; visualization, P.Y.; supervision, M.L.; project administration, M.L. and P.Y.; funding acquisition, M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work supported by Educational Commission of Hubei Province of China(NO.D20211701).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The links to the Fabric Dataset and ZJU-Leaper Dataset used in the paper are: <https://pan.baidu.com/share/init?surl=nA2gIhL35iSsbMstkEugfg&pwd=0927> and <http://www.qaas.zju.edu.cn/zju-leaper/>. If you are interested in the Four Fabric Defects Dataset used in this paper, please send an email to ye-pei@outlook.com and state the purpose.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ngan, H.Y.; Pang, G.K.; Yung, N.H. Automated fabric defect detection—A review. *Image and vision computing* **2011**, *29*, 442–458.
2. Rasheed, A.; Zafar, B.; Rasheed, A.; Ali, N.; Sajid, M.; Dar, S.H.; Habib, U.; Shehryar, T.; Mahmood, M.T. Fabric defect detection using computer vision techniques: a comprehensive review. *Mathematical Problems in Engineering* **2020**, *2020*, 8189403.
3. Zhang, C.; Feng, S.; Wang, X.; Wang, Y. Zju-leaper: A benchmark dataset for fabric defect detection and a comparative study. *IEEE Transactions on Artificial Intelligence* **2020**, *1*, 219–232.
4. Zhao, S.; Yin, L.; Zhang, J.; Wang, J.; Zhong, R. Real-time fabric defect detection based on multi-scale convolutional neural network. *IET Collaborative Intelligent Manufacturing* **2020**, *2*, 189–196.
5. Huang, Y.; Jing, J.; Wang, Z. Fabric defect segmentation method based on deep learning. *IEEE Transactions on Instrumentation and Measurement* **2021**, *70*, 1–15.
6. Koulali, I.; Eskil, M.T. Unsupervised textile defect detection using convolutional neural networks. *Applied Soft Computing* **2021**, *113*, 107913.
7. Chen, M.; Yu, L.; Zhi, C.; Sun, R.; Zhu, S.; Gao, Z.; Ke, Z.; Zhu, M.; Zhang, Y. Improved faster R-CNN for fabric defect detection based on Gabor filter with Genetic Algorithm optimization. *Computers in Industry* **2022**, *134*, 103551.
8. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence* **2016**, *39*, 1137–1149.
9. Lu, B.; Huang, B. A texture-aware one-stage fabric defect detection network with adaptive feature fusion and multi-task training. *Journal of Intelligent Manufacturing* **2024**, *35*, 1267–1280.
10. Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems* **2017**.
11. Dosovitskiy, A. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* **2020**.
12. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF international conference on computer vision, 2021*, pp. 10012–10022.
13. Qu, H.; Di, L.; Liang, J.; Liu, H. U-SMR: U-SwinT & multi-residual network for fabric defect detection. *Engineering Applications of Artificial Intelligence* **2023**, *126*, 107094.
14. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition, 2016*, pp. 770–778.
15. Xu, H.; Liu, C.; Duan, S.; Ren, L.; Cheng, G.; Hao, B. A Fabric Defect Segmentation Model Based on Improved Swin-Unet with Gabor Filter. *Applied Sciences* **2023**, *13*, 11386.

16. Cao, H.; Wang, Y.; Chen, J.; Jiang, D.; Zhang, X.; Tian, Q.; Wang, M. Swin-unet: Unet-like pure transformer for medical image segmentation. *European conference on computer vision*. Springer, 2022, pp. 205–218.
17. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural networks* **1989**, *2*, 359–366.
18. Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T.Y.; Tegmark, M. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756* **2024**.
19. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.
20. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.
21. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.
22. Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; Sun, J. Unified perceptual parsing for scene understanding. *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 418–434.
23. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.
24. Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; Agrawal, A. Context encoding for semantic segmentation. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7151–7160.
25. Kirillov, A.; Girshick, R.; He, K.; Dollár, P. Panoptic feature pyramid networks. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6399–6408.
26. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P.H.; others. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6881–6890.
27. Strudel, R.; Garcia, R.; Laptev, I.; Schmid, C. Segmenter: Transformer for semantic segmentation. *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 7262–7272.
28. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems* **2021**, *34*, 12077–12090.
29. Zhang, W.; Pang, J.; Chen, K.; Loy, C.C. K-net: Towards unified image segmentation. *Advances in Neural Information Processing Systems* **2021**, *34*, 10326–10338.
30. Cheng, B.; Schwing, A.; Kirillov, A. Per-pixel classification is not all you need for semantic segmentation. *Advances in neural information processing systems* **2021**, *34*, 17864–17875.
31. Cheng, B.; Misra, I.; Schwing, A.G.; Kirillov, A.; Girdhar, R. Masked-attention mask transformer for universal image segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1290–1299.
32. Li, C.; Liu, X.; Li, W.; Wang, C.; Liu, H.; Liu, Y.; Chen, Z.; Yuan, Y. U-kan makes strong backbone for medical image segmentation and generation. *arXiv preprint arXiv:2406.02918* **2024**.
33. Rege Cambrin, D.; Poeta, E.; Pastor, E.; Cerquitelli, T.; Baralis, E.; Garza, P. KAN You See It? KANs and Sentinel for Effective and Explainable Crop Field Segmentation. *arXiv e-prints* **2024**, pp. arXiv–2408.
34. Bodner, A.D.; Tepsich, A.S.; Spolski, J.N.; Pourteau, S. Convolutional Kolmogorov-Arnold Networks. *arXiv preprint arXiv:2406.13155* **2024**.
35. Yang, X.; Wang, X. Kolmogorov-arnold transformer. *arXiv preprint arXiv:2409.10594* **2024**.
36. He, Y.; Xie, Y.; Yuan, Z.; Sun, L. MLP-KAN: Unifying Deep Representation and Function Learning. *arXiv preprint arXiv:2410.03027* **2024**.
37. Wang, J.; Xu, G.; Li, C.; Gao, G.; Wu, Q. Sddet: An enhanced encoder-decoder network with hierarchical supervision for surface defect detection. *IEEE Sensors Journal* **2022**, *23*, 2651–2662.
38. Shi, D. TransNeXt: Robust Foveal Visual Perception for Vision Transformers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17773–17783.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.