Neural Computing and Applications
https://doi.org/10.1007/s00521-022-07593-8

ORIGINAL ARTICLE

Prediction of hydraulic blockage at culverts from a single image using deep learning

Umair Iqbal[1] · Johan Barthelemy[1] · Pascal Perez[1]

Received: 6 October 2021 / Accepted: 28 June 2022

Abstract

Cross-drainage hydraulic structures such as culverts and bridges in urban landscapes are prone to get blocked by the transported debris (e.g., urban, vegetated), which often reduces their hydraulic capacity and triggers flash floods. Unavailability of relevant data from blockage-originated flooding events and complex nature of debris accumulation are highlighted factors hindering the research within the blockage management domain. Wollongong City Council (WCC) blockage conduit policy is the leading formal guidelines to incorporate blockage into design guidelines; however, are criticized by the hydraulic engineers for its dependence on the post-flood visual inspections (i.e., visual blockage) instead of peak floods hydraulic investigations (i.e., hydraulic blockage). Apparently, no quantifiable relationship is reported between the visual blockage and hydraulic blockage; therefore, many consider WCC blockage guidelines invalid. This paper exploits the power of Artificial Intelligence (AI), motivated by its recent success, and attempts to relate visual blockage with hydraulic blockage by proposing a deep learning pipeline to predict hydraulic blockage from an image of the culvert. Two experiments are performed where the conventional pipeline and end-to-end learning approaches are implemented and compared in the context of predicting hydraulic blockage from a single image. In experiment one, the conventional deep learning pipeline approach (i.e., feature extraction using CNN and regression using ANN) is adopted. In contrast, in experiment two, end-to-end deep learning models (i.e., E2E_ MobileNet, E2E_ BlockageNet) are trained and compared with the conventional pipeline approach. Dataset (i.e., Hydraulics-Lab Blockage Dataset (HBD), Visual Hydraulics-Lab Dataset (VHD)) used in this research were collected from laboratory experiments performed using scaled physical models of culverts. E2E_ BlockageNet model was reported best in predicting hydraulic blockage with R^2 score of 0.91 and indicated that hydraulic blockage could be interrelated with the visual features at the culvert.

Keywords Cross-Drainage Hydraulic Structures · Visual Blockage · Hydraulic Blockage · Artificial Intelligence · Deep Learning · End-to-End Learning · Scaled Physical Models

1 Introduction

Blockage of cross-drainage hydraulic structures such as culverts and bridges is a commonly occurring phenomenon during floods which often results in a reduced hydraulic capacity of the structure, increased damages to property, diversion of flow, downstream scour, failure of the structure, and risk to life [13, 22, 25, 26, 34, 53–55]. Few highlighted examples of blockage-originated floods around the world include Newcastle (Australia) floods [25, 61], Barpeta (India) floods [59], Pentre (United Kingdom) floods [15] and Wollongong (Australia) floods [25, 54]. In the context of Australia, many councils and institutions have mentioned blockage as a critical issue (e.g., NSW Floodplain Management Manual [49], Queensland Urban Drainage Manual [35], Australian Rainfall and Runoff (ARR) [10, 26, 50, 62]), however, none comprehensively addressed consideration of blockage into design guidelines. Research in blockage management is hindered by the highly variable nature of blockage formulation and the unavailability of historical floods data to investigate the behavior of blockage [16, 17, 38]. Wollongong City Council (WCC), under the umbrella of ARR, developed a

✉ Umair Iqbal
ui010@uowmail.edu.au

[1] SMART Infrastructure Facility, University of Wollongong, Wollongong, Australia

Published online: 28 July 2022

conduit blockage policy for the first time to incorporate the blockage within the design guidelines [36, 62]. The WCC policy suggested that any hydraulic structure with a diagonal length less than 6m is prone to 100% blockage during peak floods.

The problem of blockage at cross-drainage hydraulic structures has been studied from two main perspectives (i.e., hydraulically, visually) based on the subjective interpretation by researchers in literature. The WCC blockage policy was developed under the “visual blockage” perspective and was based on the post-flood visual surveys of cross-drainage hydraulic structures. Visual blockage is defined as the function of visual hindrance caused by debris material at the opening of cross-drainage hydraulic structures. The idea behind this perspective is that the probability of blockage-originated floods can be significantly reduced by regular maintenance of cross-drainage hydraulic structures using the visual blockage information [12, 32, 62]. The WCC blockage policy was criticized by the hydraulic engineers because of its dependence on visual assessments rather than hydraulic assessments. This introduced the perspective of hydraulic blockage, which is defined as the reduction in the hydraulic capacity of the structure due to the presence of debris material [62]. This perspective emphasizes the need to investigate the quantifiable hydraulic impacts of blockage during peak floods to include it within the design guidelines of the cross-drainage hydraulic structures.

It is argued that “visual blockage” assessed from post-flood visual information cannot be considered as the true representation of the “hydraulic blockage” during peak floods until a quantifiable translation exists between both terms [27]. One highlighted case differentiating both terms is when a structure is blocked with porous vegetative debris. For this case, the degree of visual blockage will be high, but the degree of hydraulic blockage will be very low. Therefore, a structure with high visual blockage doesn’t need to be hydraulically blocked. To date, there is no quantifiable relationship reported in the literature to translate visual blockage into hydraulic blockage or otherwise.

In recent times, the world has seen the success of computational intelligence [2–5] and Artificial Intelligence (AI) [6, 32, 63] approaches towards solving real-world problems. Generally, in the context of computational analysis, the nonlinear activation analyses have been widely used to address forward (e.g., dynamic analyses) and reverse (e.g., fault diagnosis) problems across various application domains in the literature. Some highlighted examples include the nonlinear semi-continuum model for material analysis [42], a mechanoelectrical flexible hub-beam model for fluid analysis [31] and First Order Approximate Coupling (FOAC) model for hub-beam dynamic analysis [20]. Specifically, in the context of deep learning, the aim of a model is to get linearly related separable features from nonlinearly separable input instances by performing multiple transformations over the number of layers [24]. To achieve nonlinearity in the neural networks, various types of activation functions are used. Few commonly used nonlinear activation functions are Sigmoid, Tanh, Rectified Linear Unit (ReLU) [48], Swish [51], and Mish [46]. In earlier days of deep learning, Sigmoid and Tanh activation functions were used; however, they were limited by vanishing gradient and complexity. ReLU-based functions were introduced to deal with the complexity and presented a simpler concept; however, they offered limitations of limited nonlinearity and the non-utilization of negative values. To deal with the saturated output problem of existing activation functions, exponential unit based activation functions were introduced (e.g., Exponential Linear Unit (ELU) [21], Scaled ELU (SELU) [37]). In recent times, learning-based adaptive activation functions (e.g., Adaptive Piecewise Linear (APL) [7], Swish) are proposed, which have the ability to adapt the parameters during learning and are hence more robust. Few most recent nonlinear activation functions proposed for images datasets include Wide Hidden Expansion (WHE) [60], Soft-Root-Sign (SRS) [65] and Pade Activation Unit (PAU) [47].

A significant shift has been observed in literature from local hand-crafted features (i.e., conventional machine learning) to deep features (i.e., deep learning) for improved and generalized performance. Motivated by this success, this paper implements the combination of the latest deep learning CNN (i.e., best suited for the images) and ANN (i.e., latest regression model) architectures to predict hydraulic blockage from a single image. Research in this paper attempts to relate hydraulic blockage with visual blockage by proposing the use of a culvert image for the prediction of corresponding hydraulic blockage. In this context, two experiments are reported where a conventional deep learning pipeline approach and end-to-end deep learning approach are implemented. The conventional deep learning pipeline consisted of three modules; extraction of visual features from an image using the CNN model (i.e., MobileNet, ResNet50, EfficientNetB3), pre-processing of the extracted deep visual features, and predicting the hydraulic blockage by feeding it to regression model (i.e., Artificial Neural Network (ANN)). In experiment two, the functionality of the conventional pipeline proposed in experiment one is achieved by using a single end-to-end deep learning model. In this context, two end-to-end deep learning models (i.e., E2E_ MobileNet, E2E_ BlockageNet) were trained and compared with the best performing conventional deep learning pipeline from experiment one. The dataset (i.e., Hydraulics-Lab

Blockage Dataset (HBD), Visual Hydraulics-Lab Dataset (VHD)) used in this research was collected from a series of comprehensive laboratory experiments performed using scaled physical models of culverts to replicate different flooding and blockage scenarios. As a summary, the followings are the main contributions of the presented research in this article:

1. Development of numerical (i.e., HBD) and visual (i.e., VHD) datasets from the hydraulics laboratory experiments to facilitate the implementation of AI algorithms.
2. Design, implementation, and analysis of a conventional deep learning pipeline using CNN and ANN algorithms to predict the hydraulic blockage at cross drainage hydraulic structure from a single image of the culvert.
3. Development and analysis of end-to-end deep learning models for the improved prediction of hydraulic blockage from a single image of the culvert.

The rest of the paper is organized as follows: Section 2 summarizes the latest benchmark research where blockage at cross-drainage hydraulic structures is addressed. Section 3 presents the research methodology, including data collection, deep learning architectures, and research approach for experiments. Section 4 talks about the experimental design and evaluation measures used to assess the performance. Section 5 presents the results of the experiments and reports the important insights. Section 6 discusses the results and builds the important accounts generated from the performed experiments. Section 7 concludes the study and provides the potential future directions of the presented research.

2 State of the art in blockage management

The problem of blockage at cross-drainage hydraulic structures is not comprehensively addressed in the literature primarily because of the limited availability of data and the highly complex nature of blockage accumulation. This section summarizes the benchmark literature related to blockage management in chronological order to demonstrate the advancements in this domain.

In year 2010, Balkham et al. [9] studied blockage problem in the context of the United Kingdom using a risk-based methodology. For local hydraulic structures, detailed guidelines were formulated to deal with the blockage issue at culverts and bridges. Later in 2013, Blanc [16] performed laboratory experiments to investigate the impact of trash screens on the upstream blockage. Straight wooden dowels of varying lengths were used to replicate the wooden debris. The study concluded that the probability of the trash screen being blocked increased with an increase in the debris length relative to trash screen bar spacing. The study lacked to discuss the impact of blockage on upstream water levels during the peak floods and used a simplified definition of blockage, which may not be valid in practice.

In the year 2015, Manning-Dickfos [43] validated the existing blockage guidelines in the context of the Sunshine Coast region by performing open channel laboratory experiments using scaled physical models of the culvert. The blockage effect was simulated by controlling the flow using a gate mechanism. From the results, it was concluded that blockage is more critical at lower flow rates in comparison to the higher flow rates. The impact of debris at the upstream flood levels and the accumulation behavior of debris were not studied. Later in the same year, Kramer et al. [38] proposed the mathematical formulation of hydraulic blockage and performed laboratory experiments to investigate the impact of urban debris on upstream flood levels. From the investigation, varied trends of blockage were reported for different debris types indicating the complexity of the blockage problem. Furthermore, alignment of debris and type of debris were reported as two main factors significantly affecting the blockage outcome.

In 2016, Sullivan et al. [57] proposed the idea of using the remote sensing data towards identifying the hydraulic structures susceptible to blockage issue. The idea of automatically detecting the debris piles and classifying them into one of three classes (i.e., small, medium, large) was coined, however, no computer vision algorithm was reported in this context. In 2020, Brooks [19] investigated the blockage of culverts due to boulders by performing laboratory experiments using scaled physical models. From the field observations and corresponding laboratory investigations, the inlet of the culvert was reported as the dominant location for the boulder deposition, and multiple culvert designs were proposed to counter this problem. In 2021, Iqbal et al. [33] investigated the blockage at culverts by performing laboratory experiments using scaled physical models of culverts. A comprehensive study was undertaken where multiple debris types were used, and different blockage scenarios were simulated to explore the relationships between blockage-related factors (e.g., debris orientation, culvert type, inlet discharge, debris type, debris volume). From the investigations, interesting trends were reported where the blockage was found highly dependent on the debris orientation, debris compactness, culvert type, and debris type. Further, it was reported that hydraulic blockage increases towards the falling limb of the flood hydrograph, however, it may not be as critical as during the peak floods.

In a most recent study, Iqbal et al. [32] investigated the blockage from a visual perspective and proposed the use of AI models for the automation of visual blockage classification for maintenance purposes. The idea of using a computer vision algorithm to classify a culvert as "blocked" or "clear" was used to automate the manual visual surveys performed by the flood management teams for maintaining the structures. Nine CNN classification models were implemented on the manually labeled data from the real-culverts and laboratory experiments. From the results, the NASNet model was reported as the best with a classification accuracy of 85%, however, the MoibleNet was reported as the fastest with a classification accuracy of 78%. Background cluttering and simplified labeling criteria were identified as the main factors in the degraded performance of CNN models.

3 Methodology

3.1 Data collection

Two different types of datasets (i.e., hydraulic and visual) were used in this research collected from comprehensive laboratory experiments using scaled physical models of culverts (see Iqbal et al. [33] for more details). The experiments aimed to replicate blockage scenarios using multiple debris types under different flooding conditions and record visual and hydraulic data. Percentage hydraulic blockage was recorded using mathematical formulation proposed by Kramer et al. [38] as given in Eq. 1.

$$\text{Percentage Hydraulic Blockage} = \frac{\text{Upstream WL}_{\text{blocked}} - \text{Upstream WL}_{\text{unblocked}}}{\text{Upstream WL}_{\text{blocked}}} \times 100, \tag{1}$$

where Upstream $\text{WL}_{\text{blocked}}$ denotes the upstream water level when the culvert is blocked and Upstream $\text{WL}_{\text{unblocked}}$ denotes the upstream water level when the culvert is not blocked.

Experiments were performed in a 12m × 0.2m flume with single and double circular culvert models. Vegetative and urban debris was used at scale to simulate different blockage scenarios. Figure 1 shows the Two-Dimensional (2D) schematic diagram of the experimental setup used to collect the dataset. A point gauge was used to measure the water levels and was placed at 1m distance from the culvert. In total, 173 unique blockage scenarios were simulated, while some scenarios were repeated. A total of 352 hydraulic data samples were recorded from the experiments to organize in a dataset called HBD.

In addition to the hydraulic data collection, a web camera-based setup was established to record the videos of each simulated blockage scenario, and the corresponding dataset is referred to as VHD. For this investigation, images were extracted from VHD for the time instances when the hydraulic measurements were taken. In total, 352 images were extracted from video clips, each representing the visual of the culvert for the time instance at which the corresponding hydraulic measurement was taken.

3.2 Deep learning architectures

Deep learning is an approach within machine learning that uses the multiple layer structure to automatically extract the feature representations without human involvement. Deep learning models work in a hierarchy where lower to higher level features are learned as the network goes deep [28]. Recently, deep learning-based models and pipelines have successfully addressed complex real-world problems because of their ability to learn useful features automatically and provide a generalized performance. In the context of deep learning, end-to-end learning has emerged as an approach to use the power of layered structure and model the intermediate operations in conventional pipelines using the network layers. Therefore, end-to-end learning is defined as the approach of training a complex target system represented by a single deep neural network and bypassing the intermediate layer operations [18]. The followings are the theoretical details of the deep learning architectures used in the presented research.

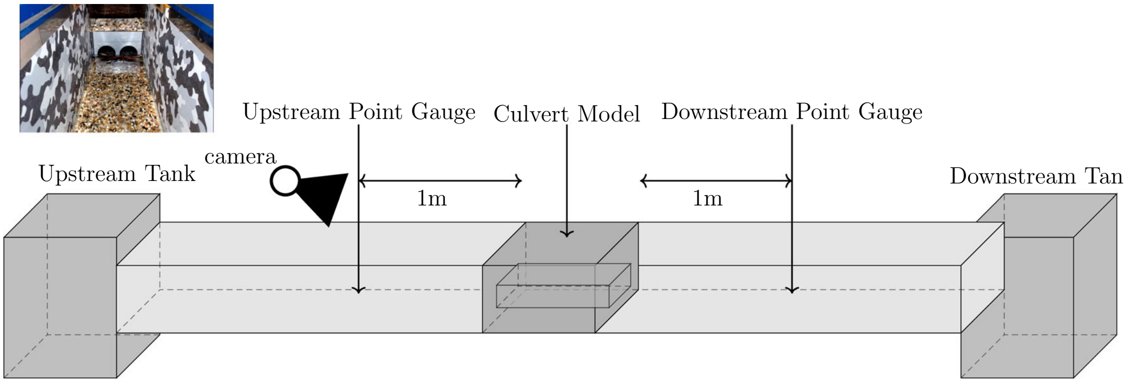


Fig. 1 Two-dimensional schematic diagram of hydraulics laboratory experimental setup

3.2.1 Convolutional neural networks (CNNs)

CNNs or ConvNets are deep feedforward networks inspired by the visual cortex functionality and considered one of the most powerful networks to process grid-like data (i.e., visual data). CNN follows the different approach of connectivity inspired by the visual cortex and is way less complex in terms of connectors in comparison to a fully connected multilayer perceptron. In the visual cortex, a single neuron responds to stimuli within a limited region, also known as the receptive field. These receptive fields overlap partially with each other and cover the entire visual field [40]. In general, a CNN architecture consists of convolution layers, pooling layers, activation functions, fully connected layers, classifiers, loss functions, optimizers, and regularization. The following subsections provide a brief introduction to CNN models used in this research as feature extractors.

3.2.2 MobileNet

MobileNets are the class of deep networks specifically designed for the mobile utility and consist of compact, streamlined architecture [30]. Depthwise separable convolution, a form of factorize convolution [56] makes them computationally cheaper deep networks. First, a single filter is applied to each input channel and then 1 × 1 pointwise convolution is applied to combine the depthwise convolution outputs (i.e., depthwise separable convolution consists of separate filtering and combining layers). The accuracy and latency of the network are controlled by two hyperparameters (i.e., width multiplier, resolution multiplier) to help in building a model suitable for a custom problem. Depthwise convolution for a single filter per input can be expressed mathematically as in Eq. 2.

$$\hat{\mathbf{G}}_{k,l,m} = \sum_{i,j} \hat{\mathbf{K}}_{i,j,m} \cdot \mathbf{F}_{k+i-1,l+j-1,m}, \tag{2}$$

where $\hat{\mathbf{K}}$ denotes the depthwise kernel for convolution, $\mathbf{F}$ denotes the input channel, and $\hat{\mathbf{G}}$ denotes the filtered output feature map. Figure 2 shows the graphical illustration of how the depthwise separable convolution works and how it is different from the standard convolution.

3.2.3 ResNet50

A deep residual network was proposed by He et al. [29] towards improving the training of extremely deep networks by introducing the idea of reformulating layers as learning residual functions instead of unreferenced functions. From empirical results, ResNet proved easier to optimize and improved the accuracy with increased depth of the network. In other words, ResNet allows the network layers to fit residual mapping instead of fitting for each layer. If $H(x)$ represents the mapping to be fit by layers of the network with input (x), the residual learning is based on the hypothesis that if a certain number of layers can asymptotically approximate the complicated function, they can also approximate the residual function $(F(x) := H(x) - x)$.

3.2.4 EfficientNetB3

EfficientNet is proposed by Tan and Le [58] as an accurate and efficient family of ConvNets based on the scaled-up version of the baseline Neural Architecture Search (NAS) model. The idea of using a simple compound coefficient to uniformly scale the model in all dimensions (i.e., depth, width, resolution) is implemented in developing EfficientNets. Scaling up ConvNets by balancing all dimensions using a constant ratio resulted in better accuracy of models. Based on this idea, if it is intended to use 2^n times more computational power, model can be scaled up in depth by a^n, in width by b^n and in resolution by c^n, where a, b, c represent constants.

3.2.5 Artificial neural network (ANN)

ANNs are machine learning models inspired by the biological functionality of the animal brain and are layer-based deep architecture. ANN consists of nodes, layers, and connections. Each node in the network represents a neuron and applies a transformation to input by non-linear activation and transmits it to other neurons in the network. Layers of ANN consist of a number of nodes and are designed to perform a specific transformation to input. Furthermore, each layer is characterized by weights which are updated during the training process to optimize the desired performance of layer [1, 14, 39, 45]. A neuron k in layer $L+1$ takes x_i^L as input and transforms it by applying non-linear activation into x_k^{L+1}. The processing of a single neuron in the network can be mathematically expressed as given in Eq. 3.

$$x_k^{L+1} = f \left. \sum_i w_{ik}^L x_i^L + w_{bk}^L \right), \tag{3}$$

where w_{ik}^L represents the layer L weights, w_{bk}^L represents the bias term of neuron k and f represents the non-linear activation function.

3.3 Research approach

This section presents a detailed description of the conventional deep learning based pipeline and end-to-end deep learning models implemented in this article for the

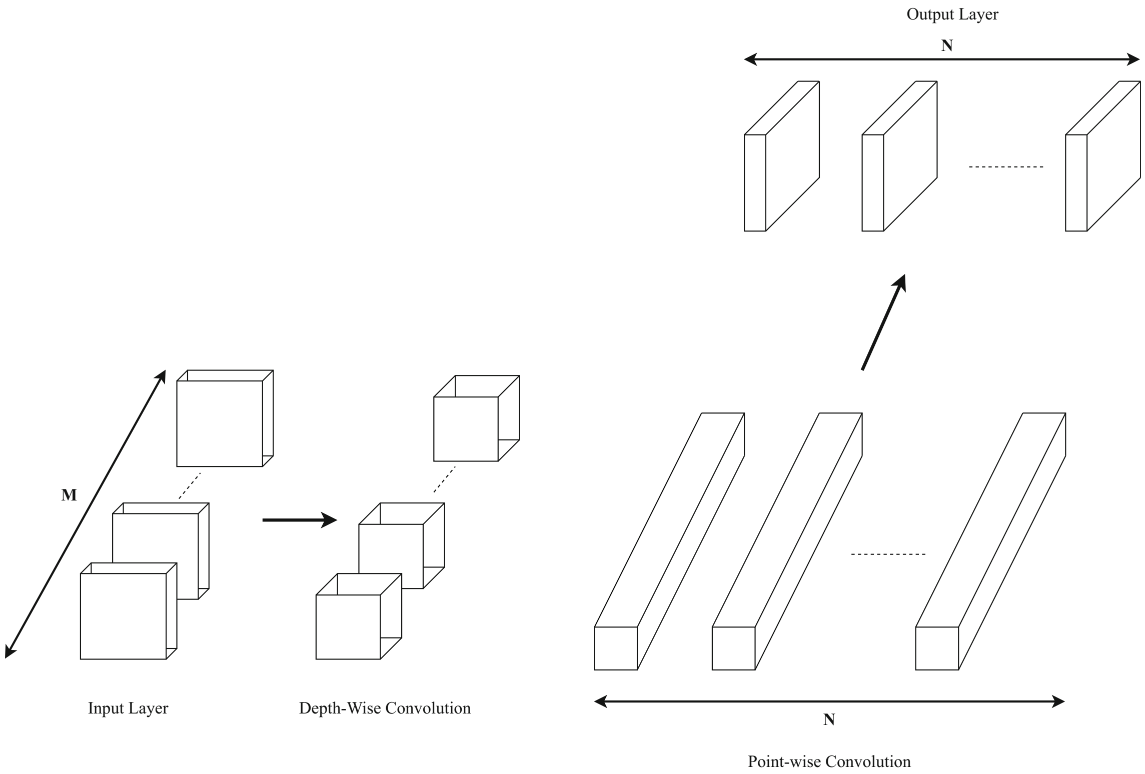


Fig. 2 Graphical illustration of depthwise separable convolution (conceptualized from [30])

prediction of hydraulic blockage from a single image of the culvert.

3.3.1 Experiment one: conventional deep learning pipeline

The proposed deep learning pipeline aimed to relate the visual blockage with hydraulic blockage using a combination of deep learning models and consisted of three main modules; visual feature extraction, data processing, and ANN regression. The proposed pipeline was designed to take an image of the culvert, extract visual features using a deep learning model, pre-process the extracted features, and feed them into the ANN regression model to predict the hydraulic blockage. Figure 3 shows the functional block diagram of the proposed deep learning pipeline.

- **Module 1: Deep Visual Feature Extraction –** As a first step in the pipeline, an image of the culvert is processed through a deep CNN model (e.g., MobileNet, ResNet50, EfficientNetB3) to extract the deep visual features. In experiment one, three CNN models are compared to assess the impact of the number of visual features extracted and the fundamental principle by which the visual features are extracted. All the CNN models were used with ImageNet [23] pre-trained weights and as a feature extractor by removing the top layers.
- **Module 2: Data Processing –** At the second step of the pipeline, extracted visual features were transformed before feeding to the regression model for improved performance. The standard scalar transformation was applied, which transforms the data with a distribution having zero mean and unit standard deviation. Given a sample x, standard scalar transformation score z can be determined as given in Eq. 4.

$$z = \frac{x - \mu}{\sigma}, \tag{4}$$

  where μ represents the mean and σ represents the standard deviation. In literature, it has been reported that standard scalar transformation improves the performance of the regression model in comparison to the case where no transformation is applied.
- **Module 3: ANN Regression –** At the final stage of the proposed pipeline, processed visual features were fed into the ANN regression model to predict corresponding hydraulic blockage. Three different regression models with different layer depths were trained corresponding to the number of extracted visual features.

3.3.2 Experiment two: end-to-end deep learning model

In experiment two, end-to-end models were designed to achieve the functionality of the proposed pipeline in

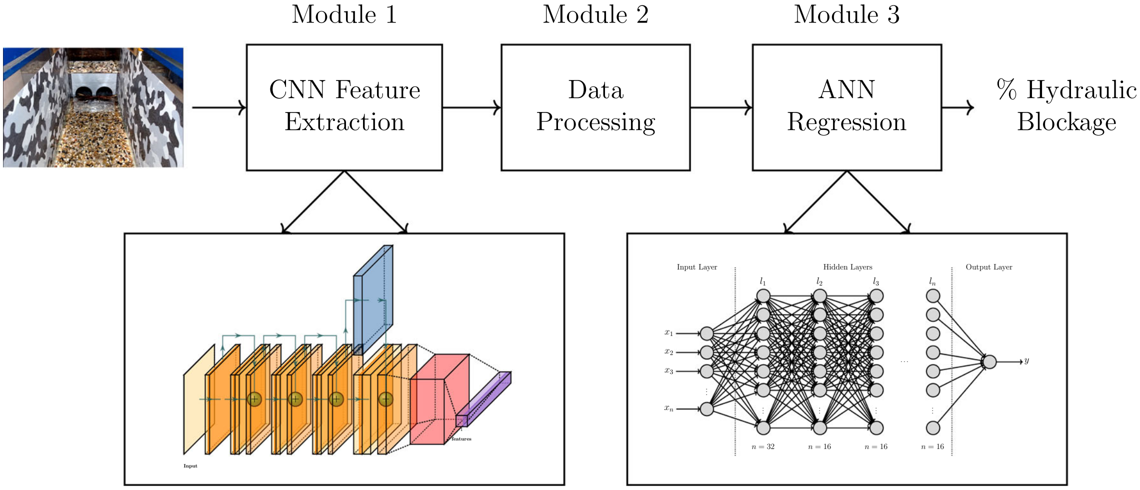


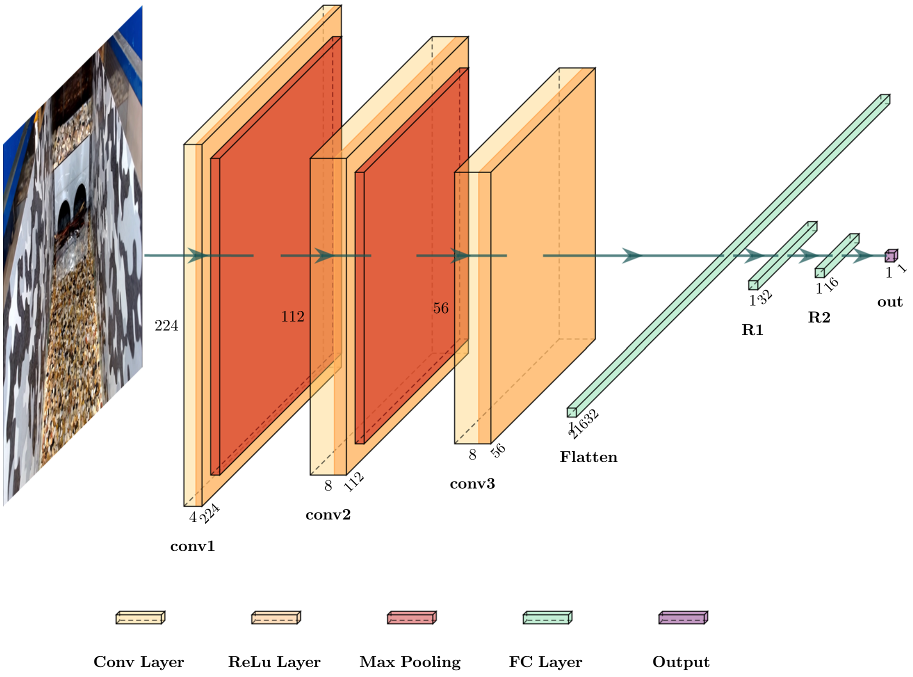
Fig. 3 Functional block diagram of proposed conventional deep learning pipeline for blockage prediction

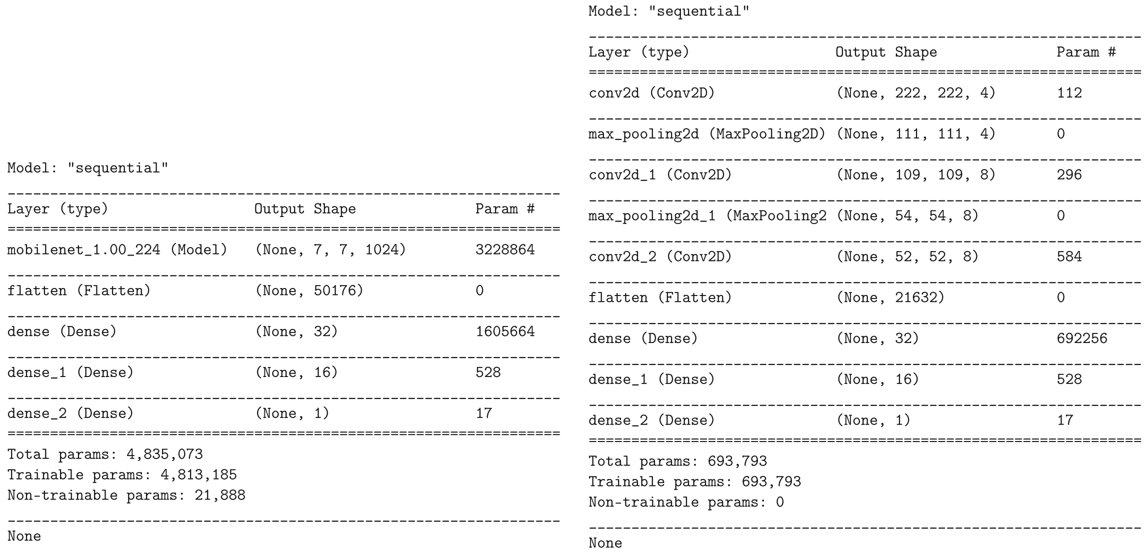
experiment one for predicting hydraulic blockage. A single model architecture with CNN layers as feature extractor and fully connected dense layers as ANN regressor was designed. Based on the results of experiment one, two end-to-end models were designed, one with MobileNet as feature extractor (i.e., E2E_ MobileNet) and the other with custom CNN layers as feature extractor (i.e., E2E_ BlockageNet). Figure 4 shows the structure of the end-to-end deep learning model (i.e., E2E_ BlockageNet) for the prediction of hydraulic blockage. Models were designed and trained using Keras and TensorFlow platforms. Figure 5 shows the summary of both models with the number of parameters and features at each layer.

4 Experimental design and evaluation measures

Two sets of experiments (i.e., conventional deep learning pipeline, end-to-end deep learning) were performed in this article to predict the hydraulic blockage at culverts from images. Experimental design for both investigations is presented in this section, along with standard evaluation measures.

Fig. 4 Structure of proposed end-to-end deep learning model E2E_ BlockageNet for hydraulic blockage prediction





```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
mobilenet_1.00_224 (Model)   (None, 7, 7, 1024)        3228864
_________________________________________________________________
flatten (Flatten)            (None, 50176)             0
_________________________________________________________________
dense (Dense)                (None, 32)                1605664
_________________________________________________________________
dense_1 (Dense)              (None, 16)                528
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 17
=================================================================
Total params: 4,835,073
Trainable params: 4,813,185
Non-trainable params: 21,888
_________________________________________________________________
None
```

(a) E2E_MobileNet

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 222, 222, 4)       112
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 111, 111, 4)       0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 109, 109, 8)       296
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 54, 54, 8)         0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 52, 52, 8)         584
_________________________________________________________________
flatten (Flatten)            (None, 21632)             0
_________________________________________________________________
dense (Dense)                (None, 32)                692256
_________________________________________________________________
dense_1 (Dense)              (None, 16)                528
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 17
=================================================================
Total params: 693,793
Trainable params: 693,793
Non-trainable params: 0
_________________________________________________________________
None
```

(b) E2E_BlockageNet

Fig. 5 Summaries of proposed end-to-end deep learning models

4.1 Experiment one

Experiment one implemented a conventional deep learning pipeline approach and investigated the performance of three different CNN models (i.e., MobileNet, ResNet50, EfficientNetB3) as feature extractors to select the best among them. All the CNN models were pre-trained on the ImageNet dataset and were used as feature extractor by removing the top layers. Each CNN model resulted in a different number of visual features (i.e., MobileNet = 50176, ResNet50 = 100352, EfficientNetB3 = 153600), therefore, three variants of ANN in terms of the number of hidden layers were used to locally optimize the training. ANN1 was used with MobileNet features, ANN2 was used with ResNet50 features, and ANN3 was used with EfficientNetB3 features, respectively. The depth of hidden layers was decided based on a trial and error process with the criteria that an increasing number of hidden layers do not improve the performance anymore. Table 1 presents the information about the three variants of ANN. All the ANN models were trained for 100 epochs with Adam optimizer and a constant learning rate of 0.001. A standard 60:20:20 split of the dataset was used for training, validation, and testing. Furthermore, Mean Absolute Error (MAE) was used as a loss metric during the training process. Models were trained using NVIDIA GeForce RTX 2060 Graphical Processing Unit (GPU) with 6GB memory and 14 Gbps memory speed.

4.2 Experiment two

Experiment two implemented and trained two end-to-end deep learning models (i.e., E2E_ MobileNet, E2E_ BlockageNet) based on the results of experiment one. For E2E_ MobileNet, CNN layers from the pre-trained MobileNet model were used as feature extractor, and fully-connected layers were stacked on top of CNN layers to achieve the regression functionality. However, in the case of the E2E_ BlockageNet model, three CNN layers were used as feature extractor with 4, 8, and 8 filters, respectively. Both the models were trained using Adam optimizer with a learning rate of 0.001 for 100 epochs. MAE was used as a loss metric during the training. A conventional holdout dataset split of 60:20:20 was used for training, validation, and testing.

4.3 Evaluation measures

The performance of implemented models was assessed over unseen test data using standard evaluation metrics, including Mean Squared Error (MSE), MAE, and R^2 score.

Table 1 ANN regression model variants investigated in experiment one

	# of hidden layers	# of nodes	# of input features	# of trainable parameters
ANN1	2	[32,16]	50176	1606209
ANN2	8	[32, 16,..., 16]	100352	3213505
ANN3	10	[32, 16,..., 16]	153600	4917985

MSE measures the model's absolute goodness for fit and is calculated by dividing the sum of the square of prediction error (i.e., actual minus predicted) by the total number of data samples. It gives an absolute real number which informs about how much the predicted results deviate from the actual results. MSE is best suited for the comparison of different regression models and the selection of the best model against the compared. Equation 5 presents the mathematical expression for the MSE.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\delta_i - \hat{\delta}_i\right)^2, \tag{5}$$

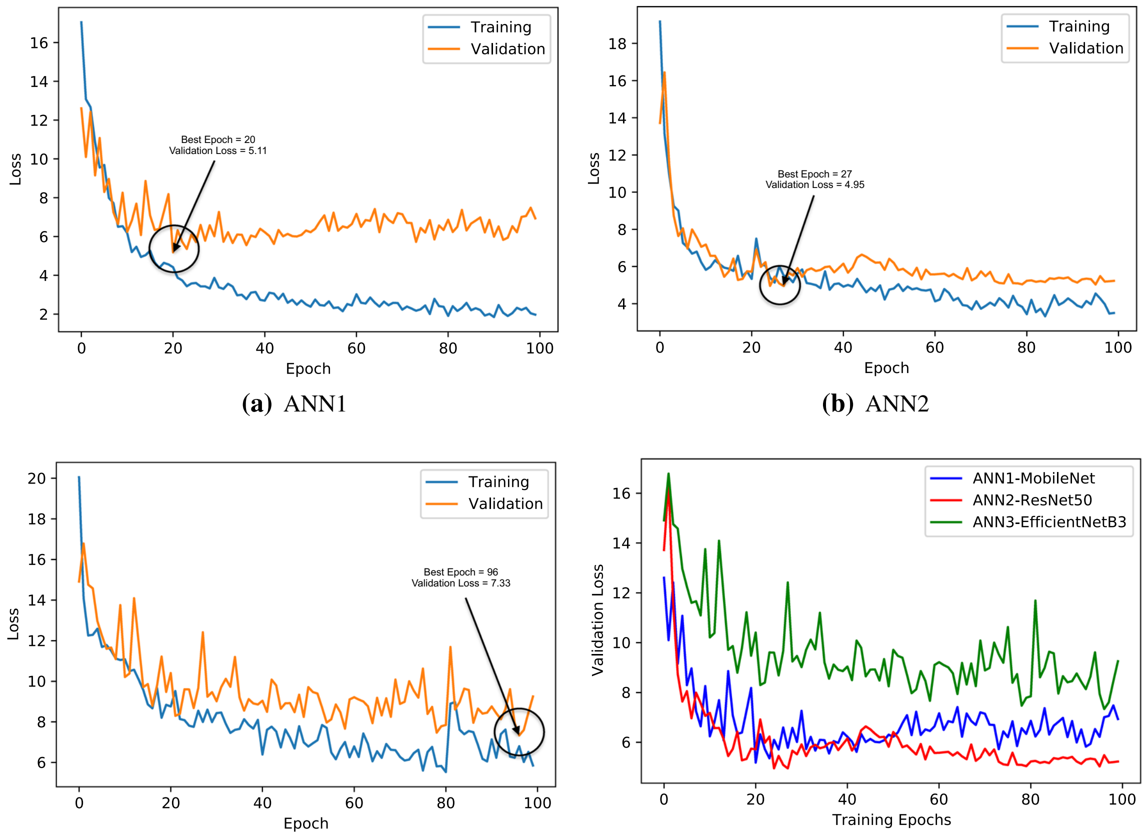
where n denotes the total number of data samples, δ denotes the actual output, and $\hat{\delta}$ denotes the predicted output.

MAE is similar to MSE but takes the sum of the absolute value of error instead of the square value of error. It measures the mean error without considering the direction. It is most suited for the case when the training data contains possible outliers. Equation 6 presents the mathematical expression for the calculation of MAE.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\delta_i - \hat{\delta}_i|. \tag{6}$$

R^2 score is one of the most commonly used measure for evaluating the regression model performance. It measures the capability of a model to explain the dependent variable variability and is calculated by squaring the correlation coefficient (R). By definition, R^2 score or coefficient of determination is the percentage measure of the model's ability to replicate the observed results. R^2 is considered an important measure in machine learning regression because it provides the goodness of fit for a machine learning model (i.e., how well the model predictions approximate the actual data) [64]. Mathematically, it is calculated by dividing the sum of the square of prediction error by the total sum of the square. Equation 7 presents the mathematical expression for the calculation of R^2 [64].

$$R^2 = 1 - \frac{\sum_i\left(\delta_i - \hat{\delta}_i\right)^2}{\sum_i\left(\delta_i - \bar{\delta}_i\right)^2}. \tag{7}$$



(a) ANN1

(b) ANN2

(c) ANN3

(d) Validation Loss Comparison

Fig. 6 Training performance of ANN regression models investigated in experiment one

5 Results

This section presents the results of the conventional deep learning pipeline and end-to-end models investigated in experiments one and two. Results are presented as empirical numerical summaries, training plots, scatter plots, prediction plots, and error box plots. Furthermore, important insights interpreted from the results are reported and discussed.

5.1 Experiment one

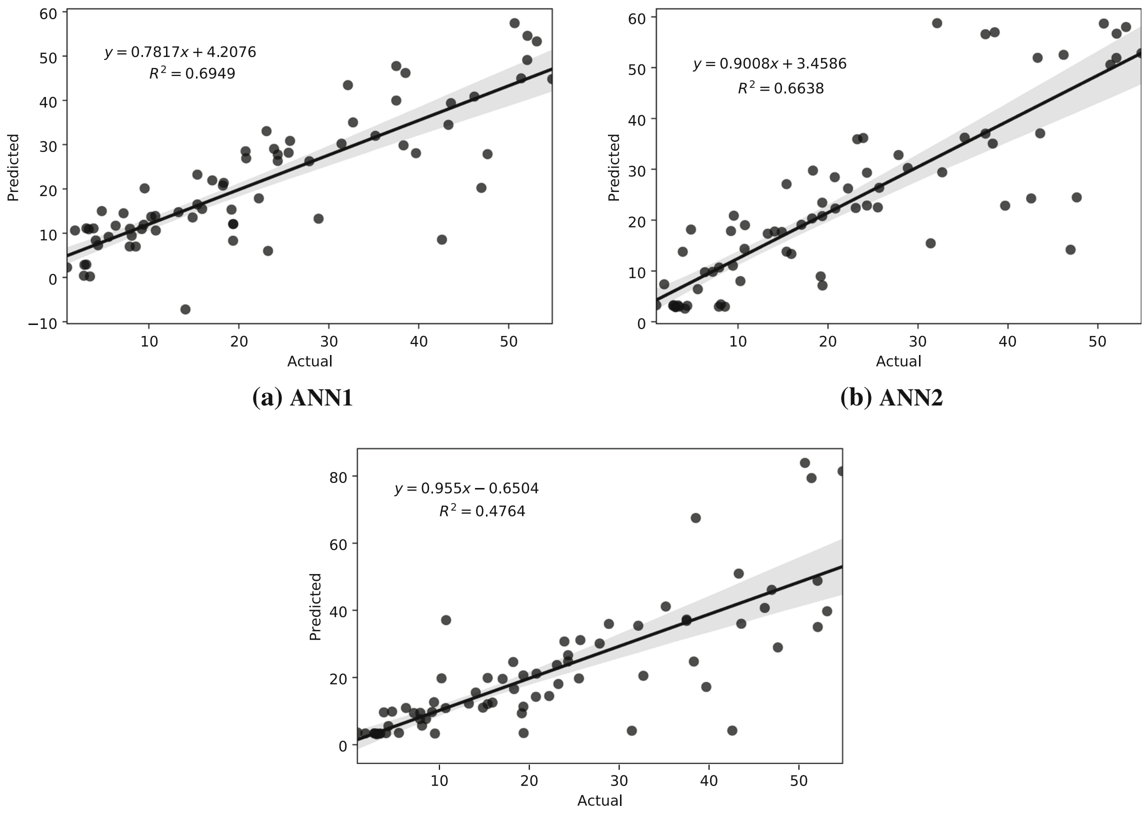
Implemented ANN regression models in the conventional deep learning pipeline were assessed for their training and testing performance. Training performance was evaluated by monitoring the individual loss curves and comparative plots. Figure 6 shows the training plots for the implemented models. From the training plots, it can be observed that for all cases, training loss followed a negative exponential curve while validation loss tried to follow the training curve. This is the indication of a normal training process in machine learning. From the comparative plot in Fig. 6d, it can be observed from validation loss curves that both ANN1 and ANN2 performed relatively the same, with ANN1 on the slightly better end.

Table 2 presents the summary of recorded quantitative test results for the implemented regression models. From the Table 2, it can be observed that the ANN1 model produced the best results with R^2 of 0.6949. Interestingly, it has been observed that with the increase in the number of

Table 2 Summary of empirical results for Implemented ANN Regression Models in Experiment One

	MSE	MAE	R^2
ANN1 (with MobileNet features)	**76.15**	**6.22**	**0.6949**
ANN2 (with ResNet50 features)	83.90	6.23	0.6638
ANN3 (with EfficientNetB3 features)	130.70	7.13	0.4764

Bold values represent the best performing model

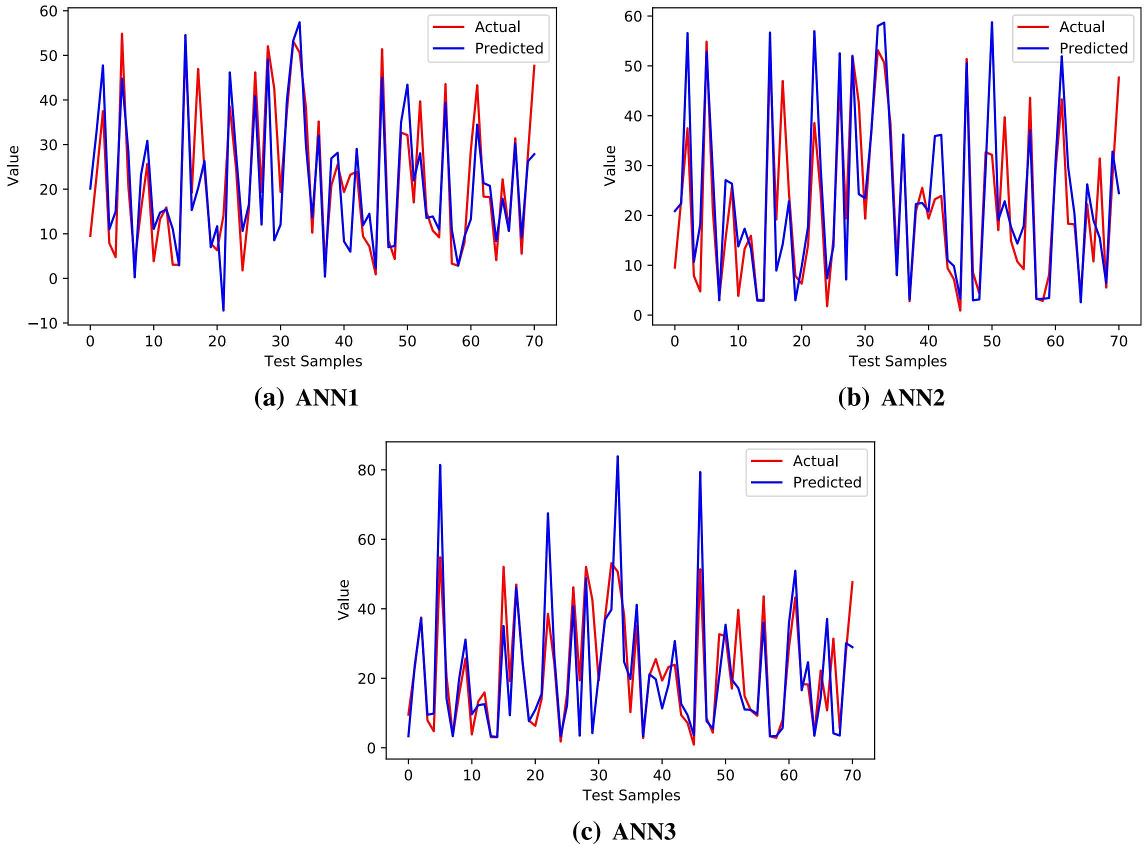


(a) ANN1

(b) ANN2

(c) ANN3

Fig. 7 Scatter plots for implemented ANN regression models in experiment one



(a) ANN1 **(b) ANN2**

(c) ANN3

Fig. 8 Actual vs. predicted Plots for implemented ANN regression models in experiment one

deep visual features, the performance of ANN regression degraded. This may be attributed to the presence of a large number of irrelevant and uncorrelated features for the ANN2 and ANN3 cases.

Figure 7 shows the scatter plots for each ANN regression model. From the plots, it is evident that ANN1 produced the best fit on test data. Figure 8 shows the actual vs. predicted plots for all three ANN models to demonstrate how well each model was able to track the actual value. ANN1 model was observed to best track the actual values, however, over-prediction can be observed in the majority of data instances. In all three cases, over-prediction was more dominant in comparison to under-prediction. Figure 9 shows the box plots of absolute error for the implemented models in experiment one. From the box plot, it can be observed that the least spread of box (i.e., containing 50% of samples) is for the ANN1, indicating more consistent performance. Furthermore, the maximum error was recorded least for the ANN1 (i.e., ≈16). The number of outliers was least for the ANN2, however, the spread of outliers was least for ANN1. Therefore, statistics indicate the comparable performance of ANN1 and ANN2, with ANN1 slightly on the better end.

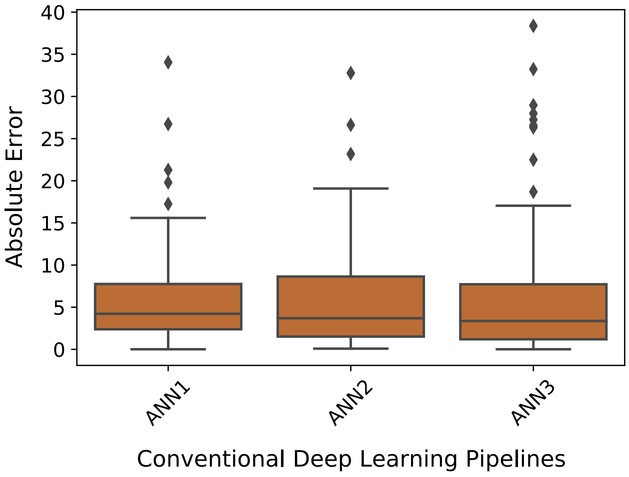
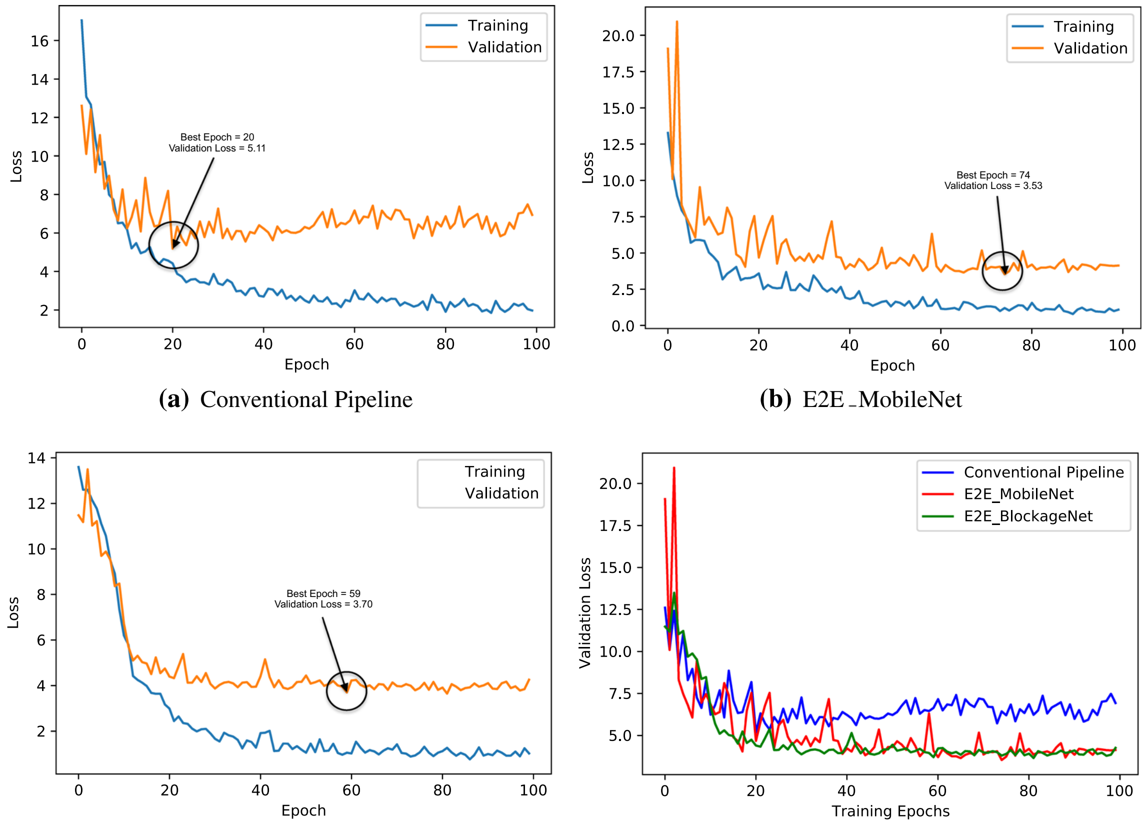


Fig. 9 Absolute error box plot for implemented ANN regression models in experiment one

5.2 Experiment two

Training and testing performance of proposed end-to-end deep learning models is reported and compared with the best conventional pipeline model combination from experiment one. Training performance was evaluated from loss curves and comparative plots. Figure 10 shows the



(a) Conventional Pipeline **(b)** E2E _MobileNet

(c) E2E_BlockageNet **(d)** Validation Loss Comparison

Fig. 10 Training performance of end-to-end deep learning models implemented in experiment two

Table 3 Summary of empirical results for end-to-end deep learning models implemented in experiment two

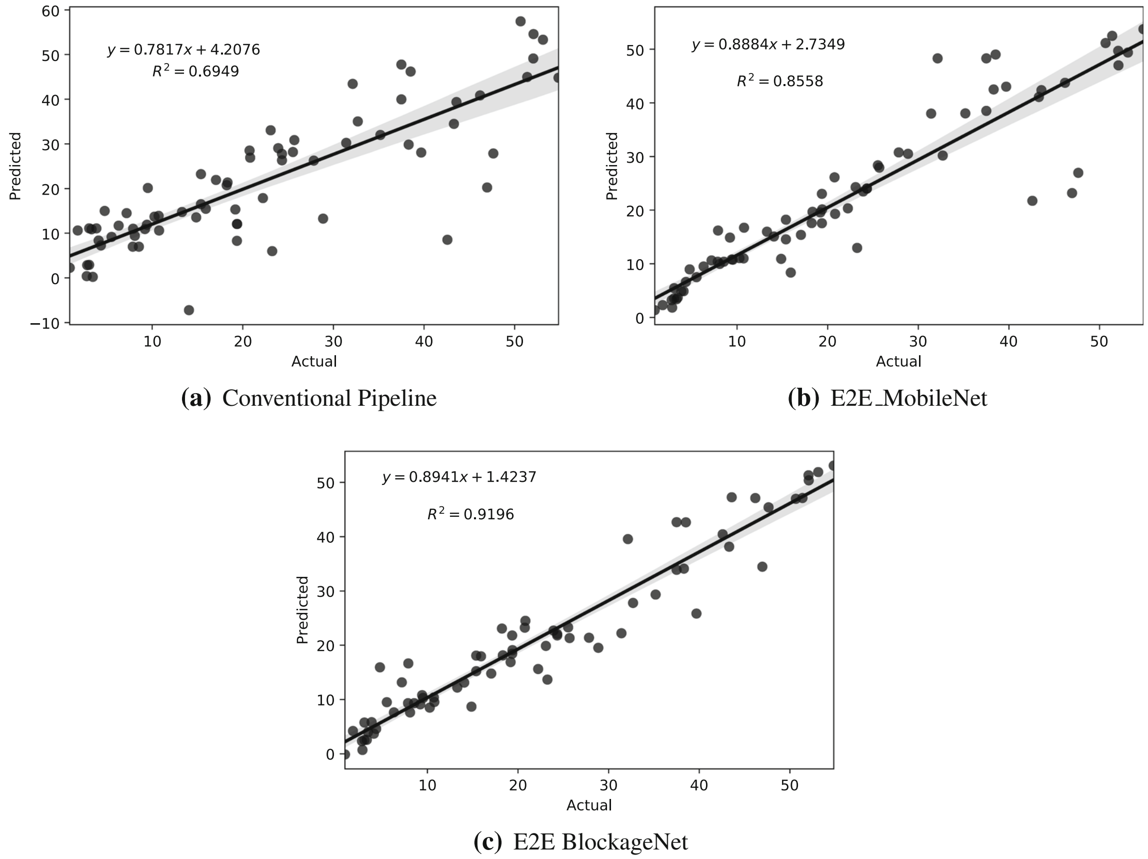
	MSE	MAE	R^2
Conventional pipeline	76.15	6.22	0.6949
E2E_ MobileNet	35.98	3.59	0.8558
E2E_ BlockageNet	**20.05**	**3.25**	**0.9196**

Bold values represent the best performing model

training plots for conventional pipeline and end-to-end models. In all cases, training loss followed the negative exponential curve, while validation loss followed the training loss, which is the indication of the normal training process. From Fig. 10d comparative plot, it can be observed that validation loss for E2E_ MobileNet and E2E_ BlockageNet was similar, however, E2E_ BlockageNet curve was more stable, indicating better training performance.

Table 3 presents the summary empirical results for end-to-end models and conventional pipeline. From the results, it can be observed that the E2E_ BlockageNet model performed best with R^2 score of 0.9196. From the relative comparison of conventional pipeline and E2E_ MobileNet, it can be seen that the end-to-end approach resulted in significantly improved performance (R^2 of 0.8558 in comparison to 0.6949). Figure 11 shows the scatter plots for conventional pipeline and end-to-end models to demonstrate the fit on the test data. From the scatter plots, it can be observed that E2E_ BlockageNet was able to best fit the test data.

Figure 12 shows the predicted vs. actual plots for all three models in experiment two. E2E_ BlockageNet was the one with the closest tracking of the actual test values. Figure 13 presents the box plots of absolute error for the implemented end-to-end models in experiment two. The box plot shows that E2E_ MobileNet has the least box spread suggesting more consistent performance. The E2E_ BlockageNet box plot was also comparable to E2E_ MobileNet, however, with the least number of outliers (i.e., 4) and condensed overall spread including outliers (i.e., the maximum error of approx 15). On the other hand, E2E_ MobileNet is better in terms of median and maximum error statistics, however, the number and spread of outliers suggest its slight degraded performance in comparison to E2E_ BlockageNet.



(a) Conventional Pipeline **(b)** E2E_MobileNet

(c) E2E BlockageNet

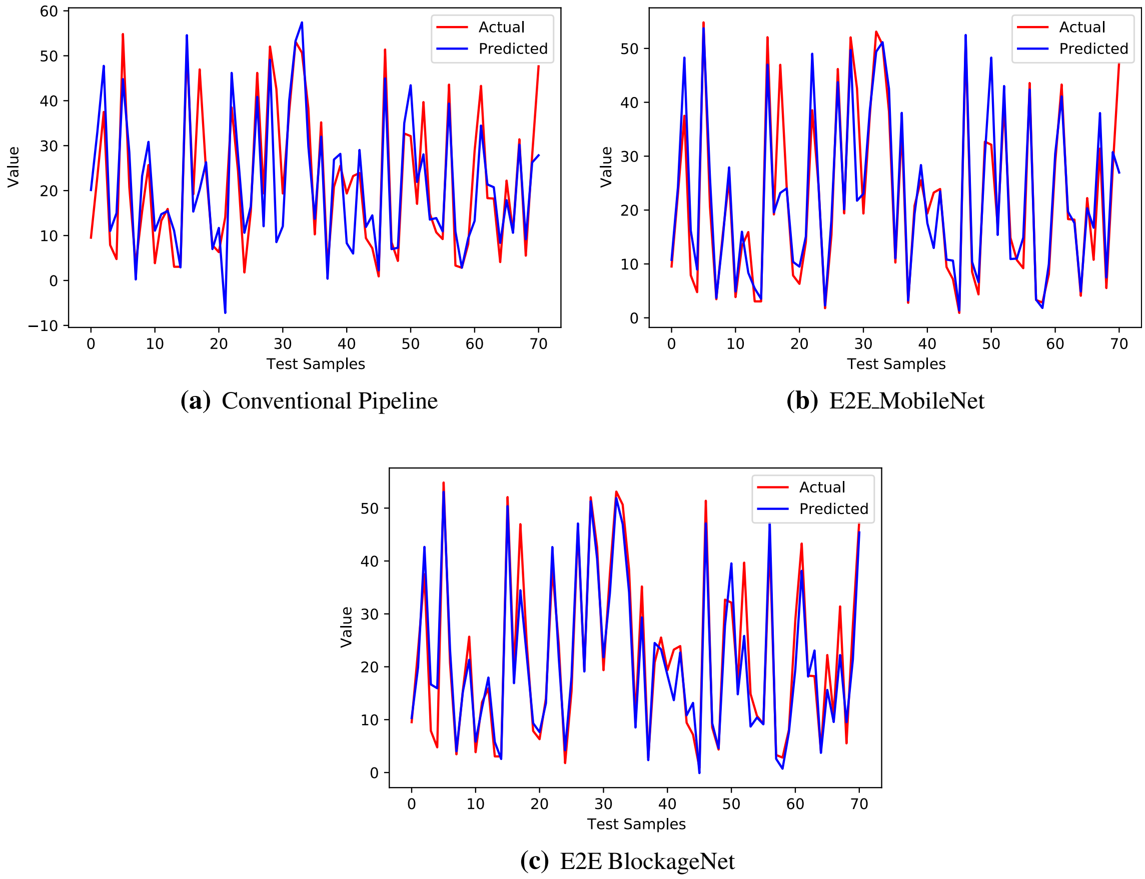
Fig. 11 Scatter plots for implemented end-to-end deep learning models in experiment two

6 Discussions on results

Results from both the experiments (i.e., conventional pipeline, end-to-end model) validated the hypothesis that both visual blockage and hydraulic blockage can be interrelated. A maximum R^2 score of 0.91 for the E2E_BlockageNet model and positive scores for all other cases are clear indicators that visual variations at culvert inlet due to the presence of debris material can be used in the prediction of corresponding hydraulic blockage, which otherwise is almost impossible to model using conventional mathematical modeling. Improved performance of end-to-end models relative to conventional deep learning pipeline is in line with the literature [18] and may be attributed to the capability of end-to-end models in self-optimizing the internal components of the network and learning the layer weights more cohesively.

It is important to mention that the dataset used for this investigation was recorded with the same background and lighting conditions with only variations in culvert type, debris, and water levels. This suggests that, for real-world application, as part of the calibration process, the camera should focus only on the culvert region, avoiding any vegetative background; otherwise, performance may significantly degrade given the visual similarity between vegetative background and vegetative debris material causing blockage. Furthermore, one may argue the dependence of visual appearance (i.e., visual features) on the other factors (e.g., lighting conditions, debris types, background, weather) besides the debris itself and may not consider it reliable for hydraulic blockage prediction. For this specific study, all these factors were controlled in the laboratory experiments, however, for real-world application, it is in the plans to develop a data pre-processing block to negate such irrelevant visual variations and account only for variations caused by the presence of debris.

Deployment of such AI-powered solutions to the complex real-world problems (e.g., pedestrian detection [11], wildlife monitoring [8], teaching analysis [44], traffic flow prediction [41], flood risk assessment [52, 66]) has been made possible by the recent technological advancements in computing hardware and availability of edge computing hardware (e.g., NVIDIA Jetson TX2, NVIDIA Jetson Nano, NVIDIA Jetson Xavier). Specifically for blockage management at culverts, recently, an AIoT-powered camera-based system has been implemented in Illawarra, New South Wales, Australia region where culverts are classified



(a) Conventional Pipeline (b) E2E_MobileNet

(c) E2E BlockageNet

Fig. 12 Actual versus predicted plots for implemented end-to-end deep learning models in experiment two

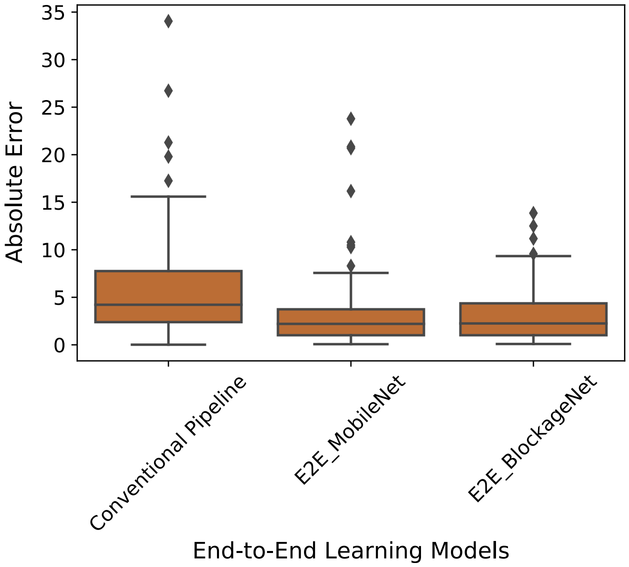


Fig. 13 Absolute error box plot for implemented end-to-end deep learning models in experiment two

into visual blockage categories using the latest computer vision algorithms (see [12] for more details). Therefore, it is potentially possible to design and deploy such a system for the presented research where it would be able to predict the hydraulic blockage given the image of the culvert captured by the camera in real-time. As a system, it will take the raw image of the culvert as input, will process the image to mitigate the irrelevant visual dependencies, will apply the trained end-to-end deep learning model to predict the hydraulic blockage, and will share the real-time statistics through the cloud on a mobile application or web dashboard. The proposed trained model can be used as a base model, which later may be fine-tuned using the real-world data. A camera, edge computing hardware (e.g., NVIDIA Jetson Nano), and 5G communication module will be the major hardware components, while cloud services (e.g., Amazon Web Services (AWS)), AI development, and dashboard development will be the major software components for such system.

7 Conclusion

Deep learning pipeline and end-to-end deep learning models have been successfully implemented and compared by performing two experiments in the context of predicting the hydraulic blockage from a single image of the culvert.

Experiment one implemented a conventional deep learning pipeline using CNN and ANN to extract the visual features and predict the hydraulic blockage, respectively. MobileNet CNN model with two-layer ANN (i.e., ANN1) was reported best with R^2 score of 0.69. Regression performance was observed to be degraded with the increase in the number of extracted visual features, which may be attributed to the presence of increased number of irrelevant and uncorrelated features. Experiment two implemented end-to-end deep learning models to achieve the functionality of the conventional deep learning pipeline and compared the results. From the results of experiment two, the end-to-end learning approach was reported to outperform the conventional pipeline by a significant margin (i.e., R^2 of 0.91 for E2E_ BlockageNet in comparison to 0.69 for the conventional pipeline). Improved performance of end-to-end models may be attributed to their capability of self-optimizing the internal components of the network. A positive R^2 score for all cases validated the hypothesis of the existence of a relation between visual features of the culvert and corresponding hydraulic blockage. The performance of proposed models is expected to be degraded significantly for the cases where the image contains a background with a similar visual appearance to the debris material blocking the culvert. The development of data preprocessing techniques to mitigate the visual variations caused by other factors (e.g., lighting, debris type, background, weather) is a potential future research direction. Furthermore, deployment of the proposed approach using the AIoT infrastructure for real-world culvert sites is also planned in the near future.

Acknowledgements I would like to thank the Wollongong City Council (WCC) for funding this investigation. This research was funded by the Smart Cities and Suburb Program (Round Two) of the Australian Government, grant number SCS69244. Further, I would like to thank the Higher Education Commission (HEC) of Pakistan and the University of Wollongong (UOW) for funding my Ph.D studies. Thanks to Associate Professor Wanqing Li for providing technical guidance.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Declarations

Conflict of Interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Abraham A (2005) Artificial neural networks. Handbook of measuring system design
2. Abu Arqub O (2017) Adaptation of reproducing kernel algorithm for solving fuzzy fredholm-volterra integrodifferential equations. Neural Comput Appl 28(7):1591–1610
3. Abu Arqub O, Abo-Hammour Z (2014) Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm. Inf Sci 279:396–415
4. Abu Arqub O, Maayah B (2018) Solutions of bagley-torvik and painlevé equations of fractional order using iterative reproducing kernel algorithm with error estimates. Neural Comput Appl 29(5):1465–1479
5. Abu Arqub O, Rashaideh H (2018) The rkhs method for numerical treatment for integrodifferential algebraic systems of temporal two-point bvps. Neural Comput Appl 30(8):2595–2606
6. Afzaal H, Farooque AA, Abbas F, Acharya B, Esau T (2019) Groundwater estimation from major physical hydrology components using artificial neural networks and deep learning. Water 12(1):5
7. Agostinelli F, Hoffman M, Sadowski P, Baldi P (2014) Learning activation functions to improve deep neural networks. arXiv preprint arXiv:1412.6830
8. Arshad B, Barthelemy J, Pilton E, Perez P (2020) Where is my deer?-wildlife tracking and counting via edge computing and deep learning. 2020 IEEE Sens. IEEE, Rotterdam, Netherlands, pp 1–4
9. Balkham M, Fosbeary C, Kitchen A, Rickard C (2010) Culvert design and operation guide. Construction and industry research and information association, London, UK
10. Ball J, Babister M, Nathan R, Weinmann P, Weeks W, Retallick M, Testoni I (2016) Australian rainfall and runoff-a guide to flood estimation
11. Barthélemy J, Verstaevel N, Forehead H, Perez P (2019) Edge-computing video analytics for real-time traffic monitoring in a smart city. Sensors 19(9):2048
12. Barthelemy J, Amirghasemi M, Arshad B, Fay C, Forehead H, Hutchison N, Iqbal U, Li Y, Qian Y, Perez P (2020) Problem-driven and technology-enabled solutions for safer communities: The case of stormwater management in the illawarra-shoalhaven region (nsw, australia). In: Augusto JC (ed) Handbook of smart cities. Springer, pp 1–28
13. Barthelmess A, Rigby E (2011) Culvert blockage mechanisms and their impact on flood behaviour. In: Proceedings of the 34th world congress of the international association for hydro- environment research and engineering, Engineers Australia, Barton, ACT, pp 380–387
14. Basheer IA, Hajmeer M (2000) Artificial neural networks: fundamentals, computing, design, and application. J Microbiol methods 43(1):3–31
15. BBC (2021) Pentre flood: 'woody debris' blocking culvert was main cause, report finds. https://www.bbc.com/news/uk-wales-57686085
16. Blanc J (2013) An analysis of the impact of trash screen design on debris related blockage at culvert inlets. PhD thesis, School of the Built Environment, Heriot-Watt University

17. Blanc J, Wallerstein NP, Arthur S, Wright GB (2014) Analysis of the performance of debris screens at culverts. Proceed Inst Civ Eng-Water Manag Thomas Telford Ltd 167:219–229
18. Bojarski M, Del Testa D, Dworakowski D, Firner B, Flepp B, Goyal P, Jackel LD, Monfort M, Muller U, Zhang J, et al (2016) End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316
19. Brooks JA (2020) Culvert blockage caused by boulders in the western cape and the development of mitigation measures: physical model study. Master's thesis, Department of Civil Engineering, Stellenbosch University, Western Cape, South Africa
20. Cai GP, Hong JZ, Yang SX (2005) Dynamic analysis of a flexible hub-beam system with tip mass. Mech Res Commun 32(2):173–190
21. Clevert DA, Unterthiner T, Hochreiter S (2015) Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289
22. Davis A (2001) An analysis of the effects of debris caught at various points of major catchments during wollongong's august 1998 storm event. Bachelor of Engineering Thesis, University of Wollongong
23. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE, Miami, FL, USA, pp 248–255
24. Dubey SR, Singh SK, Chaudhuri BB (2021) A comprehensive survey and performance analysis of activation functions in deep learning. arXiv preprint arXiv:2109.14545
25. French R, Jones M (2015) Culvert blockages in two Australian flood events and implications for design. Australasian J Water Resourc 19(2):134–142
26. French R, Jones M (2018) Design for culvert blockage: the arr 2016 guidelines. Australasian J Water Resourc 22(1):84–87
27. French R, Rigby E, Barthelmess A (2012) The non-impact of debris blockages on the august 1998 Wollongong flooding. Australasian J Water Resourc 15(2):161–169
28. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT press
29. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE, Las Vegas, NV, USA, pp 770–778
30. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861
31. Hu W, Huai Y, Xu M, Feng X, Jiang R, Zheng Y, Deng Z (2021) Mechanoelectrical flexible hub-beam model of ionic-type solvent-free nanofluids. Mech Syst Signal Process 159:107833
32. Iqbal U, Barthelemy J, Li W, Perez P (2021a) Automating visual blockage classification of culverts with deep learning. Appl Sci 11(16)
33. Iqbal U, Barthelemy J, Perez P, Cooper J, Li W (2021b) A scaled physical model study of culvert blockage exploring complex relationships between influential factors. Australasian J Water Resourc pp 1–14
34. Iqbal U, Perez P, Li W, Barthelemy J (2021) How computer vision can facilitate flood management: a systematic review. Int J Disaster Risk Reduct 53:102030
35. Jones N, Lawson C (1991) The queensland urban drainage manual. Local Govern Eng Assoc Queensland J 9(4th quarter)
36. Jones RH, Weeks W, Babister M (2016) Review of conduit blockage policy summary report. WMA Water, 160 Clarence Street Sydney, NSW, 2000
37. Klambauer G, Unterthiner T, Mayr A, Hochreiter S (2017) Self-normalizing neural networks. Adv Neural Inf Process Syst 30
38. Kramer M, Peirson W, French R, Smith G (2015) A physical model study of culvert blockage by large urban debris. Australasian J Water Resourc 19(2):127–133
39. Krogh A (2008) What are artificial neural networks? Nature Biotechnol 26(2):195–197
40. LeCun Y, Bengio Y et al (1995) Convolutional networks for images, speech, and time series. Handbook Brain Theory Neural Networks 3361(10):1995
41. Li C, Xu P (2021) Application on traffic flow prediction of machine learning in intelligent transportation. Neural Comput Appl 33(2):613–624
42. Li C, Wang P, Luo Q (2020) A nonlinear semi-continuum model for silicon micro/nanosheets and its application in bending and vibration. Int J Mod Phys B 34(27):2050252
43. Manning-Dickfos B (2015) Developing more appropriate culvert blockage factors for use in flood modelling studies on the sunshine coast. Bachelor's Thesis, Department of Civil Engineering, University of the Sunshine Coast, Queensland, Australia
44. Mao W (2021) Video analysis of intelligent teaching based on machine learning and virtual reality technology. Neural Comput Appl pp 1–12
45. Mehrotra K, Mohan CK, Ranka S (1997) Elements of artificial neural networks. MIT press
46. Misra D (2019) Mish: A self regularized non-monotonic activation function. arXiv preprint arXiv:1908.08681
47. Molina A, Schramowski P, Kersting K (2019) Pad\'e activation units: end-to-end learning of flexible activation functions in deep networks. arXiv preprint arXiv:1907.06732
48. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: ICML, pp 807–814
49. NSW (2005) Floodplain development manual. New South Wales Government, Sydney, Australia
50. Ollett P, Syme B, Ryan P (2017) Australian rainfall and runoff guidance on blockage of hydraulic structures: numerical implementation and three case studies. J Hydrol (New Zealand) 56(2):109–122
51. Ramachandran P, Zoph B, Le QV (2017) Searching for activation functions. arXiv preprint arXiv:1710.05941
52. Rezaeianzadeh M, Tabari H, Yazdi AA, Isik S, Kalin L (2014) Flood flow forecasting using ann, anfis and regression models. Neural Comput Appl 25(1):25–37
53. Rigby E, Silveri P (2001) The impact of blockages on flood behaviour in the wollongong storm of august 1998. In: 6th conference on hydraulics in civil engineering: the state of hydraulics. Engineers Australia, Barton, ACT, pp 107–115
54. Rigby E, Silveri P (2002) Causes and effects of culvert blockage during large storms. In: Ninth international conference on urban drainage (9ICUD). Engineers Australia, Lloyd Center Doubletree Hotel, Portland, Oregon, United States, pp 1–16
55. Roso S, Boyd M, Rigby E, VanDrie R (2004) Prediction of increased flooding in urban catchments due to debris blockage and flow diversions. In: Proceedings of NOVATECH, pp 8–13
56. Sifre L, Mallat S (2014) Rigid-motion scattering for image classification. PhD thesis, Ecole Polytechnique
57. Sullivan JL, McFaden S, Engel T, et al. (2016) Using remote data collection to identify bridges and culverts susceptible to blockage during flooding events. Technical report. University of Vermont. Transportation Research Center
58. Tan M, Le QV (2019) Efficientnet: rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946
59. Times H (2021) Blockage in culvert leads to inundation. https://www.thehindu.com/news/cities/Tiruchirapalli/blockage-in-culvert-leads-to-inundation/article37526114.ece

60. Wang M, Liu B, Foroosh H (2020) Wide hidden expansion layer for deep convolutional neural networks. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp 934–942
61. WBM B, (2008) Newcastle flash flood 8 June 2007 (the pasha bulker storm) flood data compendium. Prepared for Newcastle City Council, BMT WBM, Broadmeadow
62. Weeks W, Witheridge G, Rigby E, Barthelmess A, O'Loughlin G (2013) Project 11: blockage of hydraulic structures. Technical report, P11/S2/021, Engineers Australia, Water Engineering, 11 National Circuit Barton ACT 2600
63. Xenochristou M, Kapelan Z (2020) An ensemble stacked model with bias correction for improved water demand forecasting. Urban Water J 17(3):212–223
64. Zhang D (2017) A coefficient of determination for generalized linear models. Am Statist 71(4):310–316
65. Zhou Y, Li D, Huo S, Kung SY (2020) Soft-root-sign activation function. arXiv preprint arXiv:2003.00547
66. Zhu Z, Zhang Y (2021) Flood disaster risk assessment based on random forest algorithm. Neural Comput Appl pp 1–13

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.