Trustworthy Contextual Neural Networks for Deciphering Fracture in Metals

Dharanidharan Arumugam[1], and Ravi Kiran[2]

Abstract

A novel approach was proposed and implemented to assess the confidence of the individual class predictions made by convolutional neural networks trained to identify the type of fracture in metals. This approach involves utilizing contextual evidence in the form of contextual fracture images and contextual scores, which serve as indicators for determining the certainty of the predictions. This approach was first tested on both shallow and a deep convolutional neural network employing four publicly available image datasets: MNIST, EMNIST, FMNIST, and CIFAR10, and subsequently validated on an in-house steel fracture dataset - FRAC containing ductile and brittle fracture images. The effectiveness of the method is validated by producing contextual images and scores for the fracture image data and other image datasets to assess the confidence of selected predictions from the datasets. The CIFAR-10 dataset yielded the lowest mean contextual score of 78 for the shallow model, with over 50% of representative test instances receiving a score below 90, indicating lower confidence in the model's predictions. In contrast, the CNN model used for the fracture dataset achieved a mean contextual score of 99, with 0% of representative test instances receiving a score below 90, suggesting a high level of confidence in its predictions. This approach enhances the interpretability of trained convolutional neural networks and provides greater insight into the confidence of their outputs.

Keywords: Ductile fracture, transgranular fracture, fractographs, Explainability; XAI; and Convolutional Neural Networks (CNNs).

[1] Graduate Research Assistant, School of Sustainable Engineering and the Built Environment, Ira A. Fulton Schools of Engineering, Arizona State University, Tempe, AZ 85287, email: darumuga@asu.edu

[2] Associate Professor (corresponding author), School of Sustainable Engineering and the Built Environment, Ira A. Fulton Schools of Engineering, Arizona State University, Tempe, AZ 85287, email: ravi.kiran@asu.edu

Nomenclature of the symbols and abbreviations

Symbol/ Abbreviation	Explanation
s	Contextual score of an individual test instance
$\bar{s}$	Mean contextual score
p_α	Fraction of test image instances with contextual scores less than α value.
k	No. of nearest image instances in latent space
T1	A test image instance that belongs to class A
T2	A test image instance that belongs to class B
$\mathcal{L}$	Cross-entropy loss function
p_i	True probability of the i^{th} class
$\hat{p}_i$	Prediction probability of the i^{th} class
c	No. of label or class categories in a dataset
z	Net input before activation
α	Learning rate parameter used ADAM optimizer
β_1	Parameter that controls exponential decay rate for first order moment estimate in ADAM optimizer
β_2	Parameter that controls exponential decay rate for second order moment estimate in ADAM optimizer
ϵ	A small value that prevents non-zero division in ADAM optimizer
$\mathbf{X}$	Matrix containing input image pixel values
$\mathbf{z}$	Latent representation of $\mathbf{X}$
$\boldsymbol{p}$	Classification probabilities generated by a neural network
w	Width of an image instance
h	Height of an image instance
$\mathbf{X}$	Image dataset containing the matrices of image instances
$\mathbf{Z}$	Latent feature matrix of image dataset $\mathbf{X}$
z_q^i	The i^{th} latent vector components of a query instance
z_r^i	The i^{th} latent vector components of a neighboring instance
d_E	Euclidean distance between a neighbor instance and a query instance
NIST	National institute of standards and technology database
MNIST	An image dataset that contains handwritten digits from 0 to 9

EMINST	An image dataset that contains handwritten character digits derived from the NIST Special Database 19
FMNIST	A standard dataset of Zalando's article images
CIFAR10	A color image dataset of Canadian institute for advanced research consisting of 10 classes
FRAC	An in-house steel fracture dataset
CNN	Convolutional neural network
PINN	Physics informed neural network
XAI	Explainable artificial intelligence

1. Introduction

The increased availability of data and processing power has propelled the use of machine learning and deep neural networks for engineering mechanics applications. The applications of machine learning/ deep neural networks in solid mechanics can be broadly classified in to five classes 1) models used to synthesize microstructures with superior properties employing generative adversarial networks[1,2], 2) models that can extract or account for the surface morphology[3,4] microstructure mechanical property relationships[5-7], 3) image classification or characterization models where the type and extent of a specific damage[8,9] or material phase identification[10] and 4) metamodels that are used as surrogates to improve the predictive power and computational efficiency of numerical simulations[11,12] and 5) model calibration[13] and fracture and fatigue prediction models[14-20]. Several researchers reported very high accuracies in their studies when deep neural networks are used to solve mechanics problems. This can be attributed to the use of a very large number of hyperparameters and equally high mathematical transformations in deep neural networks. For this reason, the deep neural networks overfit the data leading to higher errors on newer datasets that were not used for training purposes. Hence, accuracy metrics alone are not enough to raise the confidence levels on data-driven models.

With the increase in the use of data-driven mechanics models, there was also an increased sense of skepticism as these models do not account for the underlying physics and it is not possible to unwind the mathematical operations in a neural network to explain how they account for the complex underlying relationships. This broader concern among engineers and material scientists is being addressed employing physics informed neural networks (PINNs)[21-26] and interpretable AI

methods[13,14,27-29]. The most popular approach in PINNs is to incorporate physics by modifying the cost function. On the other hand, interpretable AI encompasses several post-hoc methods that are used on a trained network to provide explanations to the predictions. Several researchers in mechanics acknowledged the need for improving the reliability of the deep neural networks in mechanics applications by employing PINNs, some others have embraced the Interpretable AI techniques to explain the predictions of trained neural networks. Both strategies aim to boost confidence in deep learning models used in mechanics.

In particular, convolutional neural networks (CNNs) have been increasingly used for the classification of fracture images due to their ability to achieve high predictive accuracy without performing complex feature extraction. For instance, Bastidas-Rodriguez et. al. proposed a modified deep adaptive wavelet network with adaptive lifting schemes to classify the metal fracture images into ductile, brittle and fatigue categories [30]. Their model achieved 74.7% accuracy on a real-scale dataset with a network with 174K parameters and 63.7% accuracy on SEM dataset with 19M parameters. Similarly, Alqahtani et. al. utilized CNNs to classify fatigue crack damage in polycrystalline alloys into no-risk, low-risk and high-risk categories, achieving around 90% accuracy [31]. CNNs have also been employed to distinguish fractures into cleavage, dimple and intergranular type [32]. In all these studies, the focus was solely on the predictive power and computational efficiency of the classifier models, with little attention given to the reliability of individual predictions.

This gap highlights the growing need for interpretability measures that can complement predictive models by providing insights into their decision-making processes. Deep neural network models, with interpretability measures, can gain significant trust from users by providing a clear understanding of their predictive behavior. Interpretable white-box models are particularly trusted due to their transparency but have limitations in predictive capability and application scope[33]. The level of understanding derived from DL models also relies on the users' domain expertise. Attribution-based post-hoc interpretation methods are effective in capturing how inputs influence model predictions[34,35]. When combined with sanity checks, saliency maps generated by these methods can enhance trust in black-box DL models[36]. However, even users with considerable domain expertise face challenges in selecting the appropriate attribution approach and conducting sanity checks. Moreover, interpreting the saliency maps themselves requires domain expertise, which may not be available to end-level users who utilize deployed models. Given these

considerations, we require diverse methods or measures to address trust concerns at different user levels.

The goal of this study is to propose new interpretability measures (contextual images and scores) to improve the confidence in the predictions of a convolutional neural network (CNN) that was trained to identify the fracture type in metals from images. In this study, contextual images, contextual scores, and mean contextual scores have been introduced for CNNs trained to recognize the fracture type in metals. The contextual images will serve as a qualitative tool for the end user to build confidence in CNN's prediction. On the other hand, the contextual score will provide a quantitative confidence measure for individual predictions. Furthermore, by averaging this score over a representative sample of fractographs, a mean contextual score can be derived, providing an estimate of the confidence with which the trained CNN can be applied to the entire fracture dataset. In addition to this mean score, we also consider the proportion of samples falling below a specified contextual score threshold (e.g., 90%), which offers a complementary view of model reliability by highlighting the extent of poor predictions. In the subsequent sections, the concepts of contextual images and score will be introduced, and this form of interpretability will first be demonstrated on generic datasets and will then be applied to an in-house fracture dataset for complete validation.

2. Contextual evidence

This study introduces the concept of contextual evidence as a valuable tool to indirectly assess the confidence level of individual predictions made by a trained CNN model to identify fracture type in steels. Contextual evidence is composed of two key components: contextual images and a contextual score, both defined as follows:

2.1.Contextual images:

These are training images that are closest to a specific test image in the latent space of the CNN model. The latent space represents a low-dimensional representation of input images formed by the fully connected layer just before the SoftMax classification layer (refer to Fig.2 and Fig.3). Examples of contextual images generated for various applications using different trained CNN models is provided in Section 6 and the qualitative advantage in establishing confidence in the CNN predictions is demonstrated.

2.2.Contextual score:

The contextual score is defined at both instance and total dataset levels, i.e., contextual score for an individual test instance (s), and the contextual score for the entire test dataset, which is referred to as the mean contextual score ($\bar{s}$). The individual contextual score, as described in Eq. 1, represents the percentage of k nearest image instances (in the latent space) whose true class matches the predicted class of the test image. In other words, it measures the proportion of training images belonging to the predicted class among the k closest training images used for exploration. This calculation is then multiplied by 100 to obtain a percentage value.

$$s = \frac{no\ of\ training\ images\ instances\ that\ belong\ to\ the\ predicted\ class\ of\ the\ test\ image}{no\ of\ training\ images\ used\ for\ the\ exploration\ (k)} \times 100 \quad (1)$$

The mean contextual score of a trained CNN model, described in Eq. 2, is calculated by taking the average of the individual contextual scores for a randomly selected percentage of test samples from the dataset.

$$\bar{s} = \frac{sum\ of\ the\ contextual\ scores\ of\ images\ in\ a\ random\ sample}{random\ sample\ size\ (n)} \times 100 \quad (2)$$

The proportion of poor contextual score predictions, denoted as p_α, is calculated as the percentage of samples in a randomly selected subset of the test dataset that have contextual scores below a specified threshold (α). This is expressed in Eq. (3):

$$p_\alpha = \frac{no\ of\ test\ instances\ with\ contextual\ score\ less\ than\ \alpha}{random\ sample\ size\ (n)} \times 100 \quad (2)$$

To further explain the concept of contextual score, let's refer to an example in Fig. 1, where we have a scatter plot representing the low-dimensional representation of training image instances in the latent space (the low-dimensional vectors generated from input images through a series of convolutions and pooling operations that feed into the final SoftMax layer of a CNN). The images belong to two different classes: class A shown as red-filled circles and class B as green-filled circles. We also have two test images, T1 and T2, with T1 predicted as class B (shown with a red-edged circle) and T2 predicted as class A (shown with a green-edged circle). The decision boundary is depicted as a straight blue line. To compute the contextual score for T1 and T2 in the latent space, we consider the five nearest images for examination (k=5). For T1, which is predicted as class A, three of the nearest image instances belong to class A, while two belong to class B. The contextual score is defined as the percentage of instances within the circle that belong to the predicted classes of the test image. Consequently, the contextual score for T1 is 60%. In the case

of T2, all five nearest image instances belong to class A, resulting in a contextual score of 100%. Notably, the test image (T1) near the decision boundary receives a lower contextual score compared to the test image (T2) situated further away from the decision boundary.

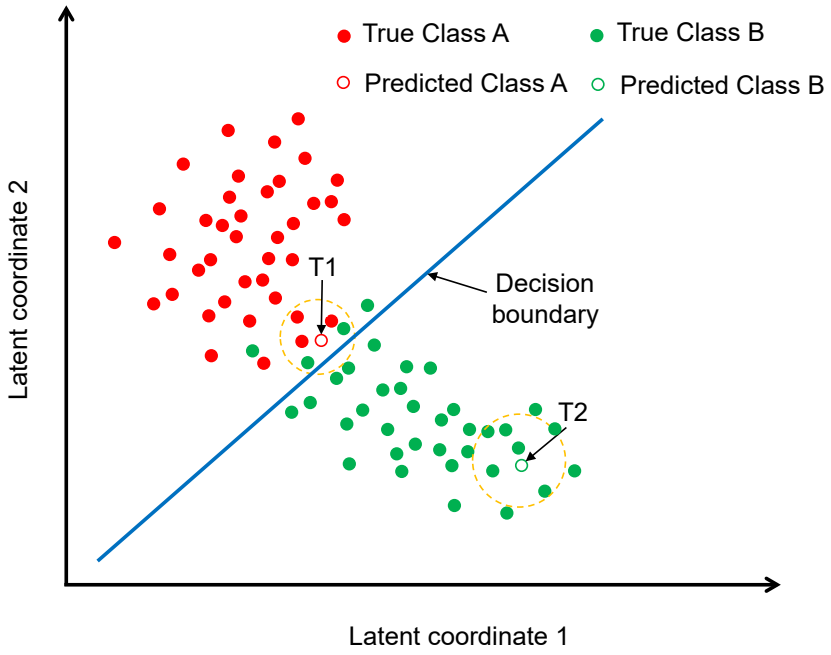

Fig. 1: The plot illustrates the distribution of training images in the latent space, categorized into two classes with a classifier's decision boundary is represented by the blue line.

3. The role of contextual evidence in building trust in CNN models

Contextual evidence, which consists of contextual images and contextual scores, can play an important role in building confidence among end-users regarding CNN model predictions. When users examine contextual images alongside their corresponding true classes, they can visually comprehend the similarities between a queried test image and the nearest training images identified by the CNN model. This qualitative understanding allows end-users, even those without subject expertise in fracture mechanics, to draw conclusions about the complexity underlying the model's predictions. Additionally, contextual images are valuable in facilitating the comprehension of unique image features that differentiate one class from another. This enhanced understanding of the model's predictions through contextual images fosters a sense of trust and reliability among end-users, as they can see firsthand the alignment between the queried test images and the training images similar to it. By inspecting visual similarities and discerning the distinctive image features,

end-users can have greater confidence in the model's ability to make accurate predictions, thus strengthening their trust in the model.

In contrast to contextual images, contextual scores provide a quantitative framework for assessing the level of confidence in predictions made by a CNN model. As depicted in Fig. 1, when input images are projected into the latent space, they form distinct clusters that correspond to their respective classes. Decision boundaries separate these clusters to facilitate classification. When a test image is located near these decision boundaries in the latent space, there is a high likelihood of misclassification. Conversely, as the test image moves farther away from the decision boundaries, the likelihood of misclassification decreases. Consequently, the class predictions of the test images positioned close to the decision boundaries should be associated with low confidence levels. The proximity of a test image in the latent space, whether nearer or farther from the decision boundaries, can be determined by inspecting the true classes of neighboring training images. This proximity influences the contextual score assigned to the test image. Hence, test image instances that are adjacent to image instances from other classes in the training latent space, typically found closer to the decision boundaries, tend to receive lower contextual scores. This phenomenon is exemplified in the previous section, where the test data instance T1, located near the decision boundary in Fig. 1, has a lower contextual score compared to the test instance T2. This contextual score measure enhances the trustworthiness of CNN model predictions by providing a transparent and quantifiable measure of confidence and empowers users to make informed decisions and develop a greater level of trust in the CNN model, knowing that predictions with higher contextual scores are more likely to be accurate.

Apart from the contextual images and scores that enhance trust in the CNN model at the individual prediction level, an additional quantitative measure known as the mean contextual score is introduced to assess the confidence in model’s predictions across multiple instances. Consequently, the mean contextual score instills trust in the model by showcasing its consistency, robustness, and suitability for effectively handling the given dataset.

4. Generic Datasets for Demonstration

In this work, four labeled image datasets i.e., MNIST, EMNIST (letters), FMNIST, and CIFAR10 were employed to train the convolutional neural networks for the initial demonstration

purposes. MNIST[37] database consists of grey images of handwritten digits ranging from 0 to 9. EMNIST[38] is an extension of the MNIST dataset. It consists of handwritten numerical digits and handwritten lowercase and uppercase English letters processed binarily from the NIST dataset. The images include both uppercase and lowercase handwritten letters. FMNIST[39] is MNIST for fashion images that belong to 10 categories namely t-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. The CIFAR10[40] dataset consists of RBG images of airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The details of the data sets used in this work are summarized in Table 1.

4.1 Metal Fracture Dataset for Demonstration

Ductile fracture, brittle (transgranular) fracture and intergranular fracture are the three types of metallic fractures. Intergranular fracture is rare and occurs in tool steels with accumulated heterogeneities at the grain boundaries. The choice of the fracture model depends on the type of metallic fracture. FRAC[41] dataset consists of brittle and ductile fracture images of ASTM A36, A572, and A992 grade steels which are popular US construction steels. The images are of greyscale and size 32×32 pixels. Ductile fracture is characterized by the presence of dimples at the microscale. These dimples are a result of micro void coalescence[42,43]. On the other hand, the brittle fracture surface is covered with river-like patterns produced as the fracture propagates across the grains. The ASTM A36, A572 and A992 steels for which the fracture data is available are popular US structural steel grades used in the construction industry. Identifying the type of fracture is crucial for choosing an appropriate fracture model and the type of fracture is determined from the fractographs. The FRAC dataset comprises 10,400 training images, with 5,200 images from each class (brittle and ductile fracture), and 2,000 testing images, with 1,000 images from each class.

Table 1: Details of the datasets used for evaluating contextual scores

Attribute	MNIST	EMNIST	FMNIST	CIFAR10	FRAC
Size of the image (pixels)	28 × 28	28 × 28	28 × 28	32 × 32	32 × 32
Channels	1 (grey)	1 (grey)	1 (grey)	3 (RGB)	1 (grey)
Number of classes	10	26	10	10	2
Training dataset size	60,000	1,24,800	50,000	50,000	10,400
Test dataset size	10,000	4800	10,000	10,000	2,000
Dataset nature	balanced	balanced	balanced	balanced	balanced

Dataset details	digits 0-9	alphabets: a-z and A-Z	fashion images	animals, vehicles	fracture images
Reference	37	38	39	40	41

5. Construction of Contextual Evidence for Classifier Predictions

The generation of contextual evidence for an image instance (a query instance) and the determination of a representative contextual score for a network model were carried out in four stages: 1. training and configuration of a model, 2. selection of a latent layer, 2. building latent coordinates, 3. determination of nearest neighbors and 4. generation of contextual evidence. These stages are further elaborated as follows.

Stage 1 – Training and configuration of a model: The generation of contextual evidence for a neural network model starts after the training and configuration of a neural network model. Configuration of a neural network involves arriving at a network architecture for a desired prediction performance for both the learned and unknown data. Configuration includes finding the appropriate amount and proportion of different network layers (convolutional layers, pooling layers, dropout layers, and so on) and their arrangements, setting the number of kernel filters in each convolution layer, choosing the number of neurons in each fully connected layer, fixing sizes of kernel filters and pooling windows and selecting suitable activation functions. Training involves tuning the learnable network parameters, which are normally kernel and dense layer weights and layer biases so that the model losses are minimized at each network model configuration. Training and configuration are interactively performed to obtain the best-performing model. More about the training and configuration of a neural network, particularly a convolution neural network, can be found in the literature[44,45].

In this work, convolutional neural networks (CNNs) were designed for the classification of the employed datasets. Two networks were trained on each of the demonstration datasets and the FRAC dataset, namely "deep" and "shallow". The "deep" network was designed with more convolutional and fully connected layers than the "shallow" network, in order to analyze the effect of the deeper architecture on the mean contextual score. The shallow and the deep models trained on the FRAC dataset are given in Fig. 2 and Fig. 3, respectively.

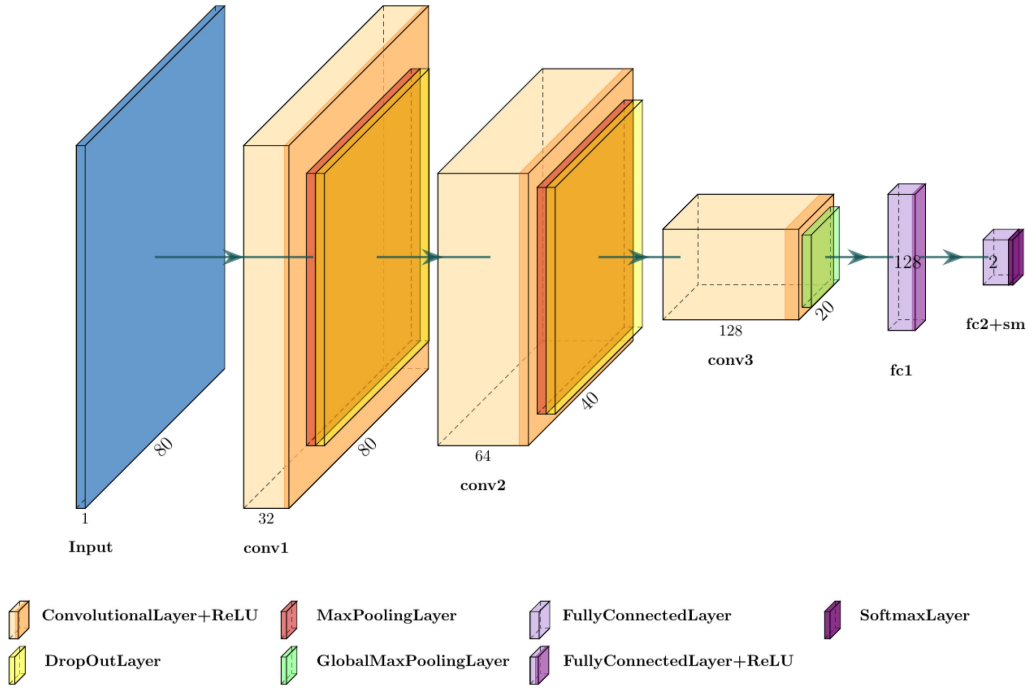

Fig. 2: Architecture of the shallow convolutional neural network model trained on the FRAC dataset.

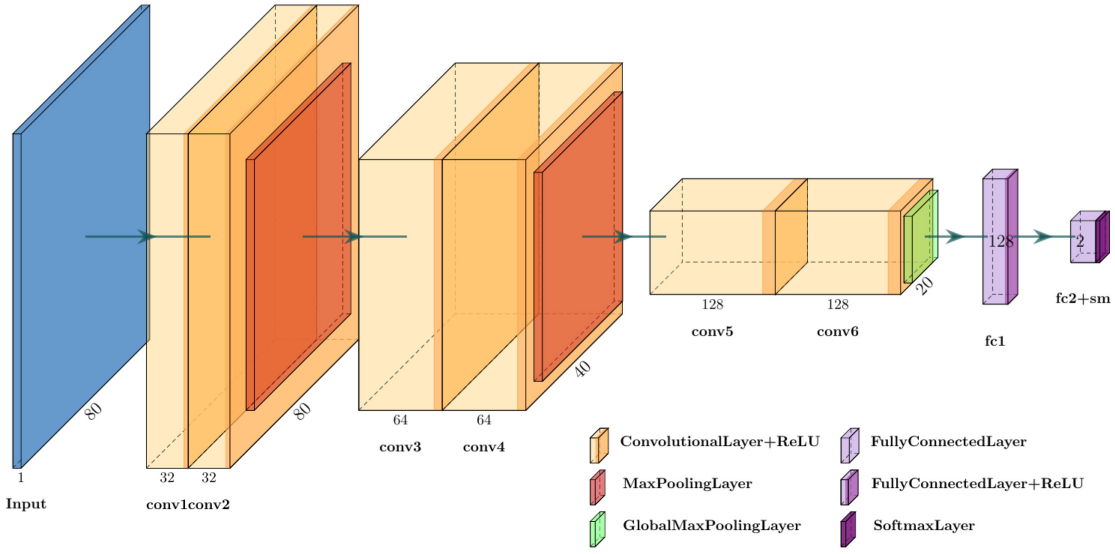

Fig. 3: Architecture of the deep convolutional neural network model trained on the FRAC dataset.

All the CNNs employed consist of alternating layers of convolutional layers and maxpooling layers. In all cases, the head of the networks comprises a global maxpooling layer followed by a fully connected layer and a SoftMax output layer. ReLU activation functions were used for the convolutional layers. Convolutional layers use a kernel of size 3×3 with a stride of 1. Maxpooling layers use a pooling size of 3×3 with a stride of 2 which results in a halving of feature

maps size along both dimensions. Shallow CNN architecture involved dropout layers for the regularization of the network. Deep CNN architectures did not include any dropout layers, and two convolutional layers were alternated with maxpooling layers. For comparison, a shallow and a deep network architecture trained on the FRAC dataset is shown in Table 2 and Table 3. The number of trainable parameters in deep CNN models is two to three times that in shallow CNN models. The number of trainable network parameters of the CNN models used in this work is summarized in Table 4.

Table 2: Network architecture of shallow network, used to classify FRAC data.

Layer	Layer size	Activation function	Data size
Input	-	-	80×80×1
Convolution 1	32×3×3	ReLU	80×80×32
Maxpooling 1	2×2	-	40×40×32
Dropout (0.25)			
Convolution 2	64×3×3	ReLU	40×40×64
Maxpooling 2	2×2	-	20×20×64
Dropout (0.25)			
Convolution 3	128×3×3	ReLU	20×20×128
Global Maxpooling	-	-	128
Fully connected	2×1	ReLU	2×1
SoftMax	2×1	SoftMax	2×1

Table 3: Network architecture of deep network used to classify FRAC data.

Layer	Layer size	Activation function	Data size
Input	-	-	80×80×1
Convolution 1	32×3×3	ReLU	80×80×32
Convolution 2	32×3×3	ReLU	80×80×32
Maxpooling 1	2×2	-	40×40×32
Convolution 3	64×3×3	ReLU	40×40×64
Convolution 4	64×3×3	ReLU	40×40×64
Maxpooling 2	2×2	-	20×20×64
Convolution 5	128×3×3	ReLU	20×20×128
Convolution 6	128×3×3	ReLU	20×20×128
Global Maxpooling	-	-	128

Fully connected	2×1	ReLU	2×1
SoftMax	2×1	SoftMax	2×1

Table 4: Network sizes of CNN models employed in the study.

Model training data	Number of network parameters	
	Deep network	Shallow network
MNIST	72,442	23,946
EMNIST	289,786	96,026
FMNIST	1,174,250	390,410
CIFAR10	1,174,826	390,986
FRAC	286,690	55,874

Efficient training and configuration of a network model also require defining an appropriate model loss function. Mean squared error, mean absolute error, Kullback-Leibler divergence, and cross-entropy are commonly used loss functions for neural networks[46]. In this work, CNN models were employed for the classification of image datasets, hence, the categorical cross-entropy function was used as the loss function. The expression used to compute the loss function is given as

$$\mathcal{L}(p, \hat{p}) = -\sum_{i=1}^{c} p_i \log(\hat{p}_i) \tag{3}$$

where, $\hat{p}_i$ is the probability of prediction for the i^{th} class, p_i is the actual probability of the i^{th} class, and c is the number of classes. The probability of prediction for a class is estimated from the net input z using the Softmax function as given below,

$$\hat{p}_i = e^{z_i} / \sum_{i=1}^{c} e^{z_i} \tag{4}$$

The losses were minimized by backpropagating the network gradients using the ADAM algorithm[47]. The hyperparameters used for the ADAM optimization are shown in Table 5. The initial learning rate (α) determines the magnitude of each parameter update during training. The exponential decay rates β_1 and β_2 are applied to the first and second moment estimates of the gradients. The first moment corresponds to the exponentially decaying average of past gradients (akin to momentum), while the second moment tracks the uncentered variance, capturing the scale of the gradients. Together, these parameters enable adaptive learning rates for each weight in the

network, improving stability and convergence speed, particularly in the presence of sparse gradients or noisy updates. ϵ – the small value is used to prevent nonzero division error. The training and testing accuracy achieved for the models are shown in Fig. 4.

Table 5: Hyperparameters used in the ADAM optimization.

Parameter	β_1	β_2	ϵ	α
Value	0.9	0.999	10^{-8}	0.001

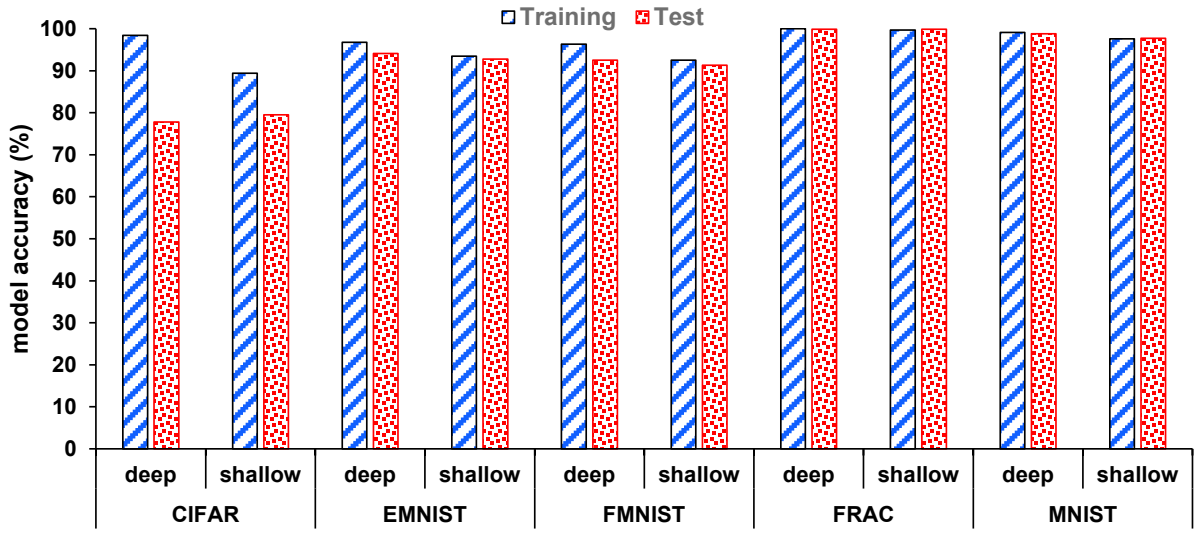

Fig. 4: Comparison of training accuracies and test accuracies for the CNN models employed in the study.

Stage 2 – Selection of a latent layer: After training and configuring a classification model, a neural network layer, called a latent layer, that produces a lower-dimensional representation of the image data was selected. A fully connected layer (one-dimensional layer) is preferred for the latent representation of the image data due to the reduced sparsity in the data representation and the lower computational effort required to determine neighbor instances. In our analysis, the penultimate layer, which is a fully connected layer, was used as the latent layer for all the CNN models. The penultimate latent layer of each CNN model was followed by the SoftMax layer which outputs the classification probabilities of the input images. An intermediate model with the latent layer as the output layer was constructed from the trained CNN model. This new intermediate model transforms and maps the higher-dimensional image inputs sparsely spread on the input space to lower-dimensional numerical vector outputs densely structured in a latent space as illustrated in Fig. 5.

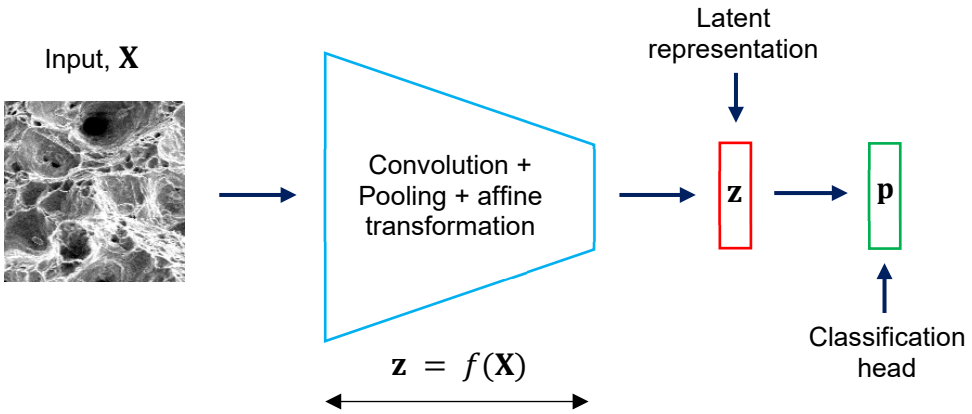

*Fig. 5: Latent transformation of inputs: The high-dimensional image input, **X**, is transformed into a low-dimensional, information-rich, latent feature vector, **z**. Classification probabilities (**p**) for each class of data are generated by a classification head operating on these latent features.*

This lower dimensional latent space with a manifold structure of input image data was used to search for instances nearer to a query instance. The number of neurons in the latent layer formed the dimension of the latent space used as the search space. The latent dimension in our analysis was equal to the number of class labels (c) of the image dataset. The dimensions of the latent space differed for the individual datasets used in the analysis. For example, the latent dimension of the FRAC dataset was two representing the class labels: *brittle* and *ductile*. The summary of the latent dimensions of the different datasets used in the current analysis is given in Table 6.

Table 6: Latent dimensions of the datasets used in the analysis.

Model training data	Latent space dimension
MNIST	10
EMNIST	26
FMNIST	10
CIFAR10	10
FRAC	2

Stage 3 – Building latent coordinates: The extracted dimensional reduction model takes an image input, $\mathbf{X}_i (\in \mathbb{R}^{w \times h})$, and outputs a latent feature vector, $\mathbf{z}_i = [z_i^1, z_i^2, \ldots, z_i^c]$. Where, w and h are the pixel width and the pixel height of an input image, respectively, and c is the number of output components in a latent feature vector or number of classes. The latent feature vector, $\mathbf{z}_i$, is considered to represent the positional coordinates of the input image, X_i, in a lower dimensional

(latent) space with a zero vector as the origin. Accordingly, the latent feature matrix, $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n]^T \in \mathbb{R}^{n \times c}$, for the input dataset, $\boldsymbol{\mathcal{X}} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n] \in \mathbb{R}^{n \times w \times h}$, contains the positional coordinates of all the image instances in an input dataset, in the latent space. In the current work, latent feature matrices were generated for all ten CNN models for both the training and the testing datasets. The generated latent feature matrices were stored in '.csv' files during the first computation of a contextual score or a generation of contextual evidence to reduce the computational effort and time spent on determining the latent coordinates repeatedly. The subsequent computation of contextual scores for the same model involved loading the already stored latent coordinates databases.

Stage 4 – Generation of contextual evidence: The latent search space built for each model was then used to find the nearest instances to the query instances. Our analysis employed the 'Euclidean' distance metric to find the nearest instances. Other distance metrics such as Manhattan, Minkowski, and cosine distances are also commonly employed to evaluate similarity between data points—particularly in tasks like clustering or retrieval. However, these metrics are generally not suitable for comparing classifier predictions. Take cosine distance, for example: while it is widely used in text analysis and information retrieval due to its sensitivity to vector orientation rather than magnitude, it performs poorly when applied to classification logits. This mismatch arises because classifiers – especially neural networks – produce output vectors whose magnitudes carry critical information. As illustrated in Fig. 10, the logit vectors tend to disperse perpendicularly to the decision boundaries, meaning that the distance from the origin (i.e., magnitude) reflects confidence or separability, not just direction. Using cosine distance in such contexts effectively discards this magnitude information, leading to misleading assessments of similarity. Similarly, Manhattan distance, which sums absolute differences across dimensions, is typically more appropriate in high-dimensional or sparse, discrete feature spaces – conditions that don't align well with the smooth, dense, continuous-valued outputs of modern classifiers. Therefore, Euclidean distance remains a more aligned choice in this context, as it naturally incorporates both magnitude and direction, better capturing the underlying geometry of the classifier's output space.

The Euclidean distance between a neighbor instance and a query instance is the L_2 norm of the relative coordinates between the neighbor instance and the query instance and it was computed as follows,

$$d_E = \sqrt{\left(z_q^1 - z_r^1\right)^2 + +\left(z_q^2 - z_r^2\right)^2 + \cdots + \left(z_q^n - z_r^n\right)^2} \quad (5)$$

z_q^i and z_r^i are the i^{th} latent vector components of a query instance and a neighboring instance, respectively. Subsequently, k nearest image instances were identified by ranking the latent vectors in an ascending fashion based on the relative distances. In our study, k was fixed to 100 to compute the contextual score. However, only the top 10 nearest image instances were used as contextual images. The parameter $k = 100$ was chosen to ensure a sufficiently large neighborhood for stable and reliable contextual score computation. In contrast, the use of the top 10 nearest image instances for display is not a tunable parameter related to score computation. Instead, it serves solely as a visual aid to help users intuitively assess the plausibility of the prediction based on the similarity of neighboring instances. While more images could be shown, doing so often introduces visual clutter and reduces interpretability, defeating the purpose of this qualitative insight. The true and predicted class labels of the nearest image instances were then used to compute the contextual scores as described in Section 2. The contextual images along with the contextual score constitute the contextual evidence of the queried image. Finally, the mean contextual score of a model ($\bar{s}$) and proportion of poor contextual score predictions (p_α) was determined by randomly selecting 10 percent of test image instances of the respective dataset and averaging the individual contextual scores of the selected test image instances.

6. Generic Datasets: Demonstration of Contextual Images and Score

In order to demonstrate the utility of contextual images and contextual scores in enhancing confidence in CNN model predictions, one case example is randomly chosen from each of the trained CNN models utilized in the study. These case examples, based on 'shallow' and ‘deep’ convolutional neural network models, are presented in Fig. 6 and Fig. 7, respectively. For each set of images, the top row represents the image we want to gather contextual evidence for (known as the query image), while the subsequent two rows consist of contextual images that are closest to the query image. In this illustration, 10 contextual images are generated for each example case. This number can be chosen based on the user’s preference. The captions below show the class labels of the respective images. The captions below the images indicate their respective class labels. The letter 't' preceding the class labels signifies the true class, while 'p' indicates the predicted class. The figure also includes contextual scores for the queried image examples.

The CNN trained on CIFAR10 dataset yielded an accuracy of 77.8% on the shallow network which implies that this is a relatively less accurate model. The contextual images for a specific image in the CIFAR10 dataset are provided in Fig. 6a. As depicted in Fig. 6a, the queried test image belongs to the *cat* class. Firstly, this was classified as a *frog* image by CNN which is incorrect. In addition, the contextual images associated with this test image consist of a mix of images from the *cat* class as well as other classes like *dog*, *frog*, and *bird* yielding a relatively low contextual score of 22%. The contextual images are mixed, and the contextual score is low which makes this individual prediction less reliable.

The second example depicts the contextual evidence of a queried test image from the EMNIST test dataset. In this case, the test image belongs to the class *letter-Q* and was correctly classified as *letter-Q*. Furthermore, unlike the CNN trained on the CIFAR10 dataset, the CNN trained on the EMNIST dataset enjoys an accuracy of 92.8%. Contextual evidence was generated to test the reliability of this individual prediction. A significant majority of the contextual images shown in Fig. 6b, however, belong to the *letter-A* class. This inference is further supported by the low contextual score of 21% indicating a low confidence in the prediction. The lower contextual score can be attributed to the fact that the queried image is surrounded by a higher proportion of images from the *letter-A* and *letter-G* classes. This occurrence is expected since the *letter-Q* in the queried image bears a close visual resemblance to the letters A and G. When contextual scores are low the predictions should be viewed with skepticism irrespective of the accuracy of the model.

Similarly, in the FMNIST dataset case, the shallow CNN enjoys a higher accuracy of 91.3%. The queried image of the *coat* is predicted correctly as *coat* by the trained network. However, a considerable proportion of the contextual images are *shirt* images resulting in a contextual score of 45% making this particular prediction less reliable although correct. On the other hand, the shallow CNN trained on the MNIST dataset has an accuracy of 97.7% which is quite impressive. *digit-0* has been queried and was predicted as *digit-0* by the network. Furthermore, most of the contextual images belonged to *digit-0* yielding a contextual score of 83% making this a reliable prediction.

In the case of the deep network model, the same queried image (see Fig. 7a) is incorrectly predicted as a dog with a high contextual score of 94, even though it actually belongs to the cat class. However, the contextual images clearly show a lack of similarity, revealing the model's poorly constructed decision boundary. This highlights a key limitation of the proposed approach:

when the model predicts the wrong class, but the neighboring (contextual) images belong to that predicted class, it still receives a high contextual score. This happens because the model has low bias, making it difficult to distinguish between the predicted and true classes. Nonetheless, the contextual images still provide valuable insight by exposing such misclassifications. In the EMNIST example (see Fig. 7b), the contextual score improved to 70%, and many contextual images belong to the letter-Q class, indicating an improved model fit. A similar trend is observed in the MNIST dataset (see Fig. 7c), where the contextual score increased to 100%, and all contextual images are unambiguously similar to the queried image. An exception is seen in the FMNIST example (see Fig. 7d), where the queried coat image received a lower contextual score, and the prediction changed to shirt.

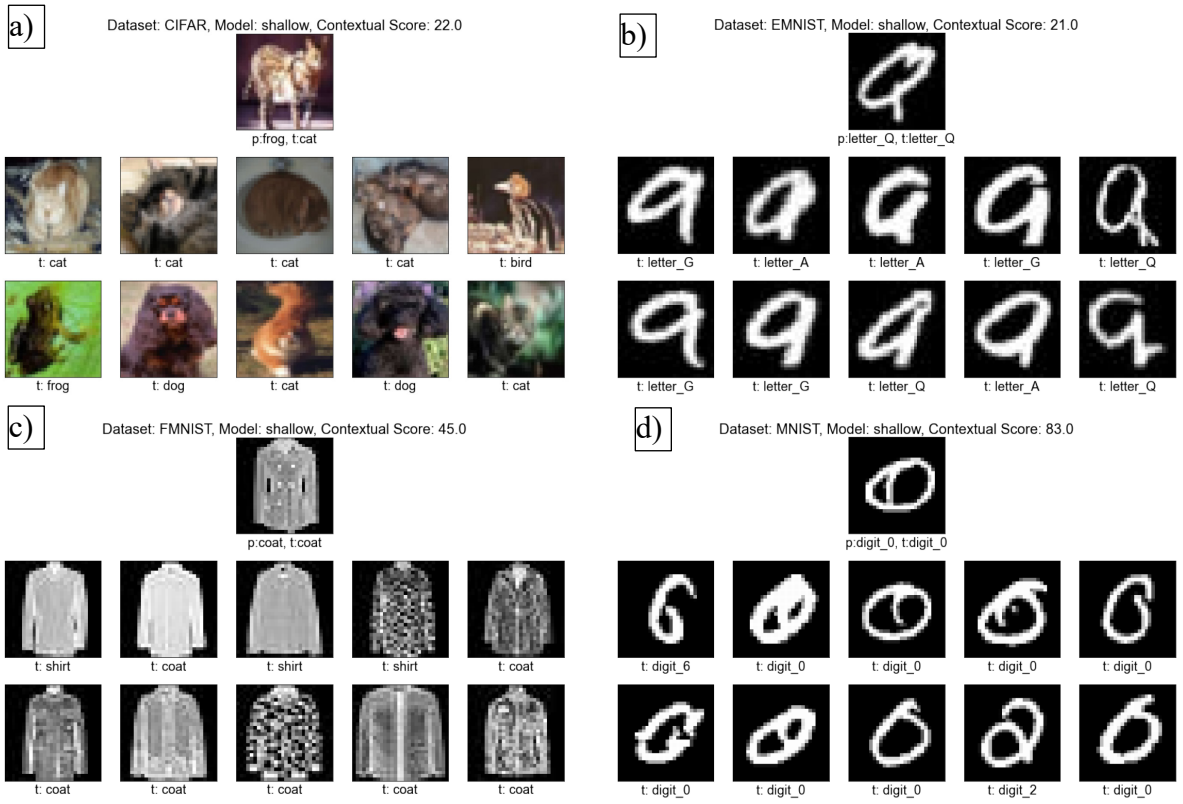

Fig. 6: Contextual images and scores generated for an image example from a) CIFAR10, b) EMNIST, c) FMNIST, and d) MNIST dataset trained on 'shallow' convolutional neural network models. The captions under the images are the class labels and the letters: 'p' or 't' before the labels denote whether the class label is predicted or true.

From this discussion it is clear that accuracy metrics cannot boost confidence on individual predictions and contextual evidence serves as a problem agnostic approach to improve the reliability of neural network predictions.

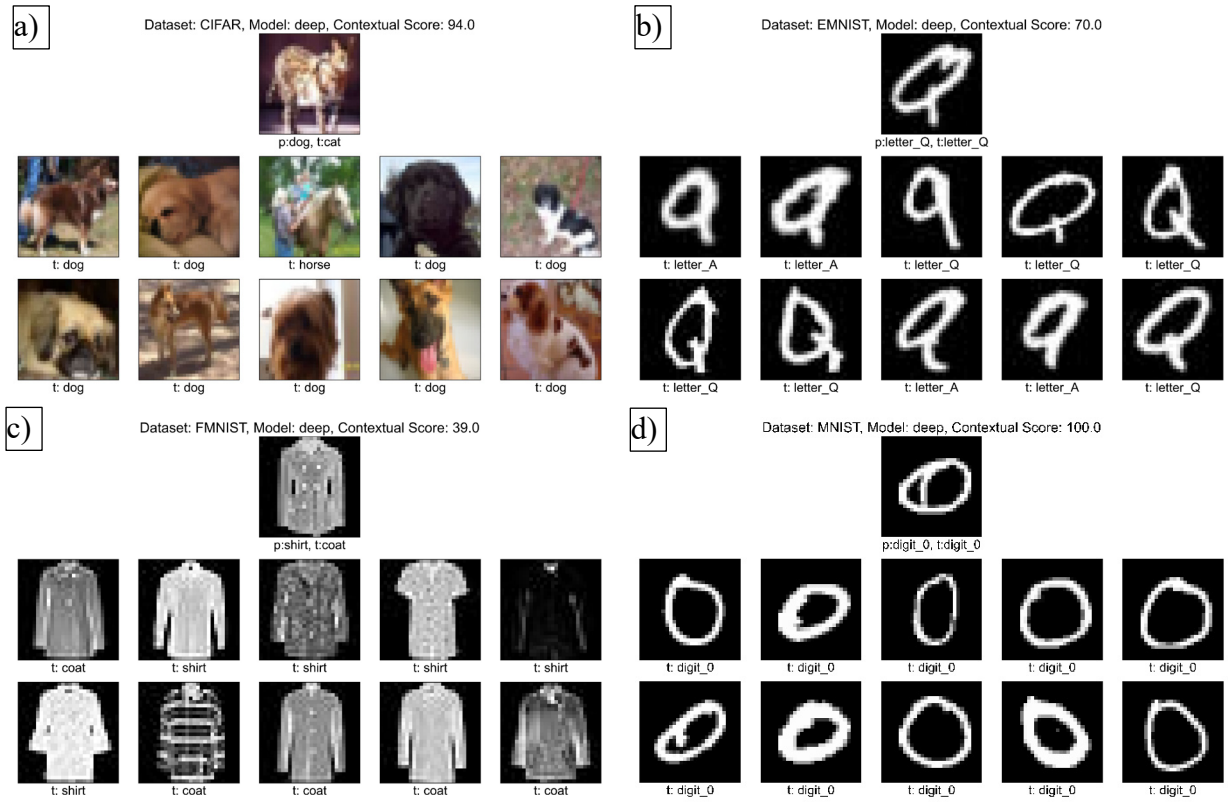

Fig. 7: Contextual images and scores generated for an image example from a) CIFAR10, b) EMNIST, c) FMNIST, and d) MNIST dataset trained on deep convolutional neural network models. The captions under the images are the class labels and the letters 'p' or 't' before the labels denote whether the class label is predicted or true.

6.1 Metal Fracture Dataset: Demonstration of Contextual Images and Score

The trained CNN model accuracy on the training dataset was 99.8% and the accuracy on the testing dataset was 99.9%. However, as mentioned previously, accuracy is an overall metric does not provide confidence on individual predictions, whereas contextual images and contextual score can improve the confidence of an individual prediction. For the fracture classification model, one ductile and one brittle fracture image were queried from both the shallow and deep models, as shown in Fig. 8 and Fig. 9, respectively. In all cases, the trained CNN correctly predicted the true class. Additionally, for each queried image, 10 contextual images were generated. The predicted classes of these contextual images largely matched the true class, yielding a contextual score close to 100% for ductile fractures. For brittle fractures, the shallow model resulted in a 94% contextual score, whereas the deep model achieved a perfect score of 100%. In both cases, the deep model achieved a higher contextual score than the shallow model, indicating a better overall fit. Moreover, the contextual images retrieved by the deep model appeared more visually similar to

the queried image compared to those from the shallow model. With this, the trained network predictions are accurate, and the contextual score and images add confidence to these predictions.

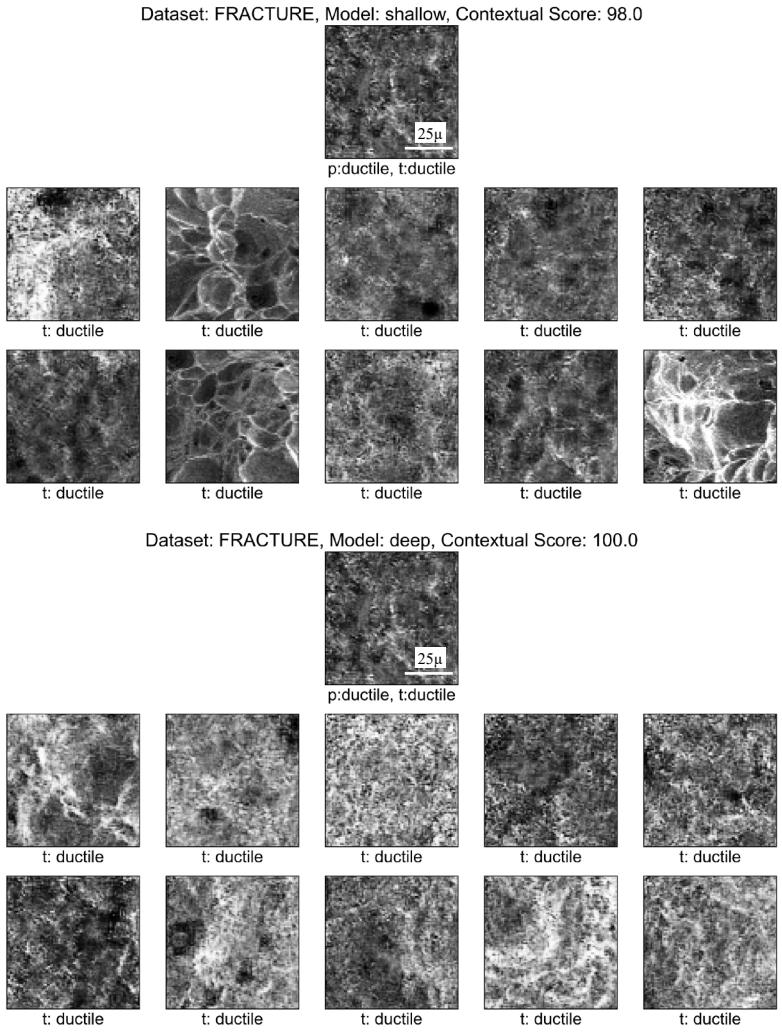

Fig. 8: Ductile fracture queried images predicted as ductile fracture and with a majority of contextual images belonging to the ductile fracture making the contextual score close to 100%.

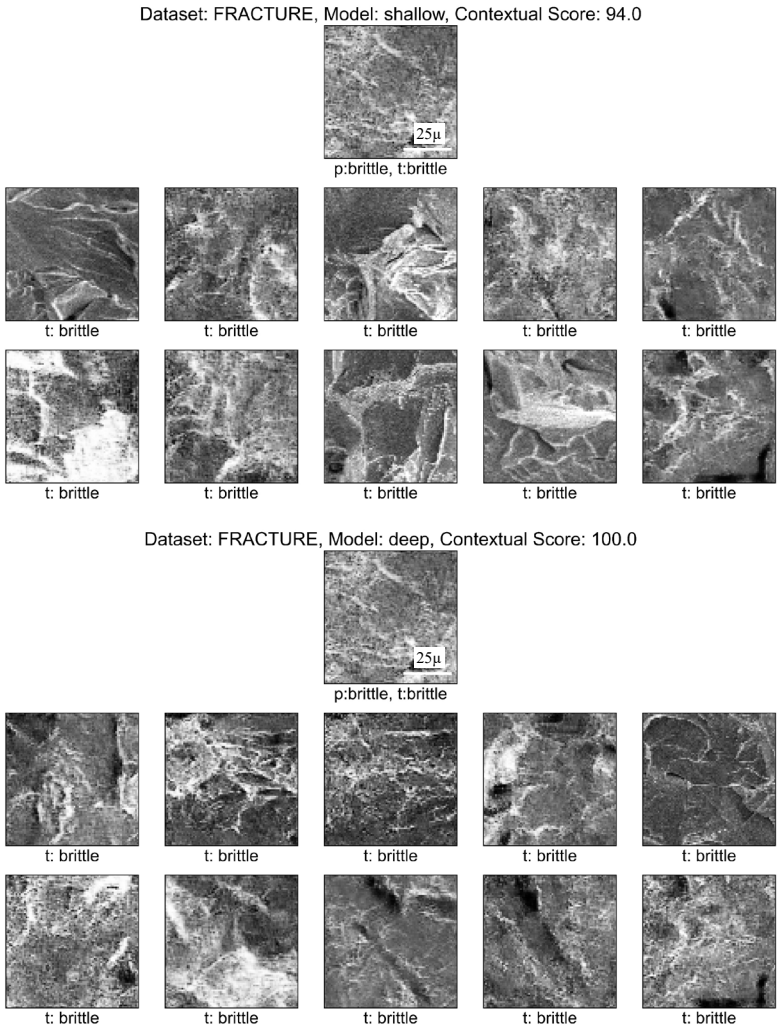

Fig. 9: Brittle fracture queried images predicted as brittle fracture and with a majority of contextual images belonging to the brittle fracture resulting in high contextual scores (94-98%).

To further explore this, we can examine the spread of the training data in the latent space of the FRAC dataset, as depicted in Fig. 10. In this dataset, which consists of only two classes, each data instance in the latent space is defined by two latent coordinates ($z_1$and z_2). The red-colored data instances in the figure represent the *brittle* class, while the blue-colored instances represent the *ductile* class. The green line in the image represents the decision surface of the model in the latent space. From the plot, we can observe that the model fits the training data well. Some *brittle* fracture data instances fall below the decision surface line, and some *ductile* fracture data instances fall above it, resulting in misclassifications. However, with a training model accuracy of 99.8% and a testing model accuracy of 99.9%, only a small percentage of the training or testing data instances are misclassified. It's important to note that any testing data instance closer to the

decision boundary has a higher likelihood of being misclassified, leading to less confidence in the prediction. However, in the current case, the queried image, represented by a yellow-filled circle in Fig. 10, falls near the center of its class group, and all the k-nearest neighbor image instances belong to the predicted class of the queried image. As a result, it achieves a contextual score of 100%, indicating a high level of confidence in the prediction. The queried ductile and brittle instances for the shallow model from Fig. 8 and Fig. 9, respectively, are represented by blue and pink circles in Fig. 10. Their proximity to decision boundaries and overlap with other class instances, as seen in Fig. 10, help explain their slightly lower contextual scores reported in Fig. 8 and Fig. 9.

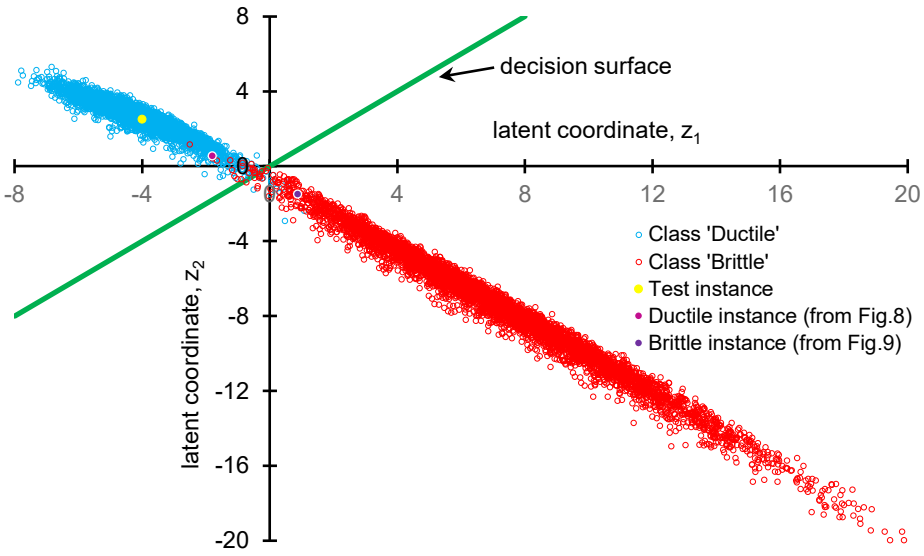

Fig. 10: Training latent space of the shallow CNN model trained on in-house FRAC dataset. The test data instance, the yellow-filled circle, in the plot, occupies the center of its class group (ductile) and has a contextual score of 100% which indicates high confidence in the prediction. The ductile and brittle instances queried in Fig. 8 and Fig. 9 are shown in pink and purple, respectively.

6.2: Sensitivity of contextual score to number of nearest neighbors (k)

To examine how sensitive the contextual scores are to the number of nearest neighbors (k), we analyzed the queried images from different datasets using the shallow model, as shown in Figs. 6, 8, and 9. In addition to the original k value, we tested two other values: $k = 200$ and $k = 50$. Contextual scores were recalculated using these alternative k values, and the results were compared in Fig. 11. As observed in Fig. 11, the contextual scores do not exhibit significant variation across different k values. This suggests a level of robustness, but it is still difficult to

generalize whether the scores will increase or decrease with a change in k. The effect appears to depend heavily on the specific image being analyzed and how its class relates to the surrounding neighboring classes in the feature space. To maintain a meaningful local context, we recommend choosing k such that it remains below 1% of the total training set size. Setting k too high may dilute the relevance of the local neighborhood, while setting it too low (e.g., k = 10) may make the scores overly sensitive to noise or outliers in the dataset.

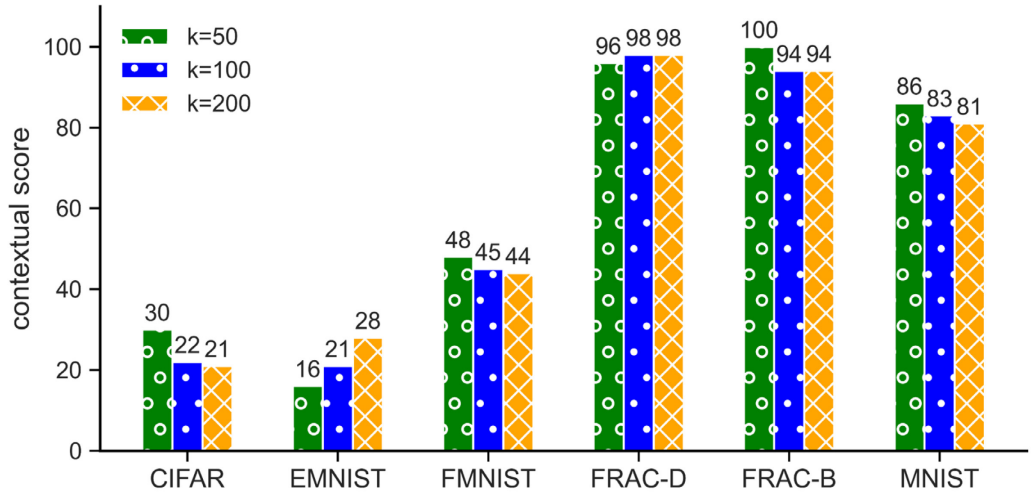

Fig. 11: Influence of the number of nearest neighboring instances (k) on the contextual scores of queried image examples used in Figs. 6–9 for the shallow models. Here, Frac-D and Frac-B indicate ductile and brittle instance used from the fracture dataset, respectively.

6.3: Contextual Scores for the Metal Fractographs

Heatmaps are generated individually for the *ductile* fracture and *brittle* fracture predictions to examine the distribution of contextual scores across the latent coordinate range of the FRAC dataset (refer to Fig. 10), as shown in Fig. 12. These heatmaps reveal the level of confidence associated with the class prediction of an image instance using its latent coordinates. The heatmap color range represents contextual scores from 0 to 100, with a smooth transition from blue (0) to red (100). The variation of the contextual scores is observed as color bands oriented approximately along the 45° line parallel to the decision surface. As depicted in Fig. 12, the variation is highly pronounced near a narrow-banded region close to the decision surface, and it remains constant for most of the upper right and bottom left regions of a heatmap. Fig. 12.a shows the heatmap of contextual scores generated for the prediction of the *ductile* class. A high degree of confidence in

predicting an image instance in the *ductile* class is indicated by the major red region on the upper left (contextual score of 100). As we move toward the decision surface, the confidence level is progressively lowered, and it becomes even lower when we move below the decision surface. Thus, the major blue region on the bottom right (contextual score of 0) indicates the lowest degree of confidence if an image instance is predicted as *ductile*, suggesting that the prediction of any image instance falling inside the blue region is more likely a misclassification. This explanation extends to Fig. 12.b, which depicts the contextual heatmap for image instances predicted as the *brittle* class. The important distinction here in the *brittle* heatmap is that it is a diagonally flipped version of the *ductile* heatmap shown in Fig. 12.a. This is because the contextual score in this case is the proportion of *brittle* fracture training data instances that are closer to the query instance and these *brittle* fracture instances are predominantly distributed below the decision surface.

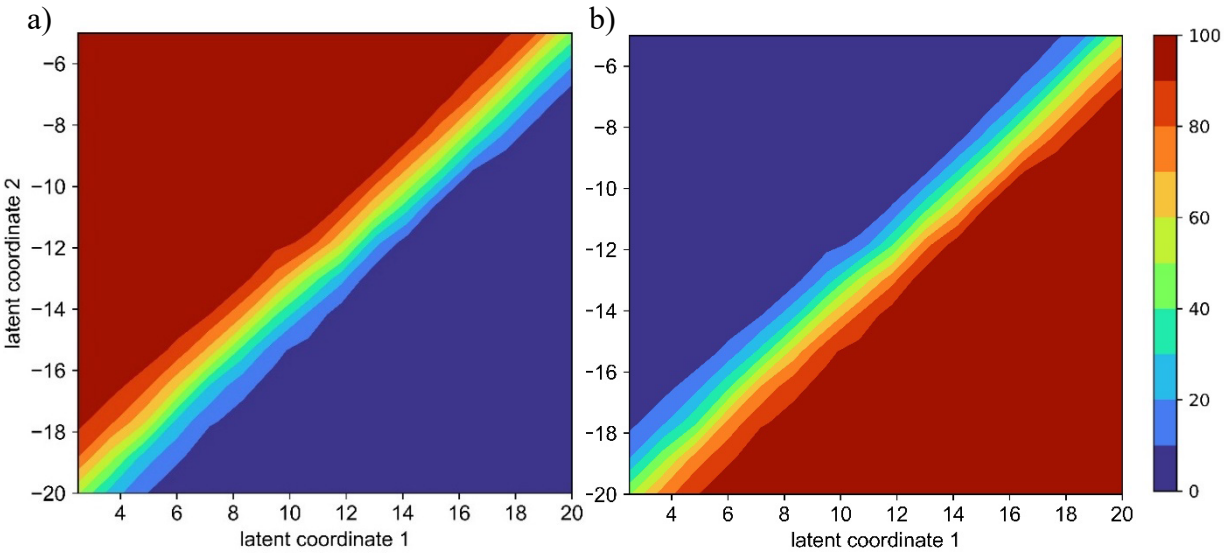

Fig. 12: Heat map of contextual scores for the testing instances that belong to a) ductile fracture class and b) brittle fracture class in the latent space of the CNN model trained on the FRAC dataset. Red indicates a contextual score of 100 and blue color indicates a contextual score of 0.

6.4 Mean Contextual Scores

To assess prediction certainty across the entire test datasets, average contextual scores – both mean and median – were calculated for all model cases. A representative sample comprising 10% of the test images was selected to obtain an overall measure. Fig. 13 shows the distribution of contextual scores for these representative samples from the shallow models trained on the various datasets used in this study. Deep models exhibit similar distributions and are therefore not shown.

Since all models achieved good fits across the datasets, the distributions are generally skewed toward high contextual scores, with longer tails observed for CIFAR, EMNIST, and FMNIST. For FRAC (fracture) and MNIST, where the model fits were excellent, the distributions are tightly concentrated between 95–100%, with no observable tails.

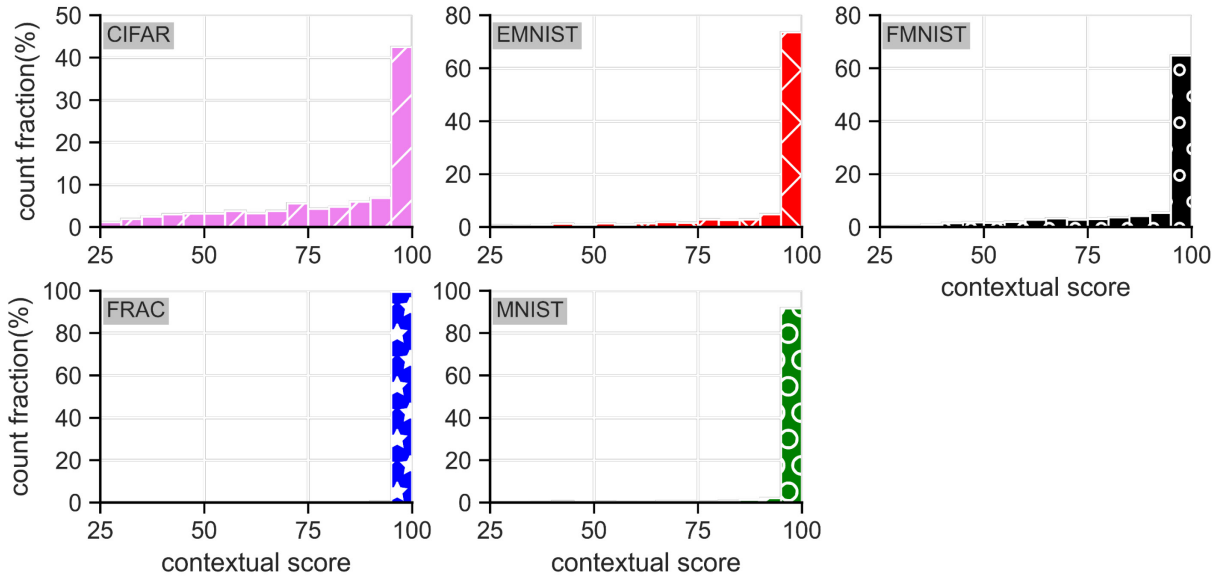

Fig. 13: Distribution of contextual scores for representative samples drawn from the test datasets of shallow models trained on the employed datasets.

Although the median is generally a better measure of central tendency for skewed distributions, it fails to capture the influence of low contextual scores present in the model's outputs. This is particularly relevant here, as a well-fitting model will naturally produce contextual scores clustered near 100, which inflates the median and masks underperforming cases. This behavior is also evident in Fig. 14, where the median scores across all datasets appear high, yet do not reflect the presence of poor predictions. Therefore, the mean is preferred in this context, as it accounts for the full distribution of scores and provides a more comprehensive view of the model's overall performance on the test dataset. The results displayed in Fig. 14 also depict the mean contextual scores of all the CNN models employed in this study. The figure also compares the mean contextual scores of shallow and deep models. For EMNIST and FMNIST datasets, the contextual scores ranged from 89 to 93. A higher mean contextual score indicates greater confidence in the multiple predictions made within the dataset. The CIFAR10 dataset exhibited relatively lower mean contextual scores compared to the other datasets, with scores of 79 and 86 for shallow and deep models, respectively. The deep models consistently outperformed the shallow models in

terms of mean contextual scores across all cases, though the differences were generally not that significant. Notably, a large difference was observed only in the case of the CIFAR10 deep model. The observed difference in performance between the shallow and deep models, particularly in the case of the CIFAR10 dataset, can be attributed to the shallow model's limited capability to effectively separate images into distinct classes within the latent space compared to the deep model. On the other hand, the CNNs trained with FRAC and MNIST datasets, both shallow and deep models achieved remarkably high contextual scores ranging from 98 to 100. A mean contextual score ranging from 95 to 100 indicates a high level of confidence in individual predictions made within the dataset. Consequently, this instills greater confidence and trust in any predictions made by the CNNs.

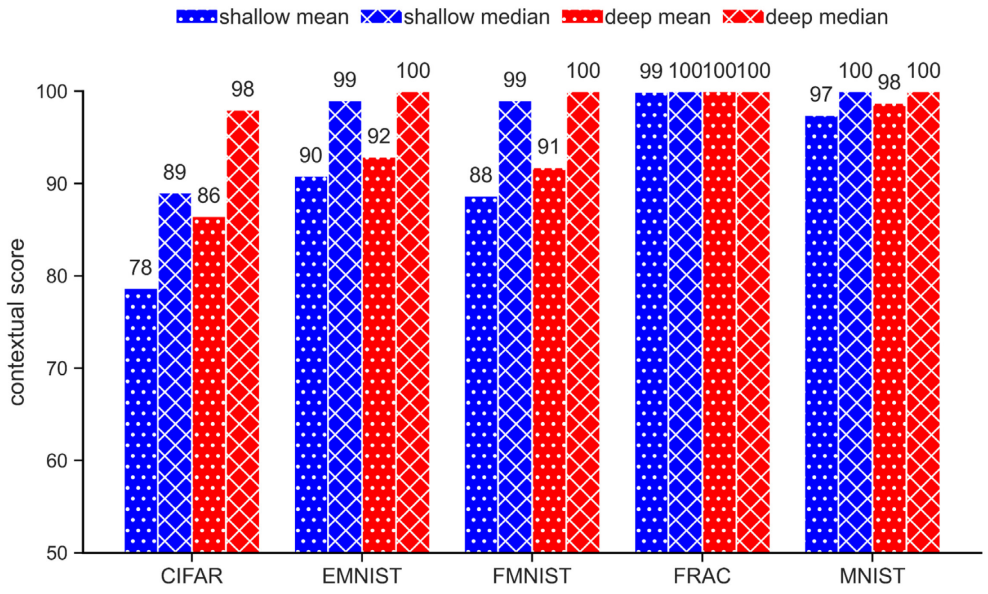

Fig. 14: Comparison of mean and median contextual scores of the deep and shallow models for various datasets employed in this study.

While the mean score offers a reliable indicator for comparing overall model performance, an additional metric is considered: the proportion of samples with contextual scores below 90% in a representative dataset. This metric adds interpretability by quantifying the extent of poor predictions. The 90% threshold, while somewhat subjective, is chosen based on domain-specific expectations of acceptable performance. Analysts may adjust this cutoff depending on the sensitivity of the application or the tolerance for prediction errors in real-world deployment.

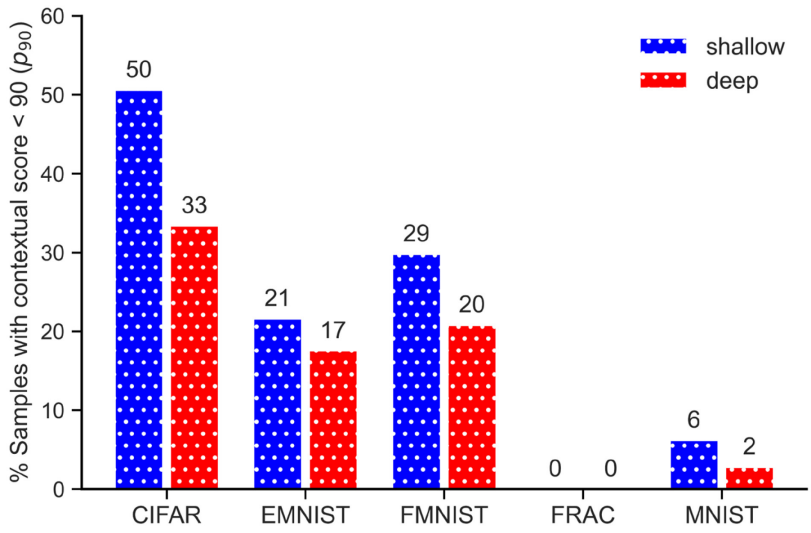

Fig. 15: Comparison of the proportion of samples with contextual scores below 90% between deep and shallow models across the different datasets used in this study.

To further illustrate this point, we examine the proportion of samples with contextual scores below a more stringent threshold of 90%, offering a focused view of extreme underperformance. As shown in Fig. 15, the shallow models consistently yield a higher proportion of low-scoring samples compared to their deep counterparts across all datasets. This disparity is particularly pronounced in complex datasets such as EMNIST and FMNIST, suggesting that deeper architectures are more robust against severe contextual failures. These findings underscore the importance of examining tail-end performance, especially in applications where even a small fraction of highly unreliable predictions could have significant consequences. Consistent with the observations in Fig. 14, the fracture dataset (FRAC) contains no test image instances with contextual scores below 90, indicating a high level of confidence in individual predictions.

7. Conclusions

This study introduces contextual evidence, comprising contextual images and scores, as a tool to enhance trust in CNN model predictions used for classifying metal fractographs. The following are the key outcomes of this study:

1) Contextual images allow users to visually understand the similarities and unique features between queried test images and nearest training images. This visual comprehension enabled by contextual images allows users to draw conclusions about the complexity of model predictions, fostering trust and reliability, despite a lack of subject matter expertise.

2) The CNN model trained on the fracture dataset had 99.8% accuracy on the training dataset 99.9% accuracy on the testing dataset.
3) Contextual scores provide a quantitative framework for assessing confidence based on the proximity to decision boundaries in the latent space. Test images closer to decision boundaries could receive lower contextual scores, indicating lower confidence in predictions. This empowers users to make informed decisions and foster a higher level of trust in the trained CNN model. The CNN model trained on the fracture dataset had very high contextual scores improving confidence on the CNN predictions.
4) The mean contextual score provides a quantitative measure to assess the confidence in the model's predictions across multiple instances. The mean contextual score serves to instill trust in the model by demonstrating its consistency and suitability in effectively handling the dataset at hand. The mean contextual score of the fracture dataset is 98 implying that the predictions over the entire dataset are very reliable.
5) The study also demonstrates the effectiveness of contextual evidence in enhancing trust and confidence by applying it on diverse datasets, including MNIST, EMNIST, FMNIST, and CIFAR10.

Overall, contextual evidence improves transparency, understanding, and trust in CNN models, reinforcing confidence in their outcomes. The framework of contextual evidence can be extended to various neural network architectures used in engineering mechanics, addressing trust concerns at different user levels.

Acknowledgment: Research presented in this paper was supported by the National Science Foundation under NSF EPSCoR Track-1 Cooperative Agreement OIA #1946202 and NSF CAREER award 2329562. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Lee JW, Goo NH, Park WB, Pyo M, Sohn KS. Virtual microstructure design for steels using generative adversarial networks. *Engineering Reports*. 2021;3(1):e12274.

2. Zhang Y, Seibert P, Otto A, Raßloff A, Ambati M, Kästner M. DA-VEGAN: Differentiably Augmenting VAE-GAN for microstructure reconstruction from extremely small data sets. *Computational Materials Science*. 2024;232:112661.

3. Li H, Zhang J, Hu L, Su K. Notch fatigue life prediction of micro-shot peened 25CrMo4 alloy steel: A comparison between fracture mechanics and machine learning methods. *Engineering Fracture Mechanics*. 2023;277:108992.

4. Diller J, Siebert L, Winkler M, et al. An integrated approach for detecting and classifying pores and surface topology for fatigue assessment 316L manufactured by powder bed fusion of metals using a laser beam using μ CT and machine learning algorithms. *Fatigue & Fracture of Engineering Materials & Structures*. 2024;47(9):3392-3407.

5. Zhao Y, Xiang Y, Tang K. Machine learning-based fatigue life prediction of lamellar titanium alloys: A microstructural perspective. *Engineering Fracture Mechanics*. 2024;303:110106.

6. Wang H, Li B, Zhang W, Xuan F. Microstructural feature-driven machine learning for predicting mechanical tensile strength of laser powder bed fusion (L-PBF) additively manufactured Ti6Al4V alloy. *Engineering Fracture Mechanics*. 2024;295:109788.

7. Awd M, Münstermann S, Walther F. Effect of microstructural heterogeneity on fatigue strength predicted by reinforcement machine learning. *Fatigue & Fracture of Engineering Materials & Structures*. 2022;45(11):3267-3287.

8. Avilés-Cruz C, Aguilar-Sanchez M, Vargas-Arista B, Garfias-Garcia E. A new machine learning-based evaluation of ductile fracture. *Engineering Fracture Mechanics*. 2024;302:110072.

9. Liu C, Kim J, Song J-J, et al. Intelligent recognition and identification of fracture types and parameters for borehole images based on developed convolutional neural networks and post-processing. *Engineering Fracture Mechanics*. 2023;292:109624.

10. Naik DL, Sajid HU, Kiran R. Texture-based metallurgical phase identification in structural steels: A supervised machine learning approach. *Metals*. 2019;9(5):546.

11. Shang H, Wang S, Zhou L, Lou Y. Neural network-based ductile fracture model for 5182-O aluminum alloy considering electroplastic effect in electrically-assisted processing. *Engineering Fracture Mechanics*. 2023;290:109476.

12. Ma S, Tian K, Sun Y, Yang C, Yang Z. Fatigue life prediction of composite bolted joints based on finite element model and machine learning. *Fatigue & Fracture of Engineering Materials & Structures*. 2024;47(6):2029-2043.

13. Yu M, Xie X, Fang Z, Lim JB. A novel machine-learning based framework for calibrating micromechanical fracture model of ultra-low cycle fatigue in steel structures. *Engineering Fracture Mechanics*. 2024;306:110200.

14. Fan J, Wang Z, Liu C, Shi D, Yang X. A tensile properties-related fatigue strength predicted machine learning framework for alloys used in aerospace. *Engineering Fracture Mechanics*. 2024;301:110057.

15. Rahman SA, Chandraker A, Prakash O, Chauhan A. Data-driven machine learning approach for predicting dwell fatigue life in two classes of Titanium alloys. *Engineering Fracture Mechanics*. 2024;306:110214.

16. Mahmoodzadeh A, Fakhri D, Mohammed AH, Mohammed AS, Ibrahim HH, Rashidi S. Estimating the effective fracture toughness of a variety of materials using several machine learning models. *Engineering Fracture Mechanics*. 2023;286:109321.

17. Yang Y, Zhang B, Wu H, et al. A deep learning approach for low-cycle fatigue life prediction under thermal–mechanical loading based on a novel neural network model. *Engineering Fracture Mechanics*. 2024;306:110239.

18. Bahrami B, Talebi H, Ayatollahi MR, Khosravani MR. Artificial neural network in prediction of mixed-mode I/II fracture load. *International Journal of Mechanical Sciences*. 2023;248:108214.

19. Wang J, Zhang Y, He Y, et al. A deep neural network-based method to predict J-integral for surface cracked plates under biaxial loading. *Engineering Fracture Mechanics*. 2024;302:110062.

20. Hassani Niaki M, Pashaian M. Using deep learning method to predict dimensionless values of stress intensity factors and T-stress of edge notch disk bend (ENDB) specimen. *Fatigue & Fracture of Engineering Materials & Structures*. 2024;47(8):2789-2802.

21. Zhang J, Guo W. A physics knowledge-based neural network method for three-dimensional fracture mechanics of attachment lugs. *Engineering Fracture Mechanics*. 2024;306:110215.

22. He G, Zhao Y, Yan C. Multiaxial fatigue life prediction using physics-informed neural networks with sensitive features. *Engineering Fracture Mechanics*. 2023;289:109456.

23. Halamka J, Bartošák M, Španiel M. Using hybrid physics-informed neural networks to predict lifetime under multiaxial fatigue loading. *Engineering Fracture Mechanics*. 2023;289:109351.

24. Zhang X-C, Gong J-G, Xuan F-Z. A physics-informed neural network for creep-fatigue life prediction of components at elevated temperatures. *Engineering Fracture Mechanics*. 2021;258:108130.

25. Li H, Sun G, Tian Z, Huang K, Zhao Z. A physics-informed neural network framework based on fatigue indicator parameters for very high cycle fatigue life prediction of an additively manufactured titanium alloy. *Fatigue & Fracture of Engineering Materials & Structures*. 2024;47(9):3171-3188.

26. He G, Zhao Y, Yan C. A physics-informed generative adversarial network framework for multiaxial fatigue life prediction. *Fatigue & Fracture of Engineering Materials & Structures*. 2023;46(10):4036-4052.

27. Zhou T, Sun X, Yu Z, Chen X. A generalization ability-enhanced image recognition based multiaxial fatigue life prediction method for complex loading conditions. *Engineering Fracture Mechanics*. 2024;295:109802.

28. Naser M. An engineer's guide to eXplainable Artificial Intelligence and Interpretable Machine Learning: Navigating causality, forced goodness, and the false perception of inference. *Automation in Construction*. 2021;129:103821.

29. Frie C, Riza Durmaz A, Eberl C. Exploration of materials fatigue influence factors using interpretable machine learning. *Fatigue & Fracture of Engineering Materials & Structures*. 2024;

30. Bastidas-Rodriguez MX, Polania L, Gruson A, Prieto-Ortiz F. Deep Learning for fractographic classification in metallic materials. *Engineering Failure Analysis*. 2020;113:104532.

31. Alqahtani H, Bharadwaj S, Ray A. Classification of fatigue crack damage in polycrystalline alloy structures using convolutional neural networks. *Engineering Failure Analysis*. 2021;119:104908.

32. Maruschak P, Konovalenko I, Sorochak A. Methods for evaluating fracture patterns of polycrystalline materials based on the parameter analysis of ductile separation dimples: A review. *Engineering Failure Analysis*. 2023;153:107587.

33. Loyola-Gonzalez O. Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view. *IEEE access*. 2019;7:154096-154113.

34. Naik DL, Kiran R. A novel sensitivity-based method for feature selection. *Journal of Big Data*. 2021/10/09 2021;8(1):128. doi:10.1186/s40537-021-00515-w

35. Kiran R, Naik DL. Novel sensitivity method for evaluating the first derivative of the feed-forward neural network outputs. *Journal of Big Data*. 2021;8(1):1-13.

36. Arumugam D, Kiran R. Interpreting denoising autoencoders with complex perturbation approach. *Pattern Recognition*. 2023;136:109212.

37. Deng L. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*. 2012;29(6):141-142.

38. Cohen G, Afshar S, Tapson J, Van Schaik A. EMNIST: Extending MNIST to handwritten letters. IEEE; 2017:2921-2926.

39. Xiao H, Rasul K, Vollgraf R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:170807747*. 2017;

40. Krizhevsky A, Nair V, Hinton G. The CIFAR-10 dataset. *online: http://www cs toronto edu/kriz/cifar html*. 2014;55(5)

41. Naik DL, Kiran R. Identification and characterization of fracture in metals using machine learning based texture recognition algorithms. *Engineering Fracture Mechanics*. 2019;219:106618.

42. Ding H, Zhu T, Wang X, Yang B, Xiao S, Yang G. An uncoupled ductile fracture model considering void shape change and necking coalescence. *Engineering Fracture Mechanics*. 2023;292:109612.

43. Dwivedi A, Khan I, Chattopadhyay J. On the role of shape and distribution of secondary voids in the mechanism of coalescence. *Engineering Fracture Mechanics*. 2023;289:109399.

44. O'Shea K, Nash R. An introduction to convolutional neural networks. *arXiv preprint arXiv:151108458*. 2015;

45. Arumugam D, Kiran R. Compact representation and identification of important regions of metal microstructures using complex-step convolutional autoencoders. *Materials & Design*. 2022;223:111236.

46. Janocha K, Czarnecki WM. On loss functions for deep neural networks in classification. *arXiv preprint arXiv:170205659*. 2017;

47. Kingma DP, Ba J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:14126980*. 2014;