Received xxx; | Revised xxx; | Accepted xxx

DOI: xxx/xxxx

ORIGINAL ARTICLE

Evidential transformer for pavement distress segmentation

Zheng Tong | Tao Ma* | Weiguang Zhang | Ju Huyan

[1]School of Transportation, Southeasst University, Nanjing, China

***Correspondence**
Tao Ma, School of Transportation, Southeasst University, Jiulonghu Campus, Nanjing, 211189, China. Email: matao@seu.edu.cn

Present address
School of Transportation, Southeasst University, Jiulonghu Campus, Nanjing, 211189, China.

Abstract

Distress segmentation assigns each pixel of a pavement image to one distress class or background, which provides a simplified representation for distress detection and measurement. Even though remarkably benefiting from deep learning, distress segmentation still faces the problems of poor calibration and multi-model fusion. This study has proposed a deep neural network by combining the Dempster-Shafer theory (DST) and transformer for pavement distress segmentation. The network, called the evidential segmentation transformer, uses its transformer backbone to obtain pixel-wise features from input images. The features are then converted into pixel-wise mass functions by an DST-based evidence layer. The pixel-wise masses are utilized for performing distress segmentation based on the pignistic criterion. The proposed network is iteratively trained by a new learning strategy, which represents uncertain information of ambiguous pixels by mass functions. In addition, an evidential fusion strategy is proposed to fuse heterogeneous transformers with different distress classes. Experiments using three public datasets (Pavementscape, Crack500, and CrackDataset) show that the proposed networks achieve state-of-the-art accuracy and calibration on distress segmentation, which allows for measuring the distress shapes more accurately and stably. The proposed fusion strategy combines transformers on heterogeneous datasets while remaining a performance not less than those of the individual networks on their own datasets, which makes it possible to use the existing networks to build a more general and accurate one for distress segmentation.

KEYWORDS:
Distress segmentation, pavement inspection, pavement distress dataset, Dempster-Shafer theory, deep learning

1 | INTRODUCTION

Visual pavement distress inspection benefits from digital imaging technology Arabi, Haghighat, & Sharma (2020) and deep learning Jeong, Jo, & Ditzler (2020). In recent years, many deep neural networks have been proposed to perform distress segmentation A. Zhang et al. (2017). Distress segmentation Zhu et al. (2022) is defined as the process of assigning each pixel in a digital pavement image to one of the possible distress classes or "background". The results of distress segmentation, called *distress masks*, are the sets of pixels belonging to different classes. The masks are regarded as a simple representation of the original image, which has been used for distress recognition Bang, Hong, & Kim (2021), detection Gao & Mosalam (2018), and measurement Tong, Yuan, Gao, & Wang (2020).

[0]**Abbreviations:** Dempster-Shafer theory, DST; probabilistic fully convolution network, P-FCN; probabilistic segmentation transformer, PS-transformer; evidential segmentation transformer, ES-transformer; evidential neural network, ENN; multi-headed self-attention layer, MSL; two-layer perceptron, TLP; norm layer, NL; vision transformer, ViT; mass-fusion ES-transformer network, MFES network; probability-to-mass fusion, PMF; Bayesian fusion, BF; probability feature-combination, PFC; Evidential feature-combination, EFC.

There are two main directions of deep neural networks for distress segmentation: convolution- and attention-based networks. The former A. Zhang et al. (2019) used several stages of convolution and pooling layers to extract feature maps from its input and then unsampled the features into pixel-wise representations. The representations were finally converted into pixel-wise probabilities of different classes by a softmax layer for distress segmentation. This study named such a network as the *probabilistic fully convolutional network (P-FCN)*. The latter one Wang & Su (2022) used some attention-based layers to extract feature maps and unsampled the features into pixel-wise representations by a mask transformer. Similarly, pixel-wise representations were used to build pixel-wise probabilities for decision-making Qu, Li, & Zhou (2022). The network is called the *probabilistic segmentation transformer (PS-transformer)* in the study. Until now, some studies Sun, Xie, Jiang, Cao, & Liu (2022) reported that a PS-transformer outperformed a P-FCN on the distress segmentation once given enough learning images.

Even though the two types of deep neural networks have achieved state-of-the-art performances on distress segmentation, they still face two problems: over-confidence and multi-model fusion. Over-confidence, also known as *poor calibration*, means that the accuracy of a network cannot match its confidence. For example, a transformer network has an average pixel accuracy of 90% in a held-out testing set but always outputs the maximum probabilities higher than 95% in the testing set. This indicates the network is highly confident in its predictions but there is a gap of 5% between its testing accuracy and confidence. This behavior is common in the probabilistic deep neural networks owing to the disadvantage of the Bayesian probability framework, which only captures the randomness aspect of the data, but neither ambiguity nor incompleteness. More details of this disadvantage can be found in Guo, Pleiss, Sun, & Weinberger (2017). This disadvantage makes it difficult for users to know when a deep neural network will fail to segment pavement distress. In addition, over-confident networks maybe not be as accurate as users expect them to be.

Another problem is the multi-model fusion F.-C. Chen & Jahanshahi (2017), in which the outputs from different deep networks using the data from different sources are combined into ones for decision-making Diaby, Germain, & Goïta (2021). One challenge in pavement distress segmentation is to utilize the existing networks trained from heterogeneous datasets for obtaining a new one, which can improve the generality and accuracy of distress segmentation. For example, a network can segment crack and pothole pixels from the pavement background, while another can distinguish the repair and crack areas from the pavement background. If the information from the two networks can be fused, a new model can be obtained, which has the capacity of segmenting cracks, potholes, repair areas, and pavement background. Unfortunately, the problem of data uncertainty in the Bayesian probability framework, especially the partial and imperfect outputs of deep neural networks, makes it difficult to fuse heterogeneous networks. Still take the two networks as an example. Given a pixel, one network outputs probabilities of classes "crack", "pothole", and "background", while another outputs probabilities of classes "repair area", "crack", and "background". One probability distribution is partial and imperfect for another and the probability information cannot be combined by Bayes' rule. Some feature-level fusion methods were proposed to solve this problem, in which the features from different deep neural networks are concatenated for decision-making. However, they required extra training and sometimes have lower performance than those of the individual networks on their respective distress class set.

Dempster-Shafer theory (DST) provides a potential way to solve the two problems deriving from the probabilistic framework. As a generalization of probability theory, DST Dempster (1967), also referred to as *evidence theory* or *theory of belief functions*, has been a well-established formalism for representing and combining a large variety of uncertain information for decision-making Yager & Liu (2008). The framework of DST uses the mass functions with the complete monotone to represent independent pieces of evidence and then combines them into ones by a generic operator, such as Dempster's rule Shafer (1976).

DST has been increasingly applied to machine learning with uncertain data, following two main directions: designing evidential classifiers Denœux, Kanjanatarakul, & Sriboonchitta (2019) and combining multiple network models Minary, Pichon, Mercier, Lefevre, & Droit (2019). An evidential classifier Denœux et al. (2019) regarded the elements of its input vector as pieces of evidence and converts them into mass functions. The masses were finally aggregated via Dempster's rule to represent uncertain information in the input vector. For example, there existed ambiguous data in an input vector when the values of two masses in an evidential classifier outputs were very close. Data unreliability and imprecision were also represented by mass functions in the framework of DST Tong, Xu, & Denœux (2021b). The mass-function representations gave a potential way to solve the problem of over-confidence. In the direction of multi-network combination, the mass functions from different models were aggregated by Dempster's rule or any other rule Jiang, Wang, Gao, Gao, & Gao (2017). This direction provided the possibility to process incomplete data by extending heterogeneous imperfect outputs into a common frame and combining them by Dempster's rule Tong, Xu, &

Denœux (2021c). The two practical directions gives the following three advantages of DST to solve distress segmentation using deep neural networks.

Operationality: DST can easily be put into practice by breaking down a pixel-wise feature vector from an FCN on a distress segmentation task into elementary pieces of evidence, combining them by Dempsters rule Denœux (2019). This indicates that DST is easier to combine with FCNs for distress segmentation than other uncertainty theories (e.g., imprecise probability and fuzzy sets).

Generality: DST can do much more than sets or probabilities because it is based on the idea of combining sets and probabilities Denoeux & Shenoy (2020). It can extend both Bayesian probabilistic reasoning and propositional logic, such as computing with sets and interval analysis. This advantage makes it possible to calibrate the randomness and uncertainty on distress segmentation tasks and reduce the over-confidence in the neural network, which Bayesian probabilistic reasoning cannot do.

Fusion: The mass functions of DST can be easily combined by Dempsters rule, even though some only represent partial and imperfect outputs of a deep neural network Tong et al. (2021c). The advantage has the potential to fuse different FCNs with heterogeneous class sets of pavement distresses.

Motivated by the high expressivity of DST as an uncertainty representation framework, the objective of this study is to develop a new transformer network in the framework of DST to solve the problems of over-confidence and multi-network fusion in the task of pavement distress segmentation. The basic idea is that a transformer network provides the pixel-wise feature representations of pavement images and a DST-based evidential classifier converts the representations into pixel-wise mass functions for distress segmentation. The contributions of the thesis can be summarized in the following three points.

Evidential transformer network: A new transformer network has been proposed to segment pavement distresses, called the *evidential segmentation transformer (ES-transformer)*. The proposed network achieves top-performing accuracy and reasonable calibration on distress segmentation.

Evidential fusion of heterogeneous transformers: The mass-function outputs of the ES-transformer make it possible to combine heterogeneous deep networks. This approach is flexible enough to combine different transformers with inconsistent pavement distress categories to obtain a more general network.

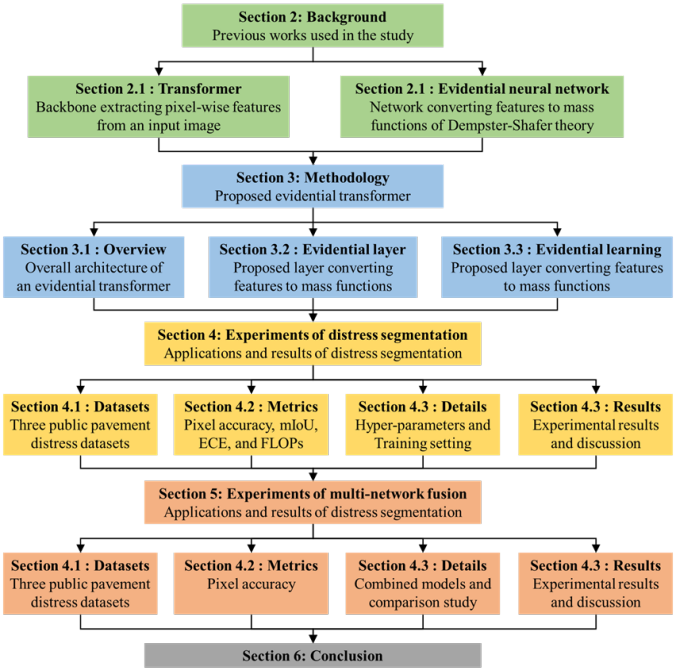

FIGURE 1 Paper framework.

Evidential learning strategy: An end-to-end learning strategy has been proposed to update the parameters in an ES-transformer network using a learning set, which represents uncertain and imprecise information represented in the form of DST-based mass functions to improve the accuracy and calibration of the transformer.

The rest of the paper is organized as Figure 1 . Section 2 begins with a brief recall of the segmentation transformer network and DST, which are the previous studies used in the proposed transformer network. Section 3 describes the details of the proposed network for pavement distress segmentation, which is the originality of the study. Sections 4 and 5 report the numerical experiments of distress segmentation and multi-network fusion, respectively, demonstrating the superiority of the proposed network. Finally, Section 6 concludes the study.

2 | BACKGROUND

This section gives the necessary background of the proposed network, including the transformer network for semantic segmentation in Section 2.1 and DST for uncertainty reasoning in Section 2.2.

2.1 | Segmentation transformer

In the two years, some attention-based networks Oktay et al. (2018); Qin et al. (2020) have been proposed to perform

semantic segmentation. Segmentation transformer Strudel, Garcia, Laptev, & Schmid (2021) is a successful case of attention-based networks, which will be combined with DST in this study. A segmentation transformer mainly consists of three parts: a transformer encoder for feature extraction and a decoder for feature upsampling. The architectures of the encoder and decoder are called the *segmentation-transformer backbone* in this study.

Encoder

The encoder part, following the architecture of vision transformer Dosovitskiy et al. (2021), split an image $\boldsymbol{X} \in \mathbb{R}^{W\times H\times C}$ into a sequence of patches $\{\boldsymbol{x}_i,\ i = 1, \dots N\},\ \boldsymbol{x}_i \in \mathbb{R}^{S\times S\times C}$, where $S\times S$ is the patch size, $N = \frac{WH}{S^2}$ is the number of patch, and C is the number of channels in the input image. Each patch is flattened into a single vector by concatenating its channels of all elements and then linearly projected to a sequence of patch embeddings $\boldsymbol{X'} = \{\boldsymbol{E}(\boldsymbol{x}_1), \dots, \boldsymbol{E}(\boldsymbol{x}_N)\}$ with $\boldsymbol{E}(\boldsymbol{x}_i) \in \mathbb{R}^D$, $i = 1, \dots, N$, and $D = C \cdot S^2$. After the flattening operation, the encoder part is agnostic to the position information about these patch embeddings. Thus, learnable position embeddings $\boldsymbol{po} = \{\boldsymbol{po}_i,\ i = 1, \dots, N\}$ with $\boldsymbol{po}_i \in \mathbb{R}^D$ are linearly added to each embedding as $\boldsymbol{z}_0 = \{\boldsymbol{E}(\boldsymbol{x}_i) + \boldsymbol{po}_i,\ i = 1, \dots, N\}$.

The tensor $\boldsymbol{z_0}$ passes through L stages in the encoder to obtain a sequential set $\boldsymbol{z}_L = \{\boldsymbol{z}_{L,i},\ i = 1, \dots, N\}$ with $\boldsymbol{z}_{L,i} \in \mathbb{R}^d$, where d is the dimension of each vector in the sequential set $\boldsymbol{z}_L$. Each stage comprises multi-headed self-attention layer (MSL) and a two-layer perceptron (TLP), followed by a norm layer (NL). Thus, the procedure of the l-th stage can be summarized as

$$\boldsymbol{a}_{l-1} = \text{MSL}(\text{NL}(\boldsymbol{z}_{l-1})) + \boldsymbol{z}_{l-1}, \tag{1a}$$

$$\boldsymbol{z}_{l} = \text{TLP}(\text{NL}(\boldsymbol{a}_{l-1})) + \boldsymbol{a}_{l-1}, \tag{1b}$$

with $l = 1, \dots, L$. In the MSL of the l-th stage, following the self-attention mechanism Vaswani et al. (2017), the tensor $\text{NL}(\boldsymbol{z}_{l-1})$ is multiplied with three sets of weights to get its representations, key $\boldsymbol{K} \in \mathbb{R}^{N\times d}$, query $\boldsymbol{Q} \in \mathbb{R}^{N\times d}$, and value $\boldsymbol{V} \in \mathbb{R}^{N\times d}$. The three representations are then converted into one tensor as

$$\text{MSL}(\boldsymbol{K}, \boldsymbol{Q}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}}\right)\boldsymbol{V}. \tag{2}$$

Therefore, the encoder part extracts a sequence of contextualized encodings $\boldsymbol{z}_L$ from the input image, which includes the high-dimension semantic information.

Decoder

In the decode part, the sequence of contextualized encodings $\boldsymbol{z}_L \in \mathbb{R}^{N\times D}$ is upsampled to a tensor $\boldsymbol{F} \in \mathbb{R}^{W\times H\times P}$, where $\boldsymbol{P}$ is the channel number of the feature maps and $\boldsymbol{F}$ is also refereed to as the pixel-wise maps and features.

The decoder part consists of L' MSLs. Let's the sequence $\boldsymbol{z}_L^{l'}$ be the inputs of the l'-th MSL, $l' = 1, \dots, L'$. The l'-th MSL initializes randomly P learnable embeddings $\boldsymbol{le} = \{le_1, \dots, le_P\} \in \mathbb{R}^{P\times D}$ and performs the scalar product between its inputs and the learnable embeddings as

$$\text{PS}(\boldsymbol{z}_L^{l'}, \boldsymbol{le}) = \left\{\text{PS}(\boldsymbol{z}_{L,i}^{l'}, \boldsymbol{le}),\ i = 1, \dots, N\right\} = \boldsymbol{z}_L^{l'} \cdot \boldsymbol{le}^T, \tag{3}$$

with $l' = 1, \dots, L'$. Then, $\text{PS}(\boldsymbol{z}_{L,i}^{l'}, \boldsymbol{le})$ are reshaped into a 2D tensor and bilinearly upsampled to a tensor $\boldsymbol{F}_i \in \mathbb{R}^{W/S\times H/S\times P}$, $i = 1, \dots, N$. Then, the N upsampled tensors are attached into a feature map $\boldsymbol{F} \in \mathbb{R}^{H\times W\times P}$. In a PS-transformer, the pixel-wise feature map $\boldsymbol{F}$ is regarded as the representations of the input image and converted into pixel-wise probabilities for semantic segmentation using a softmax layer. In Section 3, each vector with P dimension from $\boldsymbol{F}$ is considered as the representations of the pixel location and will be converted into mass functions for pixel-wise decision-making. Here, the procedure of the segmentation-transformer backbone (the encoder and decoder parts) is simply notated as a function $\boldsymbol{\psi}(\cdot)$ and $\boldsymbol{\psi}(x) = (\psi_1(x), \dots, \psi_P(x)) \in \boldsymbol{F}$ stands the feature vector of a pixel $x \in \boldsymbol{X}$ with $\psi_j(x) \in \mathbb{R},\ j = 1, \dots, P$.

2.2 | Evidential neural network

One main application of DST is to design an evidential classifier, also known as *evidential neural network (ENN)* Denœux (2019), which converts a feature vector to mass functions and quantifies the uncertainty of the vector using mass functions. Thus, ENNs establish a bridge between DST and transformers at the feature level. The output mass functions of an ENN are used for decision-making with uncertainty Tong, Xu, & Denœux (2019). With the generality and operationality of DST mass functions, an ENN gives more informative outputs than the probabilistic models that quantify prediction uncertainty using a probability distribution. This section introduces a particular ENN Denœux (2019) that is combined with transformers in the study.

Let $\Omega = \{\omega_1, \dots, \omega_M\}$ be a *class set* and $\boldsymbol{\psi}(x) = (\psi_1(x), \dots, \psi_P(x))$ be a vector of P features for a pixel $x \in \boldsymbol{X}$. Each feature value $\psi_j(x)$ is regarded as the evidence of the supports either to singleton set $\{\omega_i\}$ or to its complement $\overline{\{\omega_i\}}$ in the form as

$$\tau_{ij} := \beta_{ij}\psi_j(x) + \alpha_{ij}, \tag{4}$$

where β_{ij} and α_{ij} are two parameters associated to evidence $\psi_j(x)$ and singleton $\{\omega_i\}$, $i = 1, \dots, M$, $j = 1, \dots, P$. The weights of evidence $\psi_j(x)$ for $\{\omega_i\}$ and $\overline{\{\omega_i\}}$ are equal to the positive and negative parts of τ_{jk}, notated as τ_{ij}^+ and τ_{ij}^-, respectively. For each feature $\psi_j(x)$ and singleton set $\{\omega_i\}$, two simple mass function can be defined as $m_{ij}^+ := \{\omega_i\}^{w_{ij}^+}$ and $m_{ij}^- := \{\omega_i\}^{w_{ij}^-}$, following the notations of the weights of evidence in Shafer (1976).

The two simple masses can be fused into ones by aggregating the positive and negative weights of evidences w.r.t singleton set $\{\omega_i\}$ using Dempster's rule $\bigoplus$ as

$$m_i^+ = \bigoplus_{j=1}^{P} m_{ij}^+ = \{\omega_i\}^{w_i^+} \tag{5a}$$

$$m_i^- = \bigoplus_{j=1}^{P} m_{ij}^- = \overline{\{\omega_i\}}^{w_i^+}, \tag{5b}$$

with

$$w_i^+ := \sum_{j=1}^{P} w_{ij}^+ \quad \text{and} \quad w_i^- := \sum_{j=1}^{P} w_{ij}^-.$$

where $\bigoplus$ for two independent masses m_1 and m_2 is defined as

$$(m_1 \oplus m_2)(A) = \frac{1}{1-\kappa} \sum_{B \cap C = A} m_1(B)\, m_2(C) \tag{6}$$

for all $A \subseteq \Omega$, $A \neq \emptyset$, and $(m_1 \oplus m_2)(\emptyset) = 0$. In (6), $\kappa < 1$ is the *degree of conflict* between m_1 and m_2 as

$$\kappa := \sum_{B \cap C \neq \emptyset} m_1(B)\, m_2(C). \tag{7}$$

Therefore, once given a feature vector $\boldsymbol{\psi}(x)$, the ENN outputs mass functions as

$$m_{\boldsymbol{\psi}(x)} = \bigoplus_{i=1}^{M} \left(\{\omega_i\}^{w_i^+} \oplus \overline{\{\omega_i\}}^{w_i^-} \right). \tag{8}$$

Thus, the final outputs of the ENN is a mass-function vector $\boldsymbol{m}_{\boldsymbol{\psi}(x)} = \left(m_{\boldsymbol{\psi}(x)}(A), A \subseteq \Omega \backslash \emptyset\right)^T$.

3 | METHODOLOGY

This section describes the proposed model, so-called the *evidential segmentation transformer (ES-transformer)*. Section 3.1 presents the overview of the proposed model which is made up of a segmentation-transformer backbone for pixel-wise feature representation, a DST-based evidence layer to build mass functions, and a decision layer using mass functions to perform pavement distress segmentation. A learning strategy is exposed in Section 3.3, which qualities the uncertainty in a learning set by introducing soft labels. Finally, an approach for evidential fusion is proposed in Section 3.4, in which heterogeneous transformers can be combined using mass functions.

3.1 | Overview

An ES-transformer hybridizes a segmentation transformer described in Section 2.1 and an ENN introduced in Section 2.2 by "installing" an evidence layer behind the final MSL in the decoder part of a segmentation-transformer backbone. Figure 2 presents the overview of an ES-transformer. An ES-transformer is able to perform pavement distress segmentation and represent the confusing information about the class prediction of each pixel in an image by mass function. To distinguish the ES-transformer from the network in Section 2.1, this study named the transformer that converts the pixel-wise features from its backbone into probabilities using a softmax layer as the *probabilistic segmentation transformer (PS-transformer)*. The processes of the ES-transformer can be summarized in three steps as follows.

Step 1: A pavement image of size $W \times H \times C$ is imported into the segmentation-transformer backbone to obtain pixel-wise features of size $W \times H \times P$, in which a vector with P-dimension is the features of a pixel location. The procedure of this step has been introduced in Section 2.1. This step obtains precise and reliable features based on the input image, making the proposed network yield comparable or more effective capacity of distress segmentation than a PS-transformer using the identical segmentation-transformer backbone.

Step 2: Each vector with P-dimension in Step 1 is fed into the ENN as introduced in Section 2.2. This part transforms the feature vector into an $(2^M - 1)$-dimensional mass function vector as

$$\boldsymbol{m} = \left(m_{\boldsymbol{\psi}(x)}(A), A \subseteq \Omega \backslash \emptyset\right)^T \tag{9}$$

$$= (m(A_1), \ldots, m(A_{2^M-2}), m(\Omega))^T. \tag{10}$$

Therefore, when importing the feature maps of size $W \times H \times P$ in Step 1, this step generates a mass-function tensor of size $W \times H \times (2^M - 1)$. The procedure of the ENN in Step 2 can be summarized as neural-network layers, called the *evidence layer*, whose details are shown in Section 3.2. The mass function vector in (9) calibrates the confusing and uncertain information about the pixel location. In detail, $m(\{\omega_i\})$, $i = 1, \ldots, M$, characterizes the degree of supports to the hypothesis that the pixel x belongs to ω_i, while $m(A)$ with $|A| > 1$ supports to the hypothesis that the truth is one of the elements in A but the network cannot determine which one. The mass functions of different singleton sets represent the conflict of evidence since the ENN outputs a uniform mass distribution, e.g., $m(\{\omega_i\}) \approx m(\{\omega_j\})$, when different elements of a feature vector support to classes ω_i and ω_j. The value of $m(A)$ with $|A| > 1$ quantifies the ignorance of evidence since the ENN tends to generate a large value if the feature vector contains limited information for decision-making. For example, the ENN outputs $m(\Omega) \approx 1$ if the feature vector is from a backbone without any training and it does not contain any useful information of the segmentation task Tong et al. (2019). The superiority of the uncertainty calibration will be demonstrated by the experiments of distress

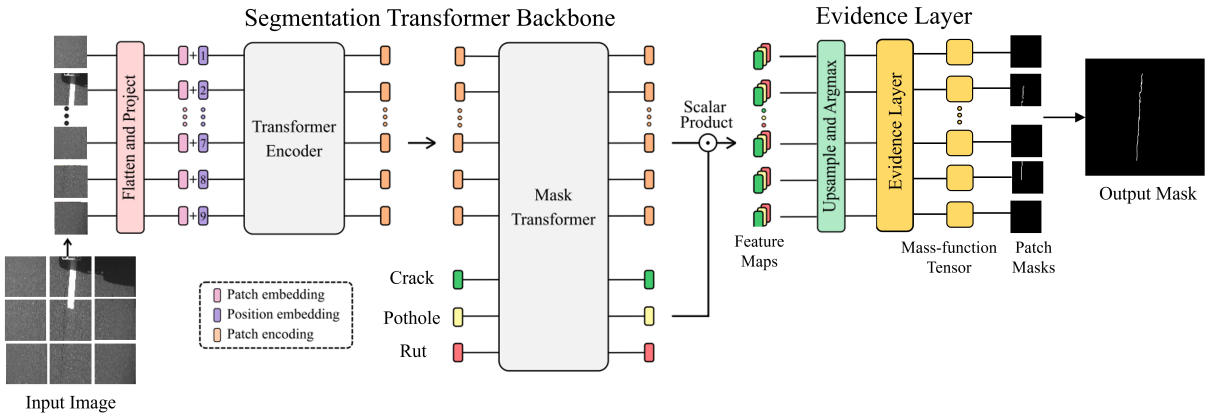

FIGURE 2 Architecture of the evidence layer.

segmentation in Section 4. Besides, the mass-function outputs make it possible to combine various transformers trained by heterogeneous learning sets with different distress classes, as introduced in Section 3.4.

Step 3: The mass-function tensor in Step 2 is used for decision-making. There are some criteria of decision-making with mass functions in Denoeux (2019) and their applications in deep learning Tong, Xu, & Denœux (2021a). As the proposed framework aims to solve the problem of over-confidence, rather than the effects of different decision-making criteria, this step only uses the pignistic criterion Smets (1990) as

$$BetP_{\boldsymbol{m}}(\omega) = \sum_{A\subseteq\Omega,\omega\in A} \frac{m(A)}{|A|}, \quad \forall\omega\in\Omega. \tag{11}$$

The final prediction of the corresponding pixel is $\widehat{\omega} = \arg\max_{\omega\in\Omega} BetP_{\boldsymbol{m}}(\omega)$.

3.2 | Evidence layer

Given a feature vector $\boldsymbol{\psi}(x) \in \mathbb{R}^P$ from a segmentation-transformer backbone, the ENN architecture builds mass functions that calibrate the ignorance and conflict in the vector $\boldsymbol{\psi}(x)$ by mass functions reasoning on $\Omega = \{\omega_1, \ldots, \omega_M\}$. The proposed network performs the processes of the ENN architecture as neural-network layers shown in Figure 3 . The neural-network layers are summarized as follows.

Layer 1: Each evidence $\psi_j(x) \in \boldsymbol{\psi}(x)$ is linearly transformed into a sign τ_{ij} as (4) and the positive and negative parts of τ_{ij} is then converted into $m_{ij}^{+} := \{\omega_i\}^{w_{ij}^{+}}$ and $m_{ij}^{-} := \{\omega_i\}^{w_{ij}^{-}}$. The two masses, m_{ij}^{+} and m_{ij}^{-}, are the degrees of belief supporting to the hypothesis that the true class of pixel x are $\{\omega_i\}$ and $\overline{\{\omega_i\}}$, respectively, based on the evidence $\psi_j(x)$. In the layer, $\alpha_{ij} \in [0,1]$ and $\beta_{ij} \in [0,1]$ in (4) are the parameters associated with evidence $\psi_j(x)$ and class ω_i.

Layer 2: The two mass functions in Layer 1 are separately combined with respect to singleton set $\{\omega_i\}$ using Dempster's rule as (5). Masses m_i^{+} and m_i^{-}, $i = 1, \ldots, M$, are the degrees of belief supporting to class $\{\omega_i\}$ and the frame of discernment Ω, respectively, based on all evidences in $\boldsymbol{\psi}(x)$.

Layer 3: The mass functions supporting to different classes and their individual complementary sets are combined by Dempster's rule as (8). In the multi-class segmentation task, the output mass functions $\boldsymbol{m}_{\boldsymbol{\psi}(x)} = \left(m(A), A \subseteq \Omega\backslash\emptyset\right)^T$ has the following expression:

$$m_{\boldsymbol{\psi}(x)}(\{\omega_i\}) = Q\exp(-w_i^{-})\left\{\exp(-w_i^{+}) - 1 + \prod_{l\neq i}[1-\exp(-w_l^{-})]\right\} \tag{12}$$

for $i = 1, \ldots, M$, and

$$m_{\boldsymbol{\psi}(x)}(A) = Q\left\{\prod_{\omega_i\notin A}[1-\exp(-w_i^{-})]\right\}\left\{\prod_{\omega_i\in A}[\exp(-w_i^{-})]\right\}, \tag{13}$$

for each $A \subseteq \Omega$ such that $|A| > 1$, and Q is a proportionality constant as

$$Q = \eta\eta^{+}\eta^{-}, \tag{14}$$

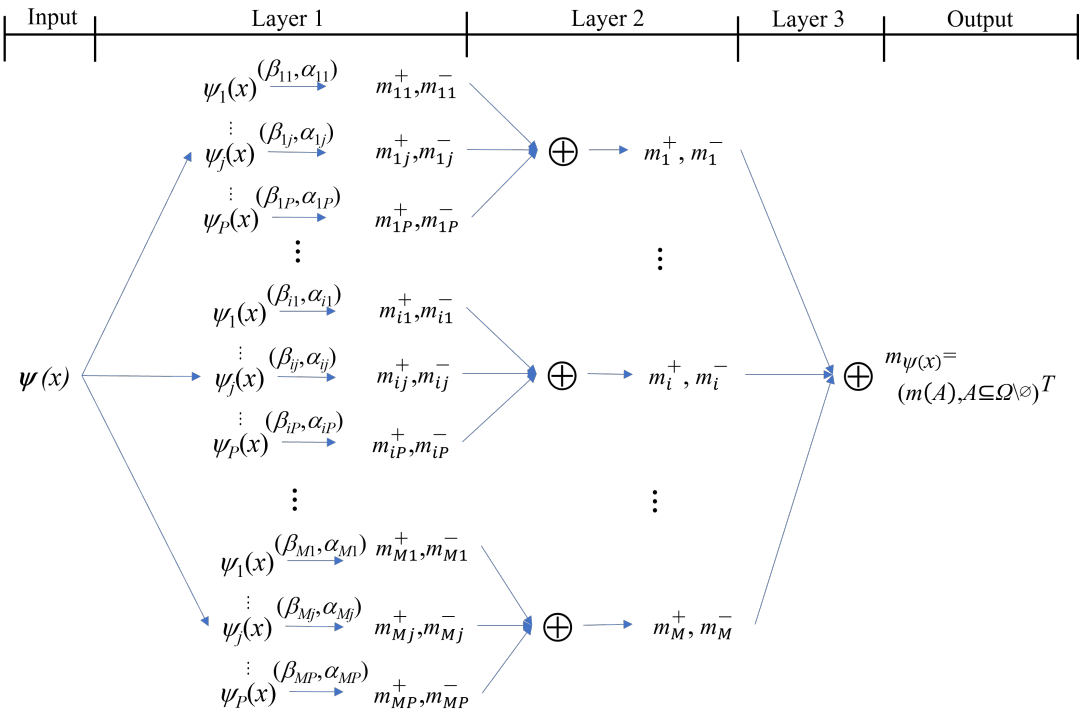

FIGURE 3 Architecture of the evidence layer.

where η is a function w.r.t the degree of conflict (7) between m^- and m^+ as

$$\eta = \frac{1}{1-\kappa} \tag{15a}$$

$$= \frac{1}{1-\sum_{i=1}^{M}\{\eta^+(\exp(w_i^+)-1)[1-\eta^-\exp(-w_i^-)]\}}, \tag{15b}$$

with $\eta^+ = \left(\sum_{i=1}^{M}\exp(w_i^+) - M + 1\right)^{-1}$ and $\eta^- = \left(1-\prod_{i=1}^{M}\left[1-\exp(-w_i^-)\right]\right)^{-1}$. The proof the output mass functions can be found in Appendix A.

3.3 | Learning

Most pavement image datasets annotate each pixel with a single class, though the annotators sometimes do not have full certainty about the true class. For example, in the Pavementscapes dataset, the true classes of the pixels at pavement distress borders may be uncertain but they are given precise labels. Additionally, annotators cannot reliably label some small distress areas in an image, such as some thin cracks. Arbitrarily giving labels with individual singleton classes may introduce incorrect and conflicting information in a learning set, harming the accuracy and reliability of a learning system for pavement distress segmentation.

This study introduces the *soft label* Denœux et al. (2019); Tong et al. (2021b) to deal with the issue. A soft label is defined as a subset $A_* \in 2^\Omega\backslash\emptyset$ that may include the true class based on the knowledge of an annotator. For instance, label $A_* = \{\omega_i, \omega_j\}$ indicate the fact that the ground truth should be either ω_i or ω_j but one is not able to easily decide which one precisely.

A learning strategy has been put forward to update the parameters of the ES-transformer with soft labels, which can reduce the negative effect on annotation uncertainty in the learning set. For a pixel with soft label $A_* \subseteq \Omega\backslash\emptyset$, its *labeling* mass function is logical as $m_l(A_*) = 1$. The *labeling* pignistic probabilities $BetP_{m_l}(\omega)$ can be computed using (11). Then the *predicted* pignistic probabilities $BetP_m(\omega)$ can be computed by the ES-transformer network, where m is the mass-function output in the evidential layer. The loss $\mathcal{L}(m, m_l)$ is defined as the regularized cross-entropy between the pignistic probabilities w.r.t. m_l and m as

$$\mathcal{L}(m, m_l) = -\left(\sum_{i=1}^{M} BetP_{m_l}(\omega_i)\log BetP_{m_l} + \sum_{i=1}^{M} BetP_{m_l}(\omega_i)\log BetP_m(\omega_i)\right) + \lambda\sum_{i'=1}^{M} w_{i'}^2 \tag{16}$$

where λ is the regularization hyperparameter to decrease the effect of weights of evidence $w_{i'}$ to obtain less informative mass functions. The loss $\mathcal{L}(m, m_l)$ is minimized for $BetP_{m_l}(\omega_i) = BetP_m(\omega_i)$, $\forall i = 1, \ldots, M$. When a learning

set only has precise labels, the loss boils down to the common cross-entropy loss. Error propagation and parameter update in the evidence layer and segmentation-transformer backbone can be automatically performed in TensorFlow.

3.4 | Evidential multi-network fusion

One challenge in pavement distress segmentation is to combine the existing models trained from heterogeneous datasets for obtaining a more general one and achieving better results. However, the problem of data uncertainty, especially the partial and imperfect outputs of deep neural networks, makes it difficult to fuse heterogeneous networks. The proposed network provides an approach to solving the problem.

An evidential strategy for multi-transformer fusion has been proposed for pavement distress segmentation. In the approach, several pre-trained ES-transformer networks are combined by adding an information-fusion module at the mass-function outputs of these evidential transformers, as shown in Figure 4 . The architecture, called the "*mass-fusion ES-transformer network (MFES-network)*", can be described as the following steps:

Step 1: A pavement image is fed into V ES-transformer networks, as described in Section 3.1. For the v-th ES-transformer network, $v = 1, \dots, V$, a segmentation-transformer backbone generates pixel-wise feature maps based on the image and an evidence layer then converts the maps into pixel-wise mass functions, where $\boldsymbol{m}^v = (m^v(A_1), \dots, m^v(A_{2^{M(v)}-2}), m^v(\Omega))^T$ is the mass-functions tensor of a pixel location and $M(v)$ is the number of classes in the v-th ES-transformer network. Therefore, this step outputs V mass-functions tensors for each pixel location in the pavement image.

Step 2: The V mass-functions tensors at the same pixel location are aggregated by a mass-function fusion module using Dempster's rule. The V ES-transformer networks may have different sets of classes since they are trained by different learning sets. However, their frames of discernment are still compatible because the "background" or "pavement" classes allow them to have a common refinement, such as the example shown in Figure 5 . The "background" or "pavement" classes have the semantics of "anything else" except for special classes of pavement distresses. Let Ω^0 be a common refinement of the V compatible frames $\Omega^1, \dots, \Omega^V$. Each frame Ω^v can be refined to the common one Ω^0 as

$$\bullet\ \{\rho(\{\omega\}), \omega \in \Omega^v\} \subseteq 2^{\Omega^0}, \tag{17a}$$

$$\bullet\ \forall B \subseteq \Omega^0, \rho(B) = \bigcup_{\omega \in B} \rho(\{\omega\}). \tag{17b}$$

for $v = 1, \dots, V$. The frame Ω^0 is called a *refinement* of Ω^v. The mass m^{Ω^v} in Ω^v can be converted into the *vacuous extension* $m^{\Omega^v \uparrow \Omega^0}$ on Ω^0 as

$$m^{\Omega^v \uparrow \Omega^0}(A) = \begin{cases} m^{\Omega^v}(B) & \text{if } \exists B \subseteq \Omega^v, \quad A = \rho(B), \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

for all $A \subseteq \Omega^0$. Thus, the final outputs of the belief-function fusion module at one pixel location are the mass functions reasoning on Ω^0 as

$$m^{v\uparrow 0} = \bigoplus_{v=1}^{V} m^{\Omega^v \uparrow \Omega^0},$$

representing the total belief of a pixel class according to the feature maps of the V ES-transformer networks.

Step 3: One of the evidence-theoretic rules Denoeux (2019) is selected to make a decision using the aggregated $m^{v\uparrow 0}$. In this work, the pignistic criterion (11) has been used for the segmentation decision-making. Thus, a pixel of the input image is classified to $\widehat{\omega}$, such that

$$\widehat{\omega} = \arg\max_{\omega_i \in \Omega^0} BetP_{m^{v\uparrow 0}}(\omega_i).$$

4 | EXPERIMENTS OF DISTRESS SEGMENTATION

This section presents the numerical experiments, indicating the superiority of the ES-transformer network on pavement distress segmentation. The datasets are first described in Section 4.1, followed by the metrics and implementation details in Sections 4.2 and 4.3, respectively. The results of distress segmentation are reported in Section 4.4.

4.1 | Datasets

Three public pavement image datasets were used in the experiment: Pavementscapes Tong, Ma, Huyan, & Zhang (2022), Crack500 Yang et al. (2019), and CrackDataset Huyan, Li, Tighe, Xu, & Zhai (2020). The datasets were used to train and test the ES-transformers as well as other deep neural networks for comparison.

The Pavementscapes dataset contains six pavement distress classes in 4k images with a resolution of 1024×2048, in which the pixel-wise annotations indicate the classes of all pixels, including one of the six distress categories or "background". In the dataset, the training, validation, and testing sets consist of 2500, 500, and 1000 pavement images. The Crack500 and CrackDataset datasets are similar to the Pavementscapes dataset but only have the "crack" class, respectively, in 500 images with a resolution of 2000 × 1024 and 3k images with

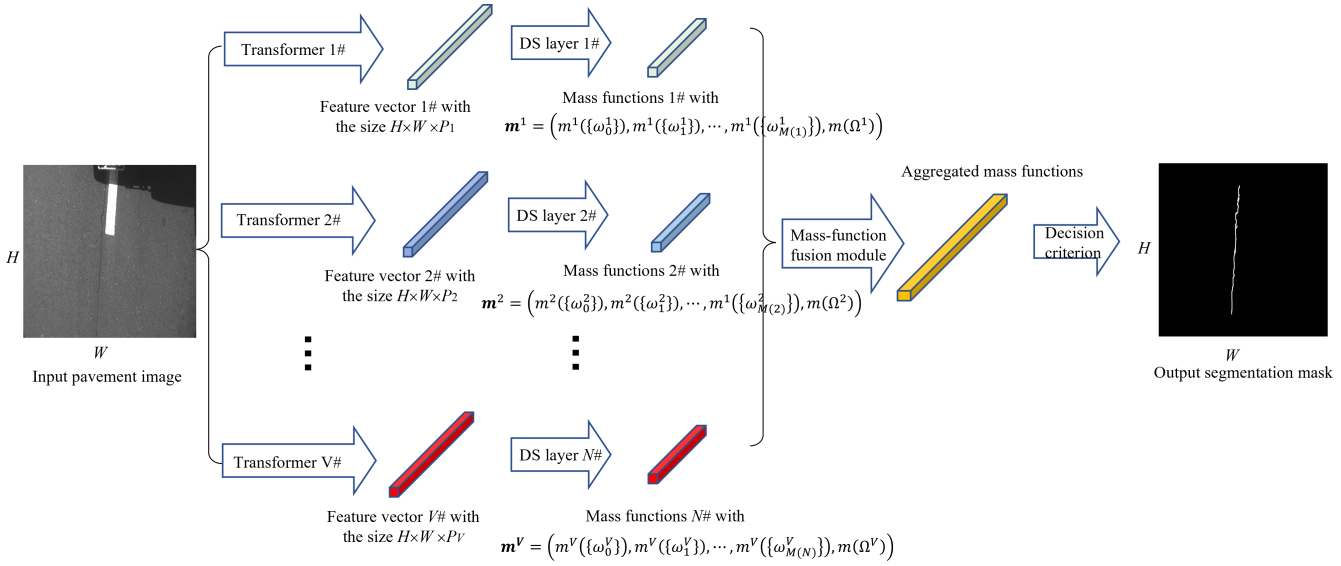

FIGURE 4 Evidential strategy in Dempster-Shafer theory for multi-transformer fusion.

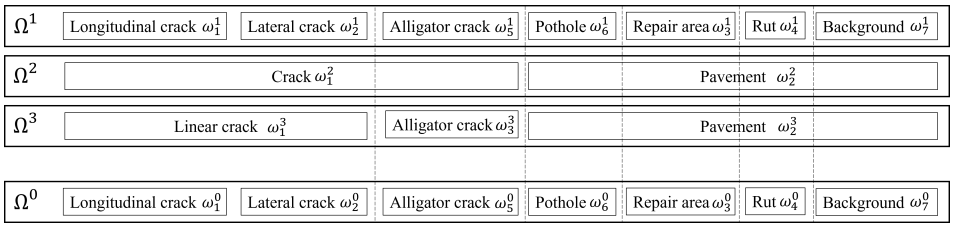

FIGURE 5 A refinement example. The notations Ω^1, Ω^2 and Ω^3 stands for the frames of discernment on the Pavementscape, Crack500, and CrackDataset, and Ω^0 is the common refinement of Ω^1, Ω^2 and Ω^3.

a resolution of 1280×960. The split protocol of the Crack500 dataset is that the 500 images are divided into 250 training images, 50 validation images, and 500 testing images. The CrackDataset dataset consists of 2000 training images and 1000 validation images. In this experiment, the validation set of the CrackDataset dataset has been separated into 500 images as the validation set and 500 images as the testing set. The effectiveness of the proposed network on distress segmentation is evaluated by the three individual held-out testing datasets.

The three datasets do not pre-define soft labels. Thus, this study defines the border pixels of each distress object as soft labels with two or more classes, such as the example in Figure 6 .

4.2 | Metrics

This experiment uses three metrics to measure the capacity of the ES-transformer network for distress segmentation: pixel accuracy (PA), mean intersection over union (mIoU), and expected calibration error (ECE).

Pixel accuracy.

Let $\Omega=\{\omega_1,\ldots,\omega_M\}$ be the set of pavement distress classes. Given an image $\boldsymbol{X}$, the *pixel accuracy* is defined as

$$PA=\frac{1}{|\boldsymbol{X}|}\sum_{x\in\boldsymbol{X}}\mathbb{1}_{A_*(x)}\left(\widehat{\omega}(x)\right), \tag{19}$$

where $|\boldsymbol{X}|$ are the number of pixels in $\boldsymbol{X}$; $A_*(x)$ and $\widehat{\omega}(x)$ are the soft label and predicted class of pixel $x\in\boldsymbol{X}$, respectively; $\mathbb{1}$ is the indicator function, returning one if $\widehat{\omega}(x)\in A_*(x)$, otherwise zero. Note that the pixels belonging to the "background" or "pavement" labels do not consider in the metric, as do in many benchmark datasets Cordts et al. (2016); Lin et al. (2014). Without considering soft labels, PA in this study is the same as its original definition Cordts et al. (2016).

Mean intersection over union.

This metric measures overlap between labeled and predicted areas of distress objects as

$$mIoU=\frac{1}{2^{|\Omega|}-1}\sum_{A_*\subseteq\Omega}\frac{|\boldsymbol{G}(A_*)\cap\boldsymbol{P}(A_*)|}{|\boldsymbol{G}(A_*)\cup\boldsymbol{P}(A_*)|}, \tag{20}$$

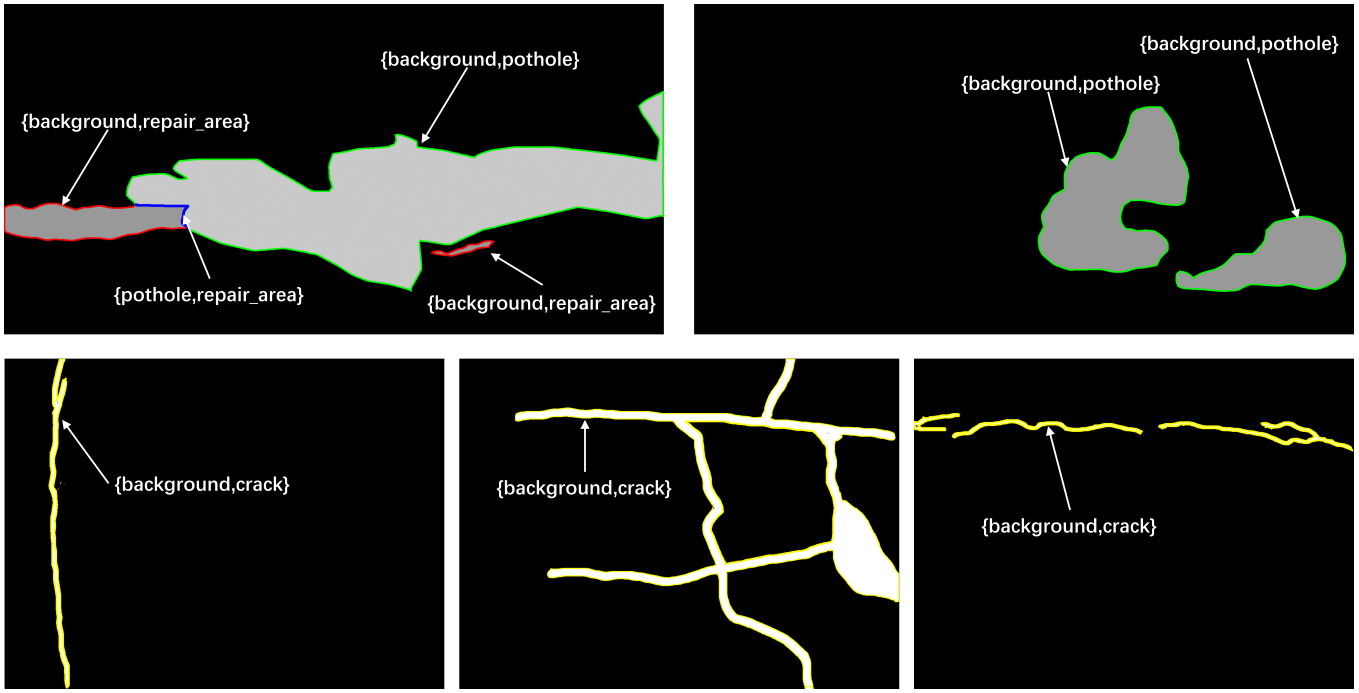

FIGURE 6 Examples of soft labels. The first and second rows are the annotations from the Pavementscapes and CrackDataset datasets. Different colors stands different soft labels.

where $\boldsymbol{G}(A_*)$ is the ground truth areas with label A_*; $\boldsymbol{P}(A_*)$ is the predicted areas whose each pixel is assigned into one of the classes in A_*. The experiments do not consider the IoUs of the "background" and "pavement" labels, as do in many benchmark datasets Cordts et al. (2016); Everingham et al. (2015). Using a dataset without soft labels, mIoU is the same as its original definition for semantic segmentation Long, Shelhamer, & Darrell (2015).

Expected calibration error.

For a learning system, a network needs not only to make correct predictions but also to show the probability that it may fail. *Confidence* in a network is widely used to represent a mass of beliefs supporting the hypothesis that the prediction of a network is correct. Therefore, a desirable transformer network should be measurably-confident and well-calibrated, in which its confidence is close to its accuracy to indicate when the network may fail to predict the pixel classes. The experiments utilize the *expected calibration error* (ECE) Guo et al. (2017) to measure the confidence and calibration performance of a network. First, the *prediction confidence* of pixel x with label $A_*(x)$ is defined as

$$pc(x) = \sum_{\omega \in A_*(x)} \widehat{BetP}_{\boldsymbol{m}}(\omega). \tag{21}$$

where $\widehat{BetP}_{\boldsymbol{m}}(\omega)$ is the predicted pignistic probability of class ω using 11. Let b_k be the bin of pixels with prediction confidences falling into interval $(\frac{k-1}{K}, \frac{k}{K}]$, $k = 1, \ldots, K$. The accuracy and confidence of bin b_k are then computed, respectively, as

$$ac(b_k) = \frac{1}{|b_k|} \sum_{x \in b_k} \mathbb{1}_{A_*(x)} \left(\widehat{\omega}(x)\right), \tag{22a}$$

$$co(b_k) = \frac{1}{|b_k|} \sum_{x \in b_k} pc(x). \tag{22b}$$

A network is well calibrated with $ac(b_k) \approx co(b_k)$ for all bins, and the ECE is defined as

$$ECE = \frac{\sum_{k=1}^{K} |b_k| \times |co(b_k) - ac(b_k)|}{\sum_{k'=1}^{K} |b'_k|}. \tag{23}$$

The ECE in the experiments does not consider the "background" and "pavement" pixels.

4.3 | Implementation details

Hyper-parameters

In the segmentation-transformer backbone, the experiments use the vision transformer (ViT) Dosovitskiy et al. (2021) to design different encoders, as shown in Table 1 . The head size of a multi-head self-attention block is fixed to 64 in the encoders, while the number of heads is the ratio of the token size over the multiplication of the head size and the hidden size

TABLE 1 Implementation details of segmentation transformer backbones.

Backbone	Encoder	Patch size	Layers	Token size	Heads	Params
Seg-T/16	ViT-Ti	16	12	192	3	6M
Seg-S/32	ViT-S	32	12	384	6	22M
Seg-S/16	ViT-S	16	12	384	6	22M
Seg-B/32	ViT-B	32	12	768	12	86M
Seg-B/16	ViT-B	16	12	768	12	86M
Seg-B/8	ViT-B	8	12	768	12	86M
Seg-L/16	ViT-L	16	24	1024	16	307M

in the multi-layer perceptron. The experiments also consider different patch sizes 8×8, 16×16, and 32×32. For the decoder part, the point-wise linear layers and mask transformer upsample the outputs of the encoder part to generate the pixel-wise features.

In the evidence layer, the dimensions of the input feature vectors are 52, 15, and 20, respectively, for the Pavementscapes, Crack500, and CrackDataset datasets. A convolutional layer with 1×1 kernels is plugged between the backbone and evidential layer to adjust the dimensions of the features vectors.

Training setting.

Before training, the weights of each ViT encoder in the segmentation-transformer backbone are initialized by the ImageNet pre-trained weights in Dosovitskiy et al. (2021), while all parameters in the decoder part and evidential layer are initialized randomly using normal distributions.

During training, as done in many segmentation tasks, data augmentation is adopted in the three datasets, including mean subtraction, random reshaping with a proportion from 0.5 to 2.0, and so on. For the three distress segmentation tasks, the proposed networks are fine-tuned with the batch size of 4 and the"poly" learning rate $\gamma = \gamma_0(1 - \frac{N_i}{N})$, where γ_0 is the base learning rate; N_i and N are the current epoch number and the total epoch number. The values of γ_0 for all three datasets equal 10^{-3}, while the total epoch number are 600, 100, and 150, receptively, for the Pavementscape, Crack500, and CrackDataset datasets.

4.4 | Results of distress segmentation

Pavementscapes

Table 2 shows the results of the proposed networks and other comparison networks in the testing set of the Pavementscapes dataset. Adding the evidence layer, an ES-transformer network exceeds the PS-transformers with the same backbone in terms of PA and mIoU, demonstrating the evidence layer can slightly improve the distress segmentation performance by improving the depth of a network. In addition, the ES-transformer network with the Seg-L/16 backbone achieves the best performance, outperforming the convolution-based deep neural networks. Also, compared to the evidential networks in Tong et al. (2021b), the proposed ones have the superiority in the terms of PA and mIoU owing to the use of an advanced evidential neural network that does not measure the distance between each prototype and the feature vector. All in all, the ES-transformers have superiority in segmentation accuracy.

Table 2 also indicates that the proposed networks are more cautious than the probabilistic ones with the same backbone because the proposed networks have lower ECE values than the probabilistic ones. This demonstrates that the accuracy of an ES-transformer network matches its confidence but the PS-transformer network cannot. Figure 7 provides an example to explain the behavior. The accuracies of the PS-transformer network in different bins are significantly lower than its confidences, while the ES-transformer network is cautious to make a decision. This behavior mainly benefits from the uncertainty calibration by the evidence layer. When the features from the segmentation-transformer backbone have some confusing and conflicting information, an evidence layer tends to output the uniform values of some single mass functions, such as $m(\{\omega_i\}) \approx m(\{\omega_j\})$ and a large values of $m(A)$ with $|A| > 1$, indicating the true class may fall in set A. Figure 8 presents an example where the ES-transformer network cannot confidently determine the pixel belonging to lateral or longitudinal cracks. When the two lateral cracks in Figure 8 rotate 45°, the proposed network has $m(\{\omega_1\}) \approx m(\{\omega_2\})$ and $m(\{\omega_1, \omega_2\}) \approx 0.21$ in the green box, indicating the features from the segmentation-transformer backbone are confusing and the network cannot determine the pixel classes between "lateral crack" and "longitudinal crack". Appendix A proves why the evidence layer has the capacity. Thus, the cautious mass-function outputs make the proposed networks well-calibrated. Unfortunately, the PS-transformer networks in the probability framework cannot calibrate the uncertainty information well since they always arbitrarily assign a large probability to one and only one possible class.

Table 2 indicates that, given the same backbone, an ES-transformer network trained by the learning set with soft labels has higher PAs, mIoUs, and ECEs than the ones without soft labels. Thus, the proposed learning approach also increases the calibration of the proposed network. This is mainly because the soft labels in the mass-function form allow a network to adapt some confusing and conflicting information, rather than arbitrarily learning. In addition, the results demonstrate that the hard but imprecise labels in the Pavementscape dataset have negative effects on a learning system. Here, "hard" means labeling a sample as one and only one class, while "imprecise" means making "hard" labels with uncertainty.

TABLE 2 State-of-the-art comparison in the Pavementscapes testing set. The notation "-sl" means the network is trained by the learning set with soft labels. GFLOPs stand for 10^9 (Giga) floating point operations. The best and second results in the three metrics (PA, mIoU, and ECE) are in bold and italics. The first part of the table is the convolution-based methods and the rest are the attention-based ones.

Method	Backbone	PA	mIoU	ECE	GFLOPs
FCN-8s Long et al. (2015)	VGG16	67.32	52.98	22.30	136.2
U-net Ronneberger, Fischer, & Brox (2015)	VGG16	69.56	54.71	22.14	72.4
DeepLabv3+ L.-C. Chen, Papandreou, Kokkinos, Murphy, & Yuille (2017)	VGG16	71.90	57.51	21.81	457.6
Self-Attention net Vaswani et al. (2017)	-	73.07	58.74	21.32	38.5
CC-Attention net Huang et al. (2019)	-	73.15	58.52	21.14	508.4
Double-Attention net Y. Chen, Kalantidis, Li, Yan, & Feng (2018)	-	74.01	59.23	21.10	21.4
PS-Transformer Strudel et al. (2021)	Seg-T/16	73.82	59.12	21.09	18.9
PS-Transformer	Seg-S/32	73.65	58.93	20.98	-
PS-Transformer	Seg-S/16	74.23	59.42	20.79	23.8
PS-Transformer	Seg-B/32	74.10	59.24	20.84	-
PS-Transformer	Seg-B/16	73.51	58.82	20.63	85.8
PS-Transformer	Seg-B/8	73.69	58.84	21.05	-
PS-Transformer	Seg-L/16	74.50	59.74	20.95	306.8
E-Transformer Tong et al. (2021b)	Seg-L/16	72.14	56.32	18.43	307.1
ES-Transformer (Proposed)	Seg-T/16	73.9	59.23	17.42	19.1
ES-Transformer	Seg-S/32	73.77	58.71	17.26	-
ES-Transformer	Seg-S/16	74.18	59.82	17.32	24.1
ES-Transformer	Seg-B/32	73.98	59.46	17.60	-
ES-Transformer	Seg-B/16	73.65	58.71	17.72	86.2
ES-Transformer	Seg-B/8	73.83	58.33	17.34	-
ES-Transformer	Seg-L/16	*75.64*	*60.32*	17.27	307.0
E-Transformer-sl Tong et al. (2021b)	Seg-L/16	72.18	57.14	18.23	307.1
ES-Transformer-sl (Proposed)	Seg-T/16	73.93	59.29	17.21	19.1
ES-Transformer-sl	Seg-S/32	73.79	58.73	*17.12*	-
ES-Transformer-sl	Seg-S/16	74.14	59.85	17.09	24.1
ES-Transformer-sl	Seg-B/32	7387	59.49	17.49	-
ES-Transformer-sl	Seg-B/16	73.72	58.74	17.61	86.2
ES-Transformer-sl	Seg-B/8	73.89	58.39	17.24	-
ES-Transformer-sl	Seg-L/16	**75.67**	**60.34**	**17.04**	307.0

Figure 9 shows the mIoU results of the ES-Transformer-sl model with the Seg-L/16 backbone under various real-world conditions, as well as the results of some previous networks. The proposed network has stable performance on different weather and pavement surface materials, exceeding the other probabilistic deep networks. This demonstrates that the proposed networks are stable under various real-world conditions.

Floating point operations (FLOPs) are used to measure how many operations are required to run a single instance in a deep neural network. A lower value of FLOPs always means that an algorithm processes a new instance with fewer computation costs. Table 2 shows that the use of an evidential layer does not introduce significant computation costs but increases the accuracy and calibration for the distress segmentation task using the same backbone. For example, the ES-transformer with the Seg-L/16 backbone has higher mIoU but similar FLOPs than the PS-transformer network with the same backbone.

Crack500 and CrackDataset

Tables 3 and 4 present the testing results on the Crack500 and CrackDataset datasets, respectively. The ES-transformer network with the Seg-L/16 backbone has the optimal PA, mIoU, and ECE, followed by the other transformers. Even though the probabilistic transformers have similar PAs and mIoUs as the evidence ones with the same backbone, they have larger ECEs, indicating that the probabilistic models are overconfident. Therefore, the use of an evidence layer at the end of the segmentation-transformer backbone can improve the accuracy and calibration of the transformer networks. As the

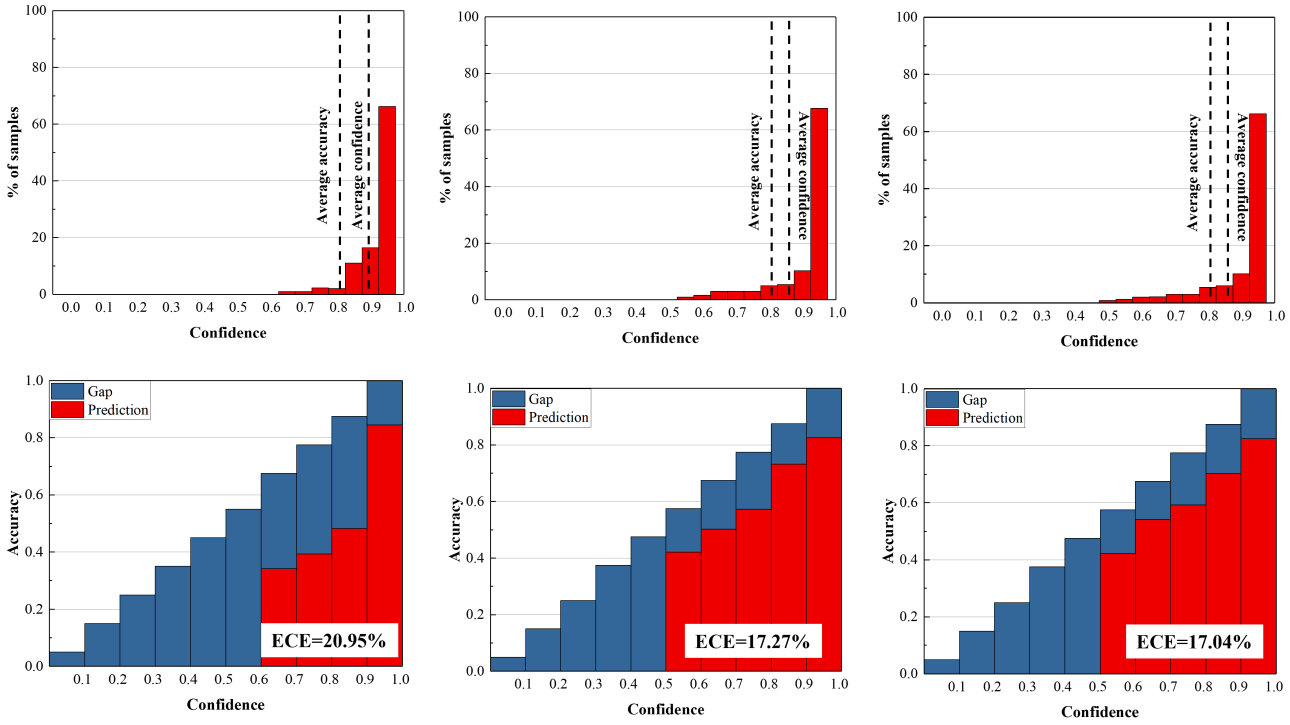

FIGURE 7 Prediction confidences (top) and pixel accuracies (bottom) in different bins of pixels for PS-transformer (left), ES-transformer (mid), and ES-transformer-sl (right) on the Pavementscapes dataset.

three used datasets (Pavementscapes, CrackTree, and CrackDataset) were generated from different image color schemes, resolutions, types of image collection methods, and so on, the proposed method with high performances is stable in real-world conditions. In addition, the proposed method does not introduce significant computation costs for the distress segmentation tasks.

5 | EXPERIMENT OF MULTI-NETWORK FUSION

The experiment of multi-network fusion is presented in the section, including the implementation details in Section 5.1 and the main findings in Section 5.2.

5.1 | Implementation details

Dataset.

The three datasets in Section 4.1 are merged into one for the experiment. The merged dataset includes the training set of 4000 images, the validation set of 1050 images, and the testing set of 2000 images. The semantics of the distress classes in the three datasets are shown in Figure 5 . After merging the three datasets, the three different frames are refined into a common one Ω^0 using 17.

Metrics.

After merging, some labels in the three datasets become soft labels. For example, a "crack" label from the Crack500 dataset is a soft label in the refined frame Ω^0 since it indicates the corresponding pixel belongs to one type of crack but one cannot determine which one. In such a case, Eq. (19) is used to evaluate the *average accuracy* of the proposed approach after network fusion in the testing set.

Implementation details

The three well-trained ES-transformers with the best performance in Tables 2 -4 are fused using the proposed evidential fusion framework in Section 3.4. This study compares the proposed fusion framework with the other four methods using the same transformers.

Probability-to-mass fusion (PMF) Diaby et al. (2021) use the feature vector of a pixel location and a softmax layer to compute the Bayesian probabilities reasoning on Ω^v, $v = 1, \ldots, V$. The Bayesian probabilities are extended into the Bayesian mass functions on the common frame Ω^0 and aggregated by Dempster's rule.

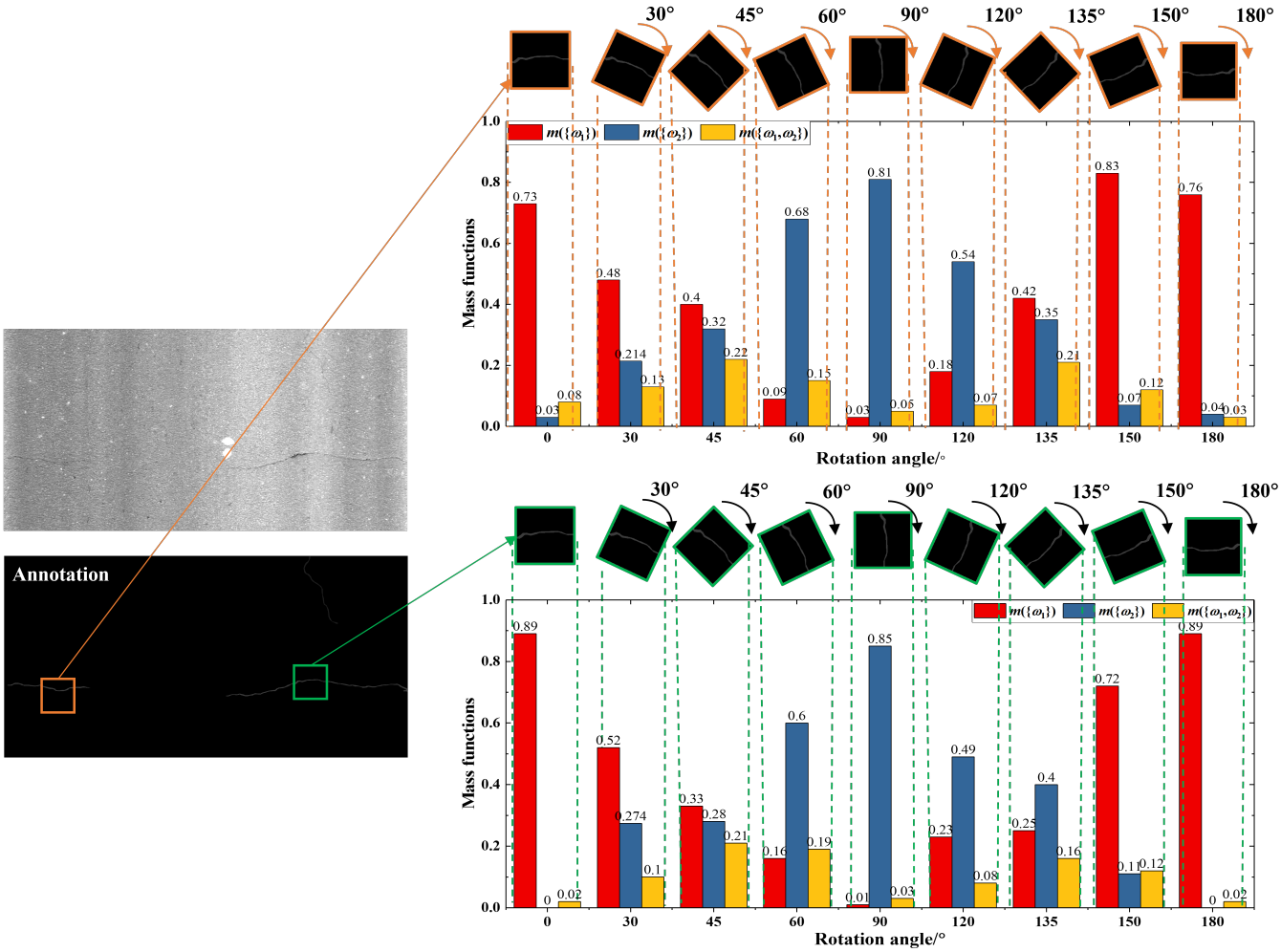

FIGURE 8 Example of uncertainty calibration using mass functions. The green and orange boxes present two lateral cracks. With the rotation of the two boxes, the mass functions $m(\{\omega_1\})$, $m(\{\omega_2\})$, and $m(\{\omega_1, \omega_2\})$ change, where ω_1 and ω_2 stand for lateral and longitudinal cracks, respectively. The values of $m(\{\omega_1\})$, $m(\{\omega_2\})$, and $m(\{\omega_1, \omega_2\})$ are the averaged mass functions of the pixels belonging the lateral cracks.

Bayesian-fusion (BF) Xu, Davoine, Bordes, Zhao, & Denœux (2016) converts the feature vector of a pixel location into a Bayesian probabilities distribution on the common frame Ω^0, where the probability of a multi-class set is equally assigned to the elements of the set. The Bayesian probabilities distributions from different networks are then aggregated by Bayes' theorem.

Probabilistic feature-combination (PFC) L. D. Nguyen, Lin, Lin, & Cao (2018) concatenates the three feature vectors of a same pixel location, which are generated from three different networks, to build a new vector of dimension 1024×3. The concatenated vector is then used to compute the probability distribution of the class membership on the common frame by a softmax layer.

Evidential feature-combination (EFC) also concatenates the three vectors into one, but the concatenated vector is imported into an evidence layer to compute a mass-function distribution on the common frame.

5.2 | Results of multi-model fusion

Table 5 presents the pixel accuracies of the ES- and PS-transformer networks trained by one of the three datasets, along with the results of the four information fusion approaches.

The proposed MFES strategy increases the average pixel accuracy on the Crack500 and CrackDataset datasets after fusion, as shown in Table 5 . This is because an ES-transformer network trained by the Pavementscapes dataset

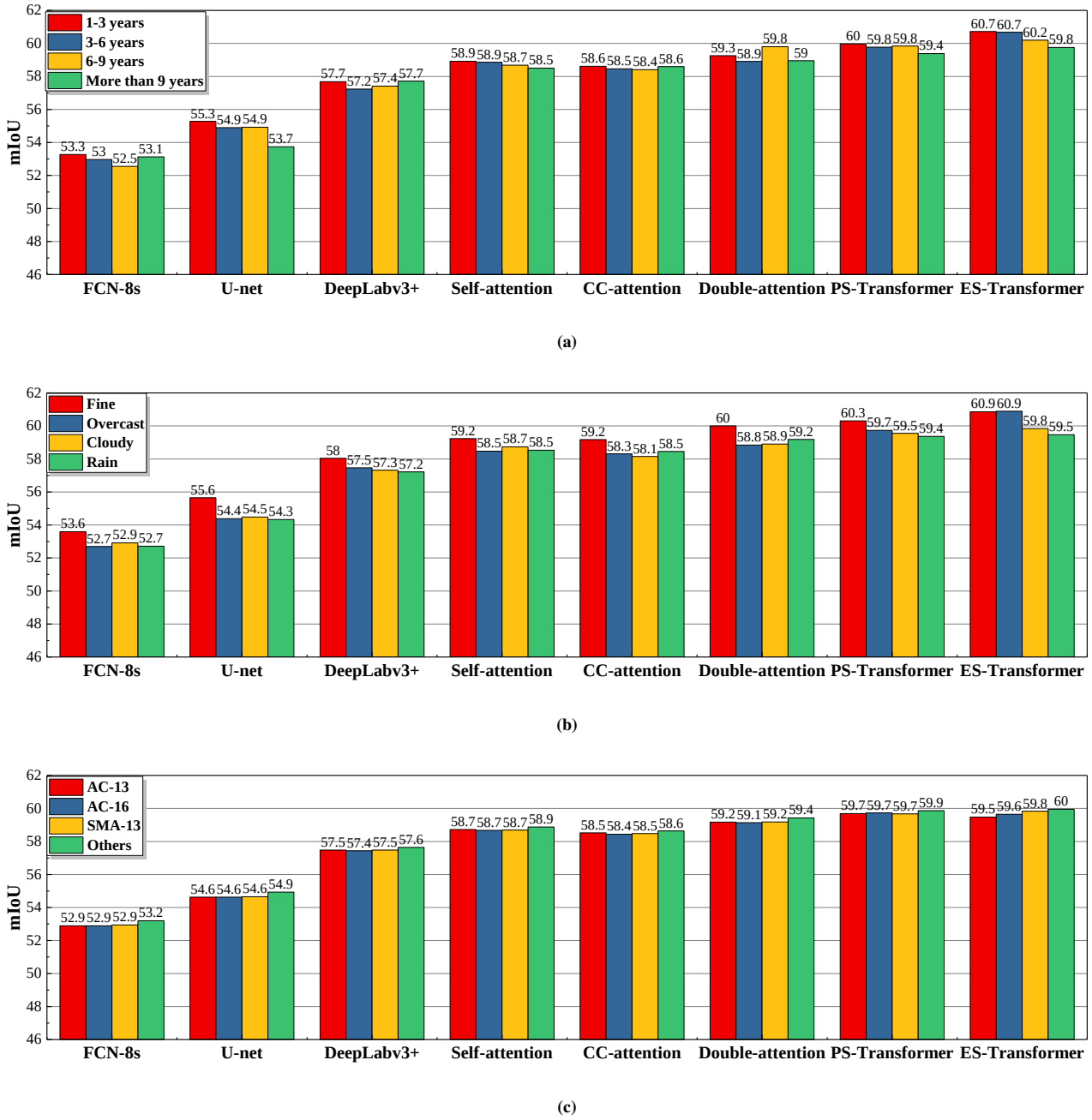

FIGURE 9 Stability analysis using the Pavementscapes dataset under different (a) service years, (b) weathers, and (c) pavement materials.

can provide useful and detailed information for the crack segmentation on the other two datasets. For example, crack pixels are misclassified into the "background" class when only using a CrackData ES-transformer network, as shown in the two examples in Figures 10 a and 10 b. After fusing the information from the Pavementscapes and Crack500 networks, the two misclassifications are corrected.

Table 5 also shows that there is a small increase in the pixel accuracies in the Pavementscapes testing set. The reason for the increase in the Pavementscapes dataset is a little different from the ones for the other two datasets. The mass functions from the Crack500 and CrackDataset networks can provide a little useful information for the Pavementscapes network, such as the examples in Figures 10 c and 10 d. The two pixels are misclassified into the "longitudinal crack" class

TABLE 3 State-of-the-art comparison on the Crack500 testing set. The notation "-sl" means the network is trained by the learning set with the soft labels. GFLOPs stand for 10^9 (Giga) floating point operations. The best and second results in the three metrics (PA, mIoU, and ECE) are in bold and italics. The first and second parts are the convolution- and attention-based methods.

Method	Backbone	PA	mIoU	ECE	GFLOPs
FCN-8s Long et al. (2015)	VGG16	67.82	50.31	25.30	194.76
FCN Yang et al. (2019)	FPHBN	70.81	54.51	23.56	207.32
Unet N. T. H. Nguyen, Le, Perry, & Nguyen (2018)	VGG16	69.01	52.61	25.17	103.53
Unet Lau, Chong, Yang, & Wang (2020)	ResNeSt-50	73.77	57.82	18.24	125.76
Split-attention network H. Zhang et al. (2022)	ResNeSt	72.97	57.42	19.36	309.42
Pyramid-attention network Wang & Su (2020)	DenseNet121	79.85	62.35	17.62	206.31
PS-transformer Strudel et al. (2021)	Seg-L/16	82.45	<u>65.82</u>	16.35	438.72
E-transformer Tong et al. (2021b)	Seg-L/16	82.16	65.14	17.30	439.15
ES-transformer (Proposed)	Seg-L/16	<u>82.78</u>	**65.98**	<u>12.38</u>	439.01
ES-transformer-sl (Proposed)	Seg-L/16	**82.84**	65.79	**11.24**	439.01

TABLE 4 State-of-the-art comparison on the CrackDataset testing set. The notation "-sl" means the network is trained by the learning set with the soft labels. GFLOPs stand for 10^9 (Giga) floating point operations. The best and second results in the three metrics (PA, mIoU, and ECE) are in bold and italics. The first and second parts are the convolution- and attention-based methods.

Method	Backbone	PA	mIoU	ECE	GFLOPs
FCN-8s Long et al. (2015)	VGG16	71.19	60.24	14.32	68.1
U-net N. T. H. Nguyen et al. (2018)	VGG16	79.87	65.32	10.63	36.2
CrackU-net Huyan et al. (2020)	VGG16	98.14	83.43	5.89	36.2
PS-transformer Strudel et al. (2021)	Seg-L/16	99.24	86.32	5.53	153.4
E-transformer Tong et al. (2021b)	Seg-L/16	96.31	83.26	6.17	153.5
ES-transformer (Proposed)	Seg-L/16	<u>99.29</u>	**86.51**	<u>2.97</u>	153.3
ES-transformer (Proposed)	Seg-L/16	**99.32**	<u>86.47</u>	**2.64**	153.3

when the truth is "repair area". The Crack500 and CrackDataset networks highly believe that the two pixels belong to the "crack" class. After aggregating the three mass functions, the two pixels are classified correctly. Therefore, the Crack500 and CrackDataset networks can provide useful information on the crack super-class for the Pavementscapes network. However, the mass functions from the Crack500 and CrackDataset networks do not help when pixels are misclassified into other distress classes by the Pavementscapes network, such as the instance in Figure 10 e. All in all, these observations show that the MFES strategy allows combining the transformers trained by different datasets with different class sets to generate a more general and accurate one, without introducing negative effects on the accuracy of the individual networks, and sometimes may increase results for some classes. Besides, this strategy does not require extra training costs.

TABLE 5 Test average accuracy of different fusion strategies. The best and second results in the each column are in bold and italics.

	Method	Pavementscapes	Crack500	CrackDataset
Before fusion	ES-transformer	<u>75.67</u>	<u>82.84</u>	99.32
	PS-transformer	74.50	82.45	99.24
After fusion	MEFS	**75.79**	**84.06**	**99.62**
	PMF	74.60	82.74	<u>99.41</u>
	BF	74.58	82.85	99.36
	PFC	74.21	82.23	99.00
	EFC	71.06	78.32	94.26

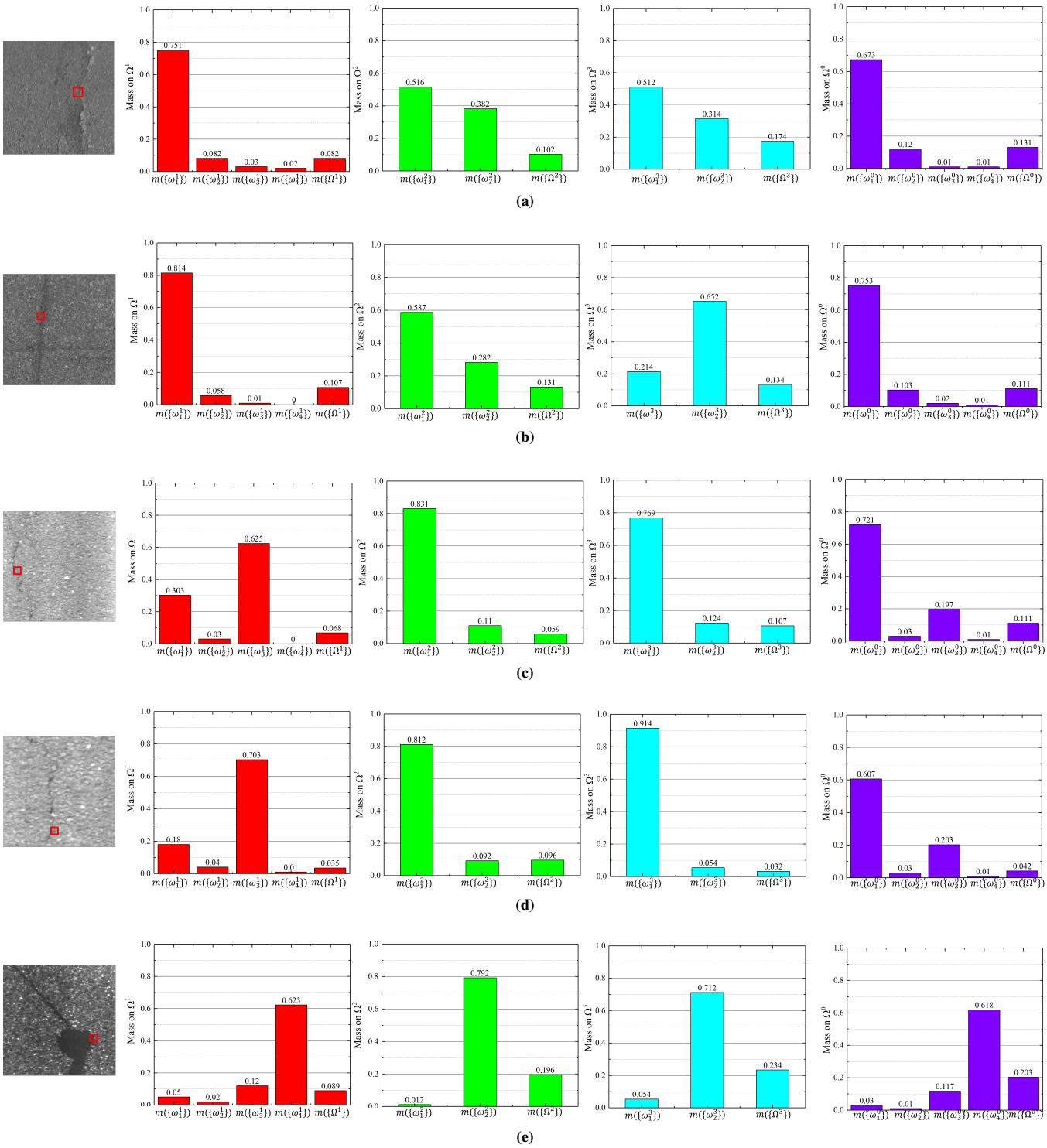

FIGURE 10 Mass-function fusion instances using MFES approach. The notations in the figure are the same as the ones in Figure 5 . Only some mass functions of the red pixels are shown for lack of space.

Table 5 indicates that the PMF and BF approaches also improve the performance of the PS-transformer networks after fusion, though they are not as good as the proposed one. The relatively low accuracies of the EFC and PFC strategies demonstrate that the simple feature-concatenation methods are less effective and have more training costs than the proposed ones. In summary, the proposed evidential fusion strategy

exceeds the other methods of multi-network fusion for distress segmentation.

6 | CONCLUSIONS

This study has proposed a new transformer network in the framework of DST for pavement distress segmentation. The following conclusions can be drawn from the presented results.

- The main finding of this study is that the ES-transformer network increases the accuracy and calibration of transformers by representing uncertainty in the form of DST mass functions.
- The state-of-the-art performance of the proposed network shows a new way to increase the performance of the segmentation transformer on distress segmentation by cautious DST-based decision-making.
- A learning approach has been proposed to train the ES-transformer network, which handles the ambiguous pixels with soft labels. This approach provides a way to solve the problem of over-confidence in a transformer by representing the knowledge of label uncertainty in a learning set.
- The mass-functions outputs of the proposed network allow to fuse of heterogeneous transformers with different distress categories, without introducing negative effects on the accuracy of the individual networks, and sometimes may increase results for some classes. The approach does not require extra training.
- The ES-transformer network faces the problem of an unbalanced learning set, especially on the Pavementscapes dataset, in which the numbers of different distress classes are very different. This phenomenon harms a learning system and some advanced loss functions should be considered to reduce the effect. In addition, more large public datasets should be used to demonstrate the advantages of the proposed network.

ACKNOWLEDGMENTS

This research was supported by National Key Research and Development Project (grant number 2020YFA0714302) and National Key Research and Development Project (grant number 2020YFB1600102).

SUPPORTING INFORMATION

The following supporting information is available as part of the online article:

Pavementscape: The dataset is available at `https://github.com/tongzheng1992/Pavementscapes`.

Crack500: The dataset is available at `https://github.com/fyangneil/pavement-crack-detection`.

CrackDataset: The dataset is available at `https://github.com/juhuyan/CrackDataset_DL_HY`.

Code availability: The code of the ES-transformer will be available at `https://github.com/tongzheng1992?tab=repositories` soon. The code of ECE is available at `https://github.com/tongzheng1992/E-FCN`, which has been released by the first author in his previous study Tong et al. (2021b).

How to cite this article: Tong Z., Ma T., Zhang W., Huyan J., , Evidential transformer for pavement distress segmentation. *Submitted to the journal of Computer-Aided Civil and Infrastructure Engineering* .

APPENDIX

A PROOF OF OUTPUT MASS FUNCTIONS IN THE EVIDENCE LAYER

The Layers 2 and 3 in Sexton 3.2 combine two or more simple mass functions by combining their weights of evidence following the definition of simple mass function in Denœux et al. (2019). Let's begin with the mass m^+. Each m_i^+ is the simple mass functions with two focal set $\{\omega_i\}$ and Ω. Thus, the positive mass m^+ is computed as

$$\begin{aligned} m^+(\{\omega_i\}) &\propto [1-\exp(-w_i^+)]\prod_{l\neq i}\exp(-w_l^+) \\ &= \prod_{l\neq i}\exp(-w_l^+) - \prod_{l=1}^{M}\exp(-w_l^+) \\ &= [\exp(w_i^+)-1]\exp(-\sum_{l=1}^{M} w_l^+) \end{aligned} \tag{A1a}$$

with

$$m^+(\Omega) \propto \exp(-\sum_{l=1}^{M} w_i^+). \tag{A1b}$$

Therefore, we can get

$$\sum_{i=1}^{M} m^{+}(\{\omega_i\}) + m^{+}(\Omega) \propto \exp(-\sum_{l=1}^{M} w_l^{+}) \left[\sum_{i=1}^{M} \exp(w_l^{+}) - M + 1\right]. \quad \text{(A2)}$$

Then the positive masses can be normalized as

$$m^{+}(\{\omega_i\}) = \frac{\exp(w_i^{+}) - 1}{\sum_{l=1}^{M} \exp(w_l^{+}) - M + 1} \textbf{ for } i = 1, \ldots, M \quad \text{(A3a)}$$

$$m^{+}(\Omega) = \frac{1}{\sum_{l=1}^{M} \exp(w_l^{+}) - M + 1}. \quad \text{(A3b)}$$

The mass function $m^{+}(\Omega)$ is a decreasing function that is close to one with $w_i^{+} \approx 0,\ i = 1, \ldots, M$. This indicates that the evidence layer tends to generate a value of $m^{+}(\Omega)$ close to 1 if any evidence $\psi_j(x) \in \boldsymbol{\psi}(\boldsymbol{x})$ cannot provide any support for the true class. This is the ignorance that the evidence vector does not contain any useful information. In addition, the evidence layer tends to generate an uniform mass-function distribution among $m^{+}(\{\omega_i\}),\ i = 1, \ldots, M$, when different evidences has similar weights. This is the confusion that the evidence vector contains much conflict information.

Similarly, following (5b), the negative mass function of each $A \subset \Omega$ is computed as

$$m^{-}(A) = \frac{\left\{\prod_{\omega_i \notin A}[1 - \exp(w_i^{-})]\right\}\left\{\prod_{\omega_i \in A} \exp(-w_i^{-})\right\}}{1 - \kappa^{-}} \quad \text{(A4a)}$$

$$m^{-}(\Omega) = \frac{\exp\left(-\sum_{i=1}^{M} w_i^{-}\right)}{1 - \kappa}, \quad \text{(A4b)}$$

where the degree of conflict κ^{-} is computed using (7) as

$$\kappa^{-} = \prod_{i=1}^{M} \left[1 - \exp(-w_i^{-})\right]. \quad \text{(A4c)}$$

After getting m^{+} and m^{-} in (A3) and (A4), respectively, the evidence layers combine them into one. Following (7), the degree of conflict between m^{+} and m^{-} is

$$\kappa = \sum_{i=1}^{M} \left\{ m^{+}(\{\omega_i\}) \sum_{\omega \notin A} m^{-}(A) \right\} = \sum_{i=1}^{M} \left\{ m^{+}(\{\omega_i\}) \left(1 - \frac{\exp(-w_i^{-})}{1 - \prod_{i'=1}^{M}[1 - \exp(-w_{i'}^{-})]}\right)\right\}. \quad \text{(A5)}$$

With $\eta^{+} = \left(\sum_{i'=1}^{M} \exp(w_{i'}^{+}) - M + 1\right)^{-1}$ and $\eta^{-} = \left(1 - \prod_{i'=1}^{M}[1 - \exp(-w_{i'}^{-})]\right)^{-1}$, Eq. (A5) can be simplified as

$$\kappa = \sum_{i=1}^{M} \left\{\eta^{+}(\exp(w_i^{+}) - 1)[1 - \eta^{-} \exp(-w_i^{-})]\right\}. \quad \text{(A6)}$$

Using Dempster's rule, the mass of each singleton set is computed as

$$m(\{\omega_i\}) = \frac{m^{+}(\{\omega_i\})\left[\sum_{\omega_i \notin A} m^{-}(A)\right] + m^{-}(\{\omega_i\})m^{+}(\Omega)}{1 - \kappa} \quad \text{(A7)}$$

Using (A3) and (A4), Eq. (A7) can be re-written as

$$m(\{\omega_i\}) \quad \text{(A8)}$$

$$= \eta\eta^{-}\eta^{+} \exp(-w_i^{-}) \left\{ \exp(w_i^{+}) - 1 + \prod_{l \neq i}[1 - \exp(-w_l^{-})] \right\}, \quad \text{(A9)}$$

with $\eta = (1 - \kappa)^{-1}$. Using (A4a) and (A3b), the mass for each multi-element set $|A| \subseteq \Omega$ is computed as

$$m(A) = \eta m^{-}(A) m^{+}(\Omega) \quad \text{(A10)}$$

$$= \eta\eta^{-}\eta^{+} \left\{ \prod_{\omega_i \notin A}[1 - \exp(-w_i^{-})] \right\} \left\{ \prod_{\omega_i \in A} \exp(-w_i^{-}) \right\}. \quad \text{(A11)}$$

Finally, Eqs. A8 and A10 are the output mass functions of the evidence neural-network layers.

References

Arabi, S., Haghighat, A., & Sharma, A. (2020). A deep-learning-based computer vision solution for construction vehicle detection. *Computer-Aided Civil and Infrastructure Engineering*, *35*(7), 753–767.

Bang, S., Hong, Y., & Kim, H. (2021). Proactive proximity monitoring with instance segmentation and unmanned aerial vehicle-acquired video-frame prediction. *Computer-Aided Civil and Infrastructure Engineering*, *36*(6), 800–816.

Chen, F.-C., & Jahanshahi, M. R. (2017). Nb-cnn: Deep learning-based crack detection using convolutional neural network and naïve bayes data fusion. *IEEE Transactions on Industrial Electronics*, *65*(5), 4392–4400.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, *40*(4), 834–848.

Chen, Y., Kalantidis, Y., Li, J., Yan, S., & Feng, J. (2018). Aˆ 2-nets: Double attention networks. *Advances in neural information processing systems*, *31*.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., … Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proc. of the ieee conference on computer vision and pattern recognition* (pp. 248–255).

Dempster, A. (1967). Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, *38*, 325–339.

Denoeux, T. (2019). Decision-making with belief functions: a review. *International Journal of Approximate Reasoning*, *109*, 87–110.

Denœux, T. (2019). Logistic regression, neural networks and Dempster-Shafer theory: A new perspective. *Knowledge-Based Systems*, *176*, 54–67.

Denœux, T., Kanjanatarakul, O., & Sriboonchitta, S. (2019). A new evidential K-nearest neighbor rule based on contextual discounting with partially supervised learning. *International Journal of Approximate Reasoning*, *113*, 287–302.

Denoeux, T., & Shenoy, P. P. (2020). An interval-valued utility theory for decision making with dempster-shafer belief functions. *International Journal of Approximate Reasoning*, *124*, 194–216.

Diaby, I., Germain, M., & Goïta, K. (2021). Evidential data fusion for characterization of pavement surface conditions during winter using a multi-sensor approach. *Sensors*, *21*(24), 8218.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., … Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the 2021 international conference on learning representations* (pp. 1–21). Vienna, Austria.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, *111*(1), 98–136.

Gao, Y., & Mosalam, K. M. (2018). Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 748–768.

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In *Proceedings of the 34th international conference on machine learning* (p. 13211330). JMLR.org.

Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., & Liu, W. (2019). Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the 2019 ieee/cvf international conference on computer vision* (pp. 603–612). Seoul , South Korea.

Huyan, J., Li, W., Tighe, S., Xu, Z., & Zhai, J. (2020). CrackU-net: A novel deep convolutional neural network for pixelwise pavement crack detection. *Structural Control and Health Monitoring*, *27*(8), e2551.

Jeong, J.-H., Jo, H., & Ditzler, G. (2020). Convolutional neural networks for pavement roughness assessment using calibration-free vehicle dynamics. *Computer-Aided Civil and Infrastructure Engineering*, *35*(11), 1209–1229.

Jiang, H., Wang, R., Gao, J., Gao, Z., & Gao, X. (2017). Evidence fusion-based framework for condition evaluation of complex electromechanical system in process industry. *Knowledge-Based Systems*, *124*, 176–187.

Lau, S. L., Chong, E. K., Yang, X., & Wang, X. (2020). Automated pavement crack segmentation using u-net-based convolutional neural network. *IEEE Access*, *8*, 114892–114899.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., … Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740–755).

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 3431–3440).

Minary, P., Pichon, F., Mercier, D., Lefevre, E., & Droit, B. (2019). Evidential joint calibration of binary SVM classifiers using logistic regression. *Soft Computing*, *23*(13), 4655–4671.

Nguyen, L. D., Lin, D., Lin, Z., & Cao, J. (2018). Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation. In *Proceedings of the 2018 ieee international symposium on circuits and systems* (pp. 1–5). Florence, Italy.

Nguyen, N. T. H., Le, T. H., Perry, S., & Nguyen, T. T. (2018). Pavement crack detection using convolutional neural network. In *Proceedings of the ninth international symposium on information and communication technology* (pp. 251–256).

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., … Rueckert, D. (2018). Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*.

Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O. R., & Jagersand, M. (2020). U2-Net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, *106*, 107404.

Qu, Z., Li, Y., & Zhou, Q. (2022). Crackt-net: a method of convolutional neural network and transformer for crack segmentation. *Journal of Electronic Imaging*, *31*(2), 023040.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–241).

Shafer, G. (1976). *A mathematical theory of evidence*. Princeton: Princeton University Press.

Smets, P. (1990). Constructing the pignistic probability function in a context of uncertainty. In M. Henrion, R. D. Schachter, L. N. Kanal, & J. F. Lemmer (Eds.), *Proceedings of the 5th uncertainty in artificial intelligence* (pp. 29–40). Amsterdam, Netherlands: North-Holland.

Strudel, R., Garcia, R., Laptev, I., & Schmid, C. (2021). Segmenter: Transformer for semantic segmentation. *arXiv preprint arXiv:2105.05633*.

Sun, X., Xie, Y., Jiang, L., Cao, Y., & Liu, B. (2022). Dma-net: Deeplab with multi-scale attention for pavement crack segmentation. *IEEE Transactions on Intelligent Transportation Systems*.

Tong, Z., Ma, T., Huyan, J., & Zhang, W. (2022). Pavementscapes: a large-scale hierarchical image dataset for asphalt pavement damage segmentation. *arXiv preprint arXiv:2208.00775*.

Tong, Z., Xu, P., & Denœux, T. (2019). ConvNet and Dempster-Shafer theory for object recognition. In *Processing of the 13th international conference on scalable uncertainty management* (pp. 368–381). Compiègne, France: Springer International Publishing.

Tong, Z., Xu, P., & Denœux, T. (2021a). An evidential classifier based on Dempster-Shafer theory and deep learning. *Neurocomputing*, *450*, 275–293.

Tong, Z., Xu, P., & Denœux, T. (2021b). Evidential fully convolutional network for semantic segmentation. *Applied Intelligence*, *51*, 6376–6399.

Tong, Z., Xu, P., & Denœux, T. (2021c). Fusion of evidential cnn classifiers for image classification. In *Proceedings of the 6th international conference on belief functions* (pp. 168–176). Shanghai, China.

Tong, Z., Yuan, D., Gao, J., & Wang, Z. (2020). Pavement defect detection with fully convolutional network and an uncertainty framework. *Computer-Aided Civil and Infrastructure Engineering*, *35*(8), 832–849.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wang, W., & Su, C. (2020). Convolutional neural network-based pavement crack segmentation using pyramid attention network. *IEEE Access*, *8*, 206548–206558.

Wang, W., & Su, C. (2022). Automatic concrete crack segmentation model based on transformer. *Automation in Construction*, *139*, 104275.

Xu, P., Davoine, F., Bordes, J.-B., Zhao, H., & Denœux, T. (2016). Multimodal information fusion for urban scene understanding. *Machine Vision and Applications*, *27*(3), 331–349.

Yager, R. R., & Liu, L. (2008). *Classic works of the Dempster-Shafer theory of belief functions* (Vol. 219). Berlin, Heidelberg: Springer.

Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X., & Ling, H. (2019). Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, *21*(4), 1525–1535.

Zhang, A., Wang, K. C., Fei, Y., Liu, Y., Chen, C., Yang, G., … Qiu, S. (2019). Automated pixel-level pavement crack detection on 3d asphalt surfaces with a recurrent neural network. *Computer-Aided Civil and Infrastructure Engineering*, *34*(3), 213–229.

Zhang, A., Wang, K. C., Li, B., Yang, E., Dai, X., Peng, Y., … Chen, C. (2017). Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network. *Computer-Aided Civil and Infrastructure Engineering*, *32*(10), 805–819.

Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., … Smola, A. (2022). Resnest: Split-attention networks. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 2736–2746).

Zhu, J., Zhong, J., Ma, T., Huang, X., Zhang, W., & Zhou, Y. (2022). Pavement distress detection using convolutional neural networks with images captured via uav. *Automation in Construction*, *133*, 103991.