INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2026, VOL. 72, NO. 2, PP. 1–8
Manuscript received February 20, 2026; revised April 2026. doi: 10.24425/ijet.2026.157931

Robustness Analysis of CNN and Convolutional KAN Architectures against Weather and Geometric Distortions in Traffic Sign Recognition

Urszula Libal, Pawel Biernacki, and Szymon Sołtysiak

***Abstract*—This paper investigates the robustness of traffic sign classification models against real-world visual disturbances. We conduct a comparative evaluation of three distinct architectures: a standard CNN, a hybrid CNN enhanced with Kolmogorov-Arnold dense layers (CNN-KAN), and a fully convolutional Kolmogorov-Arnold Network (CKAN). Unlike traditional CNNs, the KA-based models utilize learnable activation functions, potentially offering improved resilience. The experiments were conducted using the German Traffic Sign Recognition Benchmark (GTSRB) dataset, containing 43 classes of traffic signs. Models were trained and tested on both original images and versions degraded by controlled disturbances, including rotation, blur, brightness variation, and simulated rain. The results demonstrate that the proposed CNN-KAN model provides consistently superior performance under small-to-moderate rotations (up to 20 degrees) and moderate brightness increases, achieving the highest accuracy in all rain-mask scenarios. It remains competitive under blur, where it ranks second only to the standard CNN. Performance decreases were observed only at extreme brightness levels, where both the standard CNN and CKAN maintained higher stability. Overall, the findings highlight the potential of Kolmogorov-Arnold-based architectures for improving robustness in traffic sign recognition systems operating under realistic and dynamically changing environmental conditions.**

***Keywords*—Convolutional Neural Networks (CNN); Kolmogorov-Arnold Networks (KAN); traffic sign recognition; robustness; visual disturbances; weather conditions; data augmentation; deep learning**

I. Introduction

CONVOLUTIONAL Neural Networks (CNNs) have long served as the backbone of modern computer vision, achieving remarkable success in tasks such as traffic sign recognition. However, traditional Multi-Layer Perceptrons (MLPs)–the standard classifiers within these architectures–rely on fixed activation functions and linear weight aggregation. This rigid structure can limit their ability to adaptively model complex, non-linear feature landscapes. Recently, Kolmogorov-Arnold Networks (KANs) have emerged as a promising alternative. By replacing fixed scalar weights with learnable univariate functions (typically splines) on the edges, KANs offer a potentially more expressive and parameter-efficient mechanism for function approximation.

In this study, we investigate whether integrating the adaptive nature of Kolmogorov-Arnold layers into convolutional architectures enhances robustness against visual disturbances. We propose and evaluate three distinct architectures applied to the traffic sign recognition problem:

- **CNN:** A classical baseline architecture using standard convolutional layers and a fixed MLP classifier.
- **CNN-KAN:** A hybrid architecture that retains standard feature extraction but replaces the classification head with **Kolmogorov-Arnold dense layers**, leveraging their learnable activations for decision-making.
- **CKAN** [1]: A fully integrated approach that incorporates **Kolmogorov-Arnold transformations directly into the convolutional stage**, effectively creating non-linear kernels for spatial feature extraction.

Beyond standard accuracy metrics, real-world deployment requires resilience to environmental variability. To assess the stability of these learnable-function architectures, we conduct a comprehensive performance analysis using the German Traffic Sign Recognition Benchmark (GTSRB). We evaluate model robustness against a spectrum of synthetic perturbations, including rain masks, varying brightness levels, Gaussian blur, and geometric rotation.

The remainder of this paper is organized as follows. Section II outlines the theoretical foundations of Kolmogorov-Arnold Networks and details the construction of the proposed hybrid architectures. Section III describes the dataset and the specific data augmentation strategies employed to simulate real-world conditions. Section IV presents the experimental evaluation, offering a comparative analysis of model parameter efficiency and robustness against environmental corruptions. Finally, Section V summarizes the key findings and concludes the study.

II. Kolmogorov-Arnold Convolutional Network Architectures

The versatility of Kolmogorov-Arnold networks (KAN) has driven the development of diverse architectures across multiple fields. Building upon this growing body of research, this paper

U. Libal and P. Biernacki are with the Wroclaw University of Science and Technology, Faculty of Electronics, Photonics and Microsystems, Poland (e-mail: urszula.libal@pwr.edu.pl, ORCID: 0000-0002-3348-510X; pawel.biernacki@pwr.edu.pl, ORCID: 0000-0002-0818-5981).

S. Sołtysiak is with the Wroclaw University of Science and Technology, Faculty of Mathematics, Poland (e-mail: szymon.soltysiak.research@gmail.com, ORCID: 0009-0001-2054-2857).

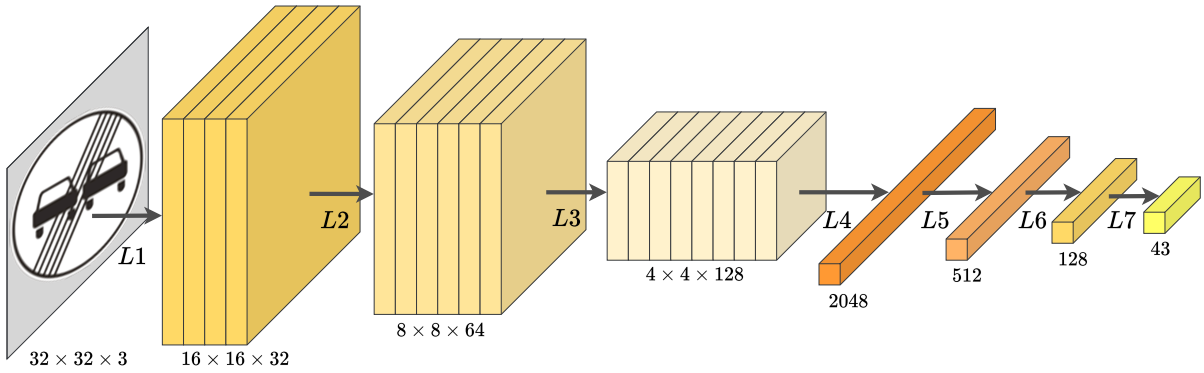

Fig. 1. The three network models share a common general topology but differ in layer implementation: (a) **CNN**: Standard convolutional backbone (L1–L3) with a dense MLP classifier (L5–L7). (b) **CNN-KAN**: Standard convolutional backbone (L1–L3) with a KAN-based classifier (L5–L7). (c) **CKAN**: Hybrid backbone introducing a KA-convolution at L3, followed by a hybrid MLP/KAN classifier.

proposes the CNN-KAN architecture, a hybrid model designed to leverage the feature extraction power of CNNs with the classification capabilities of KANs.

A. Fundamental Components

To understand the proposed hybrid architectures, it is necessary to first define the distinct operational mechanisms of standard convolutional layers, KA dense layers, and KA convolutional layers.

1) Standard Convolutional Layers:
The standard Convolutional Neural Network (CNN) remains the dominant architecture for image classification due to its ability to learn hierarchical feature representations. Mathematically, a standard convolutional layer transforms an input volume into an output feature map through a discrete linear convolution. For an input I and a learnable kernel K of dimensions $m \times n$, the value of the output feature map S at position (i, j) is calculated as:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) \cdot K(m,n). \tag{1}$$

In this operation, the kernel K consists of fixed scalar weights. The kernel slides over the input, computing a linear dot product followed by a static non-linear activation function (such as ReLU).

2) Kolmogorov-Arnold (KA) Dense Layers:
Kolmogorov-Arnold layers replace the fixed activation functions of standard Multilayer Perceptrons (MLP) with learnable non-linear functions on every edge [2]. This is based on the Kolmogorov-Arnold theorem, which posits that multivariate functions can be represented as compositions of univariate functions.

Unlike a standard neuron connection defined by a static scalar weight and fixed activation function, a KA connection learns the activation function itself. The operation is defined as:

$$f(x) = \sum_i (\alpha_i \cdot B_i(\beta_i \cdot x + \gamma_i)), \tag{2}$$

where $B_i(x)$ represents a basis function (typically a B-spline), α_i is a coefficient, and β_i and γ_i are trainable weights.

3) Kolmogorov-Arnold Convolutional Layers:
The Convolutional KAN (CKAN) adapts the KAN principle for spatial data [1]. While it retains the sliding window structure of a standard CNN to preserve spatial locality, the internal mathematical operation is fundamentally different.

In a KA-convolution, the kernel elements are not scalar weights, but learnable non-linear functions ϕ_{ij}. The kernel matrix K is expressed as:

$$K = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{bmatrix} \tag{3}$$

The operation applied to a pixel x is a sum of non-linear functions:

$$y = \sum \phi_{ij}(x_{ij}).$$

This increased mathematical complexity per kernel operation allows for extracting more complex features but contributes to a higher parameter count compared to standard convolutions.

B. Proposed Network Architectures

This study evaluates three distinct architectures: a baseline CNN, the proposed CNN-KAN, and a CKAN variant. As illustrated in Fig. 1, all three models share a common topological framework consisting of a feature extractor (Layers L1–L4) and a classification head (Layers L5–L7). The distinctions are as follows:

1) CNN:
The baseline model employs a traditional feature extractor composed of three convolutional layers (L1–L3). Each layer utilizes a 3×3 kernel (stride 1, padding 1), followed by batch normalization, ReLU activation, and 2×2 max-pooling. The classification head consists of an input flattening layer (L4) followed by three standard fully connected layers (L5–L7) with 2048, 512, and 128 neurons respectively, ending in a 43-neuron output layer.

2) CNN-KAN:
This hybrid architecture retains the identical standard convolutional backbone (L1–L3) used in the baseline. However, it

replaces the standard dense layers in the classification head (L5–L7) with Kolmogorov-Arnold dense layers. This design tests the hypothesis that KANs offer superior classification performance when processed features are extracted via standard linear convolutions.

3) CKAN:
This architecture introduces non-linearity into the spatial feature extraction itself. While L1 and L2 remain standard convolutional layers, layer L3 is replaced by a **KA convolutional layer**. Consequently, the network extracts higher-level spatial features using non-linear kernels before flattening. The classification head in this variant is also a hybrid, combining a standard dense layer (L5) with subsequent KA dense layers (L6 and L7).

III. Experimental Setup

The experiments utilized the publicly available German Traffic Sign Recognition Benchmark (GTSRB) dataset [3]. Firstly, GTSRB provides a comprehensive and realistic collection of traffic sign images captured under diverse real-world conditions. The dataset's challenging nature–due to unbalanced class frequencies, small object sizes, and complex backgrounds–drives the development of advanced machine learning and deep learning techniques, such as convolutional neural networks [4]–[8], attention mechanisms [9], [10], and transfer learning [11], [12].

This dataset comprises 43 distinct traffic sign classes with an uneven class distribution, containing a total of 39,209 training images and 12,630 test images. Preprocessing involved resizing inputs to 32×32 pixels, followed by tensor conversion and normalization. Representative samples of these RGB images are displayed in Fig. 2.

Fig. 2. Sample images from the GTSRB collection.

We adopted a 90/10 split for training and validation data. The training process utilized the Adam optimizer and Cross-Entropy loss function, with data fed in batches of 64 over 10 epochs. To ensure training stability, we monitored loss after every batch and assessed validation accuracy at the end of each epoch. Post-training, the final model quality was verified using the test set.

A. Data Augmentation

We evaluated the models' robustness against image degradation by introducing four specific noise types: rotation (Fig. 3), rain (Fig. 4), blur (Fig. 5), and brightness shifts (Fig. 6). The intensity of each disturbance was determined by sampling governing parameters–specifically rotation angle, rain coverage, blur radius, and brightness coefficient–from predefined probability distributions.

Fig. 3. Rotations.

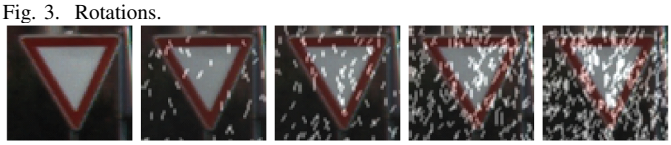

Fig. 4. Rain masks.

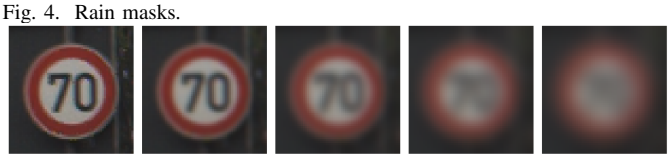

Fig. 5. Blur.

Fig. 6. Brightness changes.

Fig. 7. Examples of different image disturbances applied to GTSRB database for training and testing of neural network models.

The parameters for the noise distributions were selected to balance realistic environmental variability with extreme stress-testing scenarios:

- The **rotation angle** ϕ was sampled from a zero-mean Normal distribution
$$\phi \sim \mathcal{N}(0, \sigma^2)$$
with standard deviation σ. The standard deviation $\sigma = 60°$ was chosen based on the properties of the Normal distribution, where approximately 99.7% of samples lie within $\pm 3\sigma$. Since $3 \times 60° = 180°$, this parameterization ensures that the generated disturbances effectively cover the full range of possible rotations $[-180°, 180°]$, while maintaining a higher sampling density around the canonical upright orientation ($0°$).
- The **rain coverage factor** α was generated using an exponential distribution
$$\alpha \sim \text{Exp}(1/3)$$
with a scale parameter $1/3$, implying a mean coverage of approximately 33% of an image with the rain mask.
- Similarly, the **blur radius** r follows an exponential distribution
$$r \sim \text{Exp}(8)$$
with a scale parameter 8. An Exponential distribution was employed for both rain coverage and blur radius to reflect real-world conditions where mild disturbances are frequent, and the random variable takes only non-negative values, as it is for rain coverage factor $\alpha \geq 0$ or blur radius $r \geq 0$. For blur, the scale parameter of

8 ensures the model is tested against significant loss of high-frequency edge details.

- Finally, the **brightness factor** v was sampled from a Gamma distribution

$$v \sim \Gamma(k, \theta)$$

with a shape parameter $k = 2$ and a scale parameter $\theta = 1$. The Gamma distribution $\Gamma(2, 1)$ is particularly suitable for multiplicative intensity factors because its mode (peak probability) occurs at $(k-1)\theta = (2-1)\cdot 1 = 1.0$. This implies that the most frequent samples are close to the original image brightness ($v = 1$), preserving semantic content. However, the distribution is right-skewed with a mean of 2.0, providing rigorous testing against overexposure and glare conditions.

IV. Results and Discussion

In this section, we present the traffic sign recognition results on the GTSRB database across five distinct experimental settings. These include a baseline trained on the original, undistorted dataset, and four separate models trained on datasets augmented with specific noise types: rotation, rain mask coverage, blur, and brightness variations. The details of these augmentations are described in the previous section.

In Table I, we compare the highest achievable accuracy for each model configuration. Since the peak performance was invariably observed on clean (undistorted) test data, this table effectively illustrates the trade-off cost (i.e., the drop in clean-data accuracy) incurred when training for robustness. As expected, the highest accuracy in every experiment–even for models trained on distorted data–was obtained on undistorted images. Specifically, performance peaked at 0° rotation, zero rain coverage ($\alpha = 0$), zero blur radius ($r = 0$), and standard brightness ($v = 1$).

TABLE I
Comparison of peak accuracy [%] achieved on clean test data across different training settings. The 'No Disturbance' row represents the baseline performance, while the subsequent rows show the clean-data accuracy of models trained with specific disturbance augmentations. Best results in each row in bold font.

Training Setting	Model		
	CNN	CNN-KAN	CKAN
No Disturbance	95.11	**97.70**	93.12
Rotation	89.94	**92.28**	86.38
Rain	85.44	**93.96**	78.99
Blur	88.72	**94.49**	91.32
Brightness	92.93	**95.82**	94.46

The results highlight two key observations: The baseline training setting demonstrates the theoretical maximum performance of the architectures, with CNN–KAN achieving the best overall results (see Table I). Training for robustness on distorted images generally impacts precision. The inclusion of rain data caused the most significant drop in clean accuracy for the standard CNN ($\approx 10\%$ decrease). Conversely, brightness augmentation proved beneficial for the CKAN model, improving its baseline accuracy from 93.12% to 94.46%.

A. Model Size Comparison

Table II compares the spatial complexity of the three architectures. The baseline CNN is the most lightweight model, containing approximately 1.2 million parameters and requiring only $4,753$ kB of memory. In stark contrast, the CNN–KAN exhibits a nearly ten-fold increase in size, with over 11.2 million parameters and a memory footprint of $44,113$ kB. The CKAN architecture occupies a middle ground; despite utilizing complex Kolmogorov-Arnold convolutions, it maintains a parameter count of 2.3 million ($9,017$ kB), which is approximately double that of the standard CNN but significantly smaller than the CNN–KAN.

TABLE II
Memory requirement for different network architectures

Network Model	Size [kB]	No. of parameters
CNN	4,753	1,213,995
CNN–KAN	44,113	11,289,856
CKAN	9,017	2,304,014

B. Robustness to Rotation

The analysis of the experimental results demonstrates a critical sensitivity of the standard models to rotational disturbances. As shown in the accuracy curves in Fig. 8, the baseline models (CNN, CNN–KAN, and CKAN) lack inherent rotational invariance. While they achieve high accuracy at the canonical orientation ($\phi = 0^\circ$)—with CNN–KAN reaching a peak of 97.70%, followed by CNN (95.11%) and CKAN (93.12%)—their performance degrades catastrophically as the rotation angle deviates from zero. At moderate rotations of $\pm 30^\circ$, the accuracy for all baseline models drops by approximately 50 percentage points. Furthermore, at extreme angles ($|\phi| > 60^\circ$), the models fail to generalize entirely, with accuracy falling below 20%, which is comparable to random guessing on the GTSRB dataset.

Incorporating rotational data augmentation significantly enhances model robustness. The models trained with rotation (denoted with the suffix _rotation) exhibit substantially flatter accuracy curves, maintaining functional performance across the entire angular spectrum $[-180^\circ, 180^\circ]$. Most notably, at the complete inversion point of $\phi = 180^\circ$, the rotation-trained models maintain an accuracy between 74% and 79%, representing a massive improvement over the $\approx 25\%$ baseline of the standard models. This indicates that the networks successfully learned to extract rotation-invariant features when provided with appropriate training examples.

However, this increased robustness introduces a noticeable trade-off in peak performance. The maximum accuracy at $\phi = 0^\circ$ is consistently lower for the rotation-trained variants compared to their standard counterparts. For instance, the peak accuracy of the CNN–KAN model decreases from 97.70%

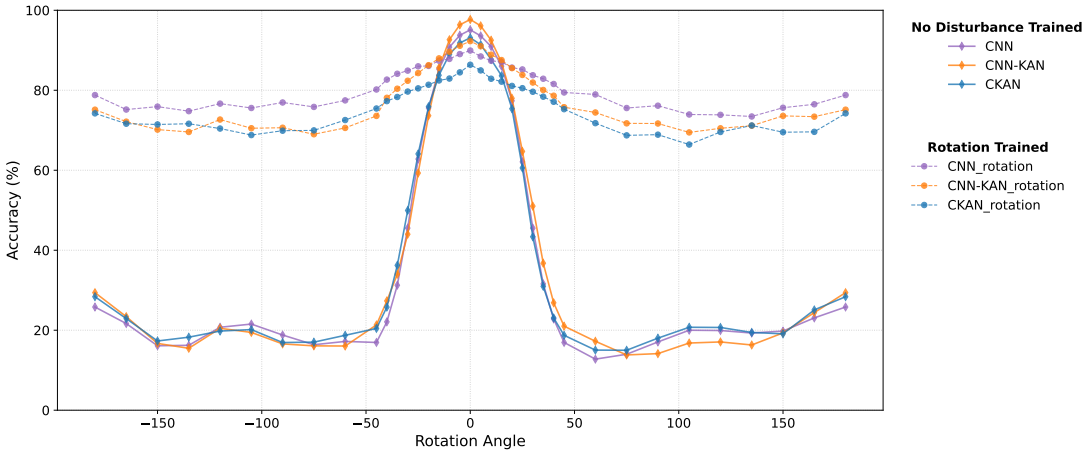

Fig. 8. Accuracy depending on the rotation angle ϕ.

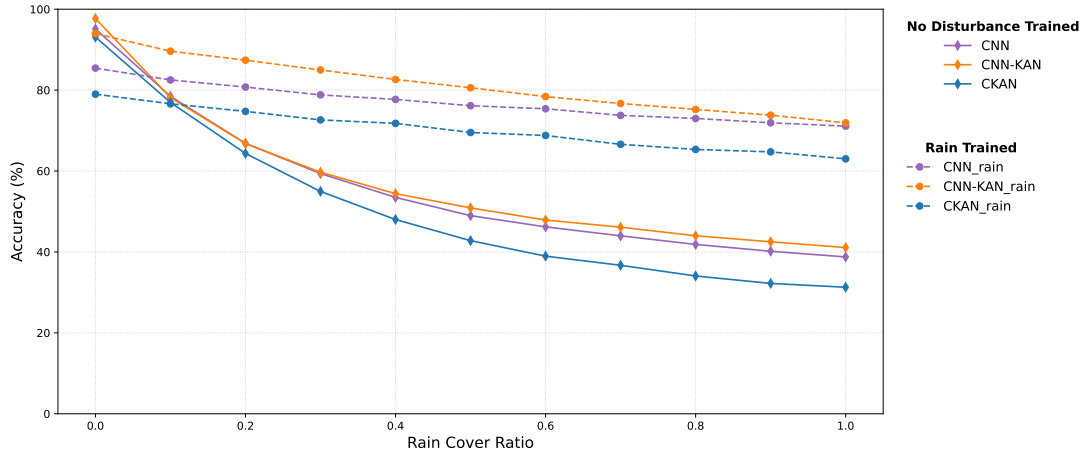

Fig. 9. Accuracy depending on the rain coverage factor α.

to 92.28% when trained on rotated data. This 5.4% drop highlights the challenge of the stability-plasticity dilemma, where the model sacrifices some specificity on aligned images to accommodate the variance required for global rotational generalization.

TABLE III
ACCURACY [%] OF MODELS UNDER ROTATIONAL DISTURBANCE. BEST RESULTS IN EACH COLUMN IN BOLD FONT.

Network Model	Rotation Angle ϕ			
	0°	30°	90°	180°
CNN	95.11	45.53	17.07	25.80
CNN-KAN	**97.70**	51.03	14.15	29.40
CKAN	93.12	43.37	18.02	28.38
CNN_rotation	89.94	**83.76**	**76.14**	**78.80**
CNN-KAN_rotation	92.28	81.93	71.69	75.15
CKAN_rotation	84.96	79.63	68.90	74.22

Comparing the specific architectures, the CNN-KAN model demonstrates superior feature extraction capabilities on aligned data, consistently achieving the highest peak accuracy in both training regimes. However, in terms of stability at extreme angles, the standard CNN architecture proves to be slightly more robust; at $\phi = 180°$, the CNN_rotation achieves 78.80%, slightly outperforming both the CNN-KAN_rotation (75.15%) and CKAN_rotation (74.22%) – see Table III. This suggests that while Kolmogorov-Arnold layers offer an advantage in peak precision, standard convolutional layers may offer marginally better stability under extreme geometric transformations when adequately trained.

C. Robustness to Rain

The experimental results regarding rain disturbances reveal a distinct vulnerability of standard models to visual occlusion. As the rain coverage factor α increases from 0 to 1.0, the accuracy of the baseline models (CNN, CNN-KAN, CKAN) exhibits a steep, nearly monotonic decline. While the CNN-KAN model achieves an impressive baseline accuracy of 97.70% on clean data, it suffers a severe degradation of over 56 percentage points, dropping to 41.08% at maximum rain coverage ($\alpha = 1.0$). Similarly, the standard CNN and CKAN models plummet to 38.77% and 31.27% respectively. This indicates that without specific training, the feature extractors

are unable to filter out high-frequency vertical noise (rain streaks) or reconstruct occluded semantic features.

Training with rain-augmented data drastically improves robustness, transforming the performance degradation from a steep drop into a gradual decline, as presented in Fig. 9 and Table IV. The models trained on rain data (suffix `_rain`) maintain functional accuracy even under severe occlusion. At $\alpha = 1.0$, the `CNN-KAN_rain` model retains an accuracy of 71.92%, a substantial improvement over the 41.08% recorded by its standard counterpart. The `CNN_rain` shows a similar recovery, improving from $\approx 39\%$ to 71.09% at the highest noise level. This suggests that the models successfully learned to treat rain streaks as background noise rather than salient features.

A critical finding in this experiment is the superior capacity of the `CNN-KAN` architecture to minimize the trade-off between robustness and clean-data accuracy. Typically, robust models suffer a performance drop on clean data.

- The `CNN_rain` model drops significantly on clean data ($\alpha = 0$), falling from 95.11% (standard) to 85.44% (robust), a loss of nearly 10%.
- In contrast, the `CNN-KAN_rain` model preserves its high performance remarkably well, dropping only from 97.70% to 93.96% on clean data.

Consequently, `CNN-KAN_rain` emerges as the most effective model, maintaining the highest accuracy across the entire domain of α. It outperforms `CNN_rain` by approximately 8.5 percentage points on clean data while matching its robustness at extreme noise levels. The `CKAN` architecture consistently underperforms in this test, showing the lowest accuracy in both standard and robust configurations across almost all values of α.

TABLE IV
ACCURACY [%] OF MODELS UNDER RAIN DISTURBANCE. BEST RESULTS IN EACH COLUMN IN BOLD FONT.

Network Model	Rain Coverage Ratio		
	$\alpha = 0.0$	$\alpha = 0.5$	$\alpha = 1.0$
CNN	95.11	48.97	38.77
CNN-KAN	**97.70**	50.89	41.08
CKAN	93.12	42.79	31.27
CNN_rain	85.44	76.14	71.09
CNN-KAN_rain	93.96	**80.59**	**71.92**
CKAN_rain	78.99	69.52	63.02

D. Robustness to Blur

The investigation into Gaussian blur disturbances reveals that blur is the most destructive noise type for all tested architectures. Unlike rotation (which preserves information but changes orientation) or rain (which occludes parts of the image), blur irreversibly removes high-frequency spatial details essential for feature extraction.

For the standard models trained on clean data (`CNN`, `CNN-KAN`, `CKAN`), performance collapses rapidly as the blur radius r increases (see Fig. 10 and Table V). While these models tolerate minimal blurring ($r = 1$) with only a slight accuracy dip, a radius of $r = 2$ triggers a massive performance drop of approximately 20–25 percentage points across all architectures. By $r = 4$, the accuracy of the best-performing model (`CNN-KAN`) falls to 40.73%, and at extreme blur levels ($r \geq 16$), all standard models converge to near-random performance ($< 10\%$). This confirms that standard convolutional filters are heavily reliant on sharp edge definitions and texture details, which are the first to disappear under blurring.

Training with blurred data (models with the suffix `_blur`) significantly mitigates this degradation but cannot fully recover performance due to the inherent information loss. The robust models exhibit a much more graceful degradation curve. For example, at a moderate blur radius of $r = 5$, the `CNN_blur` and `CNN-KAN_blur` models maintain accuracies of 55.68% and 52.30% respectively, compared to $\approx 30\%$ for their standard counterparts. However, unlike the rotation experiment where training restored high accuracy, no model could maintain high performance at extreme blur radii ($r = 24$), where accuracies remained low ($\approx$ 10–13%).

A comparative analysis of the architectures reveals a notable crossover in performance characteristics:

- **Low noise** ($r < 4$): The `CNN-KAN_blur` architecture is superior. It preserves the highest accuracy on clean data (94.49% vs. 88.72% for `CNN_blur`), demonstrating that Kolmogorov-Arnold layers are better at retaining fine-grained feature specificity when the image quality is decent.
- **High noise** ($r \geq 4$): Surprisingly, the standard `CNN_blur` surpasses the `CNN-KAN_blur` as the image becomes heavily degraded. At $r = 8$, the `CNN` outperforms the `CNN-KAN` by roughly 4.6 percentage points (40.98% vs. 36.39%), and this lead widens slightly at higher radii.

This suggests that while CNN-KANs are more precise extractors of detailed features, standard convolutional layers may be more robust at latching onto coarse, low-frequency structural shapes (silhouettes) when high-frequency details are completely obliterated.

TABLE V
ACCURACY [%] OF MODELS UNDER BLUR DISTURBANCE. BEST RESULTS IN EACH COLUMN IN BOLD FONT.

Network Model	Blur Radius			
	$r = 0$	$r = 1$	$r = 4$	$r = 12$
CNN	95.11	91.20	39.05	10.47
CNN-KAN	**97.70**	**92.45**	40.73	12.39
CKAN	93.12	87.15	38.25	9.44
CNN_blur	88.72	86.83	**62.63**	**28.52**
CNN-KAN_blur	94.49	91.58	59.97	24.83
CKAN_blur	91.32	87.38	54.84	24.16

E. Robustness to Brightness Shifts

The experiments regarding brightness variations (controlled by the factor v) reveal that all models exhibit a distinct

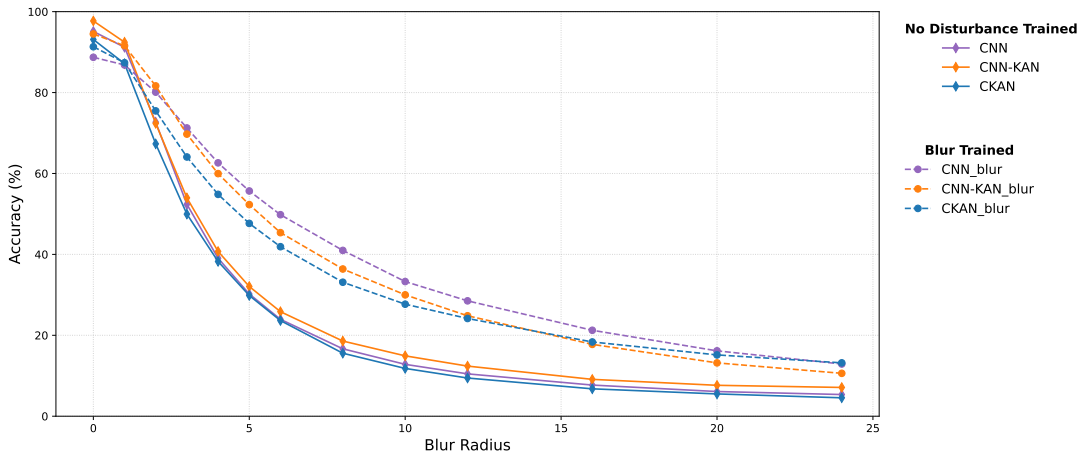

Fig. 10. Accuracy depending on the radius of blur r.

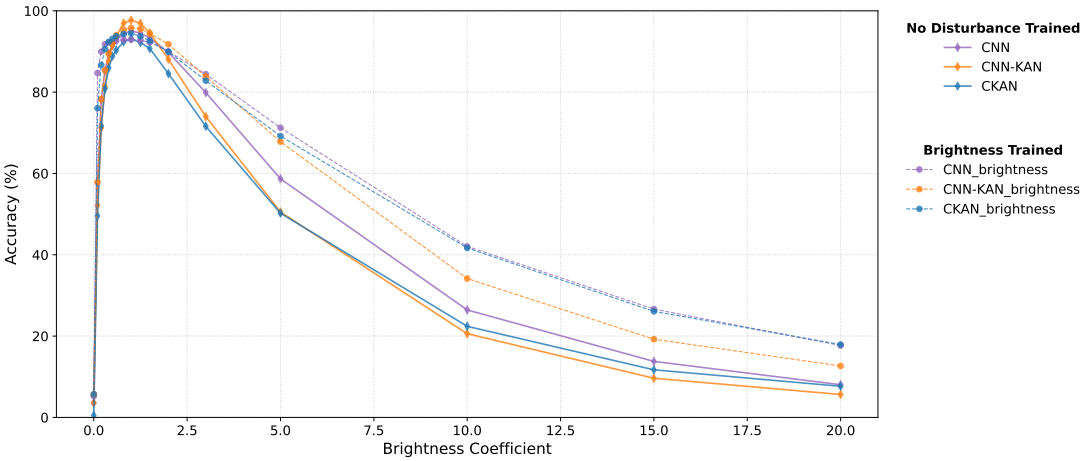

Fig. 11. Accuracy depending on the brightness factor v.

bell-shaped performance curve centered around the canonical brightness $v = 1$, as shown in Fig. 11.

For the standard models (`CNN`, `CNN-KAN`, `CKAN`), deviations from $v = 1$ cause symmetrical performance degradation, though the sensitivity is asymmetric regarding overexposure versus underexposure.

- **Underexposure** ($v < 1$)**:** The models are surprisingly resilient to moderate darkening. At $v = 0.5$ (half brightness), the `CNN-KAN` retains 91.58% accuracy. However, performance collapses in near-total darkness; at $v = 0.1$, the accuracy of all standard models drops to the range of 49%–57%.
- **Overexposure** ($v > 1$)**:** The models struggle significantly more with high-intensity saturation. At $v = 5$ (5x brightness), the best performing model (`CNN`) drops to 58.69%, and at extreme exposure ($v = 20$), all models fail completely ($< 8\%$), as the image information is washed out into pure white.

Training with brightness-augmented data (models with suffix `_brightness`) expands the effective operational range, particularly in low-light conditions. A striking disparity emerges between the architectures in the dark regime ($v = 0.1$):

- The `CNN_brightness` model demonstrates exceptional night-vision robustness, maintaining an accuracy of 84.70% at $v = 0.1$, a massive improvement over its standard version (57.66%).
- In contrast, the `CNN-KAN_brightness` fails to generalize to darkness, achieving only 57.88% at $v = 0.1$, which is statistically negligible compared to its standard counterpart (52.16%).

This suggests that the Kolmogorov-Arnold layers, while precise at feature extraction in optimal lighting, struggle to adapt to low-contrast signal-to-noise ratios compared to standard convolutions.

Uniquely, the `CKAN` architecture benefits the most from brightness augmentation – compare Table VI. It is the only architecture where the robust version outperforms the standard version even at the optimal point $v = 1$ (94.46% vs. 93.12%). This indicates that for the `CKAN` architecture, brightness variation acts as a beneficial regularizer that improves general feature learning, rather than just a robustness patch.

TABLE VI
ACCURACY [%] OF MODELS UNDER BRIGHTNESS DISTURBANCE. BEST RESULTS IN EACH COLUMN IN BOLD FONT.

	Brightness Coefficient			
Network Model	$v = 0.1$	$v = 1.0$	$v = 2.0$	$v = 5.0$
CNN	57.66	95.11	89.77	58.69
CNN-KAN	52.16	**97.70**	88.07	50.52
CKAN	49.57	93.12	84.55	50.23
CNN_brightness	**84.70**	92.93	89.99	**71.24**
CNN-KAN_brightness	57.88	95.82	**91.78**	67.81
CKAN_brightness	76.05	94.46	89.94	69.20

V. CONCLUSIONS

The comprehensive evaluation of the three architectures (CNN, CNN-KAN, CKAN) across four distinct distortion domains leads to critical insights regarding the trade-offs between model complexity, robustness, and computational efficiency. A dominant factor in this comparison is the model size. The CNN-KAN architecture, with a storage footprint of approximately 44 MB, is nearly an order of magnitude larger than the standard CNN (4.7 MB) and five times larger than the CKAN (9 MB). While the CNN-KAN consistently achieved the highest peak accuracy on clean and rain-occluded data, its massive parameter count yields diminishing returns in robust scenarios. For resource-constrained environments typical of traffic sign recognition, the standard CNN offers a superior efficiency-to-robustness ratio, providing comparable stability at roughly 10% of the storage cost.

The experiments revealed distinct inductive biases in how the different architectures process information. The standard convolutional layers (in CNN) demonstrated superior stability in low signal-to-noise conditions, such as extreme blur ($r \geq 8$) or severe darkness ($v = 0.1$). This implies that standard convolutions are robust at latching onto coarse, structural shapes (silhouettes) that remain visible when fine details vanish. In contrast, the Kolmogorov-Arnold dense layers (in CNN-KAN) excelled at the complex tasks, emerging as the undisputed leader in handling rain streaks and maintaining high accuracy on rotated inputs. However, the CNN-KAN struggled to generalize in low-contrast environments, suggesting that KA-dense layers rely heavily on clear, high-frequency feature definitions to function effectively.

Finally, the nature of the disturbance fundamentally dictated the success of the robustness training. Geometric transformations (rotation) proved to be the most solvable problem, where training with augmented data restored functional accuracy across the full spectrum. Conversely, information-loss disturbances (blur) presented the hardest challenge; even robust models could not recover performance at high blur radii because the spatial information was irreversibly destroyed. Across all experiments, the stability-plasticity dilemma was evident, where training for robustness generally incurred a penalty on clean-data accuracy. The CNN-KAN managed this trade-off best in the experiment with rain mask added, while the CKAN architecture showed a unique benefit in the simulations with brightness changes, where augmentation actually improved its baseline performance.

REFERENCES

[1] A. D. Bodner, A. S. Tepsich, J. N. Spolski, and S. Pourteau, "Convolutional kolmogorov-arnold networks," 2024. [Online]. Available: https://arxiv.org/abs/2406.13155

[2] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "Kan: Kolmogorov-arnold networks," 2024. [Online]. Available: https://arxiv.org/abs/2404.19756

[3] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: A multi-class classification competition," in *The 2011 International Joint Conference on Neural Networks*, 2011, pp. 1453–1460. [Online]. Available: https://doi.org/10.1109/IJCNN.2011.6033395

[4] A. Wong, M. J. Shafiee, and M. St. Jules, "Micronnet: A highly compact deep convolutional neural network architecture for real-time embedded traffic sign classification," *IEEE Access*, vol. 6, pp. 59 803–59 810, 2018. [Online]. Available: https://doi.org/10.1109/ACCESS.2018.2873948

[5] A. Hechri and A. Mtibaa, "Two-stage traffic sign detection and recognition based on svm and convolutional neural networks," *IET Image Processing*, vol. 14, no. 5, pp. 939–946, 2020. [Online]. Available: https://doi.org/10.1049/iet-ipr.2019.0634

[6] O. N. Manzari and S. Baradaran Shokouhi, "A robust network for embedded traffic sign recognition," in *2021 11th International Conference on Computer Engineering and Knowledge (ICCKE)*, 2021, pp. 447–451. [Online]. Available: https://doi.org/10.1109/ICCKE54056.2021.9721520

[7] S. Saxena, S. Dey, M. Shah, and S. Gupta, "Traffic sign detection in unconstrained environment using improved yolov4," *Expert Systems with Applications*, vol. 238, p. 121836, 2024. [Online]. Available: https://doi.org/10.1016/j.eswa.2023.121836

[8] S. Sołtysiak, P. Biernacki, and U. Libal, "Traffic sign recognition with convolutional kolmogorov-arnold networks," in *2025 Signal Processing Symposium (SPSympo)*, 2025, pp. 147–152. [Online]. Available: https://doi.org/10.23919/SPSympo63739.2025.11123989

[9] J. Chung, S. Park, D. Pae, H. Choi, and M. Lim, "Feature-selection-based attentional-deconvolution detector for german traffic sign detection benchmark," *Electronics*, vol. 12, no. 3, 2023. [Online]. Available: https://doi.org/10.3390/electronics12030725

[10] R. Mahadshetti, J. Kim, and T.-W. Um, "Sign-yolo: Traffic sign detection using attention-based yolov7," *IEEE Access*, vol. 12, pp. 132 689–132 700, 2024. [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3417023

[11] H. Assemlali and N. Sael, "Traffic sign classification using deep learning comparative study," *Procedia Computer Science*, vol. 233, pp. 939–949, 2024, 5th International Conference on Innovative Data Communication Technologies and Application (ICIDCA 2024). [Online]. Available: https://doi.org/10.1016/j.procs.2024.03.283

[12] D. R. Pandey and K. S. Raghatat, "Real-time traffic sign detection and recognition using faster r-cnn with vgg16 backbone," in *2025 International Conference on Automation and Computation (AUTOCOM)*, 2025, pp. 279–284. [Online]. Available: https://doi.org/10.1109/AUTOCOM64127.2025.10956931