Attentive-YOLO: On-Site Water Pipeline Inspection Using Efficient Channel Attention and Reduced ELAN-Based YOLOv7

Atanu Shuvam Roy[a] and Priyanka Bagade[b]
Indian Institute of Technology, Kanpur, India

Keywords: YOLOv7, Computer Vision, Water Pipeline Inspection, Pipe Robot.

Abstract: The effective and dependable distribution of clean water to communities depends on the timely inspection and repair of water pipes. Traditional inspection techniques frequently require expensive physical labour, resulting in false and delayed defect detections. Current water pipeline inspection methods include radiography testing, eddy current testing, and CCTV inspection. These methods require experts to be present on-site to conduct the tests. Radiographed and CCTV images are usually used for pipeline defect detection on-site, yet real-time automatic detection is lacking. Current approaches, including YOLOv5 models with Retinex-based illumination, achieve acceptable performance but hinder fast inference due to bulky models, which is especially concerning for edge devices. This paper proposes an Attentive-YOLO model based on the state-of-the-art object detection YOLOv7 model with a reduced Efficient Layer Aggregation Network (ELAN). We propose a lightweight attention model in the head and backbone of the YOLOv7 network to improve accuracy while reducing model complexity and size. The paper aims to present an efficient model to be deployed on edge devices such as the Raspberry Pi to be used in Internet of Things (IoT) systems and on-site robotics applications like pipeline inspection robots. Based on the experiments, the proposed model, Attentive-YOLO, achieves an *mAP* score of 0.962 over 0.93 (1/3rd channel width) compared to the Yolov7-tiny model, with an almost 20% decrease in model parameters.

1 INTRODUCTION

Metal pipes are an indispensable material in daily life for transporting water from reservoirs to residential and industrial areas. These underground pipes are prone to corrosion and degradation with the passage of time. In recent years, several incidents of gas pipeline leakage and water pipeline bursts have caused drastic disasters (Kalita, 2023). Figure 1 shows some of the most common defects inside underground metal pipes requiring manual labour to detect in most cases.

Destructive methods such as cutting, sectioning, and drilling for pipeline defect detection disrupt the normal services of the pipeline. Hence, non-destructive methods (NDT) such as Visual Testing (VT), Ultrasonic Testing (UT), Thermography and Radiographic Testing (RT) are used to test the pipeline defects. These sensor-based defect detection methods require manual monitoring , e.g. visual inspection via CCTV and sound sensor detection of leaks, pressure testing, and water quality sampling (Korlapati et al., 2022). Thus, for automation, the building of pipeline crawler robots with an attached camera (Ab Rashid et al., 2020), radiography-based detections (Silva et al., 2021), and Eddy Current Based detection (Sharma, 2021) have been employed.

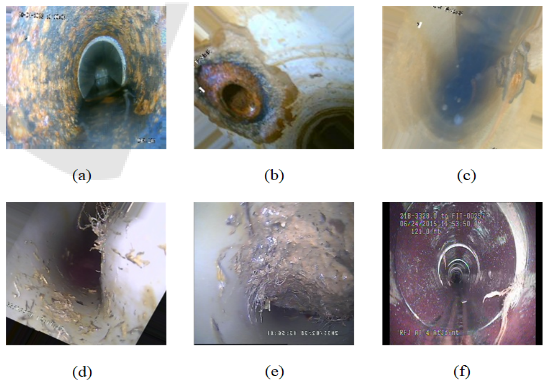

Figure 1: Various types of water pipeline defects. (a) encrustation, (b) ferrule, (c) stone, (d) deformation, (e) sludge formation, (f) roots.

[a] https://orcid.org/0000-0002-5522-9043
[b] https://orcid.org/0000-0003-1045-4148

Roy, A. and Bagade, P.
Attentive-YOLO: On-Site Water Pipeline Inspection Using Efficient Channel Attention and Reduced ELAN-Based YOLOv7.
DOI: 10.5220/0012374000003660

In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 4: VISAPP*, pages 492-499
ISBN: 978-989-758-679-8; ISSN: 2184-4321

Although NDT and automated robots for water pipeline inspection works, they are not automatic. Hence, various approaches employing traditional image processing, Machine Learning (ML), and Deep Learning (DL) techniques such as Convolutional Neural Networks (Yan and Song, 2020), Recurrent Neural Networks (Shaik et al., 2022), ResNet (Guo et al., 2022) etc. have been applied to pipeline inspection tasks like corrosion, joints, holes, encrustations, ferrule, sludge formation, cracks, and penetration of roots. However, drawbacks such as high computational requirements, lack of interpretability causes problems since the main intention remains to be run on single-chip boards for IoT and automation.

Transitioning to object detection models like YOLO (You Only Look Once) and Faster R-CNN addresses limitations in conventional DL methods for pipeline inspection. These models improve defect localization and facilitate targeted maintenance. Object detection enhances adaptability across pipelines and environments by recognizing common defect patterns, reducing the need for domain-specific data. You Only Look Once (YOLO) is a family of object detection models based on Convolutional Neural Networks (CNN). Some of its advantages are faster inference with good mAP score i.e. accuracy required for real-time monitoring and detection(Redmon et al., 2016) for single chip devices. This paper showcases an improved object detection model for the water pipeline defect dataset(solinas xml to txt, 2023) based on the YOLOv7-Tiny variant.

The primary contribution of this paper is to build a lightweight YOLOv7-based model for real-time pipeline defect detection with the following architectural changes:

1. Implemented Efficient Channel Attention Mechanism (ECAM) in the head module for enhanced feature extraction, specifically improving textures in low-light environments through local cross-channel interaction.
2. Introduced Reduced-ELAN (Efficient Layer Aggregation Networks) by removing one standard convolutional layer from each ELAN structure in the original model, reducing overall model complexity.
3. Improved detection phase efficiency by replacing the last three convolution layers with RepConv, demonstrating a favorable accuracy-speed trade-off compared to standard convolution layers.

Extensive experiments performed on a Raspberry Pi 3B+ device take 4% less inference time per frame in full channel width and up to 8% less inference time compared to the original YOLOv7-Tiny model. Our proposed model Attentive-YOLO, achieves an *mAP* score of 0.962 over 0.93 (1/3rd channel width) compared to the Yolov7-tiny model, with an almost 20% decrease in model parameters.

2 RELATED WORK

The pipeline inspection system has used various methods to analyse the pipeline defects (Mangayarkarasi et al., 2019) including image processing, use of ML and DL models to categorise the type of defects, and sensors (acoustic, radiography, camera) to capture data and then analyze that for defect detection.

Specific robots have been developed in recent years using embedded platforms. Mohd Aliff et al (Aliff et al., 2022) created a remotely operated underwater vehicle equipped with the Raspberry Pi and the Raspberry Pi camera to take images of underwater pipes and then use canny edge detection techniques to identify the cracks. Shaikat et al (Shaikat et al., 2021) designed a similar DC motor-powered robot based on the Raspberry Pi and equipped it with an ultrasound sensor, GPS, and a webcam to detect pipeline defects.

2.1 Image Processing Techniques

Digital image processing techniques identify the defective region inside the pipe. Region growing (Yang and Su, 2009), Thresholding (Zhong et al., 2018), and Otsu's binarization (Saranya et al., 2014) are frequently used for segmenting the corrosion and cracks region. Prema et al (Prema Kirubakaran and Murali Krishna, 2018) have used the Kohonen clustering network, Canny Edge detection, and mathematical morphological operator algorithm to develop a system for visually identifying oil pipeline defects. A colour-based technique was created by Venkatasainath et al (Bondada et al., 2018) to identify and measure the corrosion in the pipes. Calculating the mean saturation value across all pixels using the hue saturation and intensity of the surface photographs of the pipes was utilised to identify the corroded area. The morphological procedure is used to quantify the level of corrosion once the corroded area has been identified. This method can only be used for corrosion detection. To find the defects in the CCTV photos of the pipes low image quality and lighting, Yang and Su (Yang and Su, 2009) proposed to use Otsu's segmentation approach to find the defects.

2.2 Machine Learning-Based Techniques

The traditional ML classifiers have been employed in many projects with various feature sets for multiclass pipe defects (Mangayarkarasi et al., 2019),(De Masi et al., 2015). To categorize the defects in sewer pipes, Wei Wu et al (Wu et al., 2015) performed feature extraction and used Adaboost, Random forest and Rotation forest in ensemble learning. To identify defects in CCTV images, Halfway et al. (Halfawy and Hengmeechai, 2014) suggested a classification method based on segmentation, using Otsu's picture segmentation method to extract the Region of Interest (ROI) and used SVM to classify the defects. Duran et al. (Duran et al., 2007) created maps and visualized signal information for future analysis in their paper. A binary classifier first identifies faulty pipe segments and then the defects are classified ass holes, cracks, and foreign obstacles.

2.3 YOLO-Based Techniques

Various works and comparisons have been made using YOLO Models for underwater pipe inspection (Gašparović et al., 2022) (Gašparović et al., 2023). Bastian et al. (Bastian et al., 2019) built a comprehensive pipeline corrosion dataset and achieved 98.8% accuracy in categorizing corrosion levels using a customized convolutional neural network (CNN). Another study by Chen et al. (Chen et al., 2022) proposed an autonomous defect detection system for petrochemical pipeline systems. The approach utilized an enhanced YOLOv5 and Cycle-GAN to address issues of low sample counts and category imbalances. It incorporated a self-attention mechanism and vision transformer, and achieved an mAP of 93.10% for detecting faulty areas on pipes. Situ et al (Situ et al., 2023) have shown a real-time detection system based on YOLOv5 integrating object detection networks, migration learning, and channel pruning techniques. The strategy increased both accuracy and speed. Zhang et al (Zhang et al., 2023) proposed an improved YOLOv4 model where they used spatial pyramid pooling (SPP) to identify sewage defects. According to the experiments, the accuracy of the improved model is 4.6% higher than that of the base YOLOv4 model. Other similar fields, such as vacuum glass tube defect detection (Sheng et al., 2023), and Metal pipe surface defect detection (Nabizadeh and Parghi, 2023) have used YOLOv7 object detection model with attention. Radiography image-based defect detection (Wang et al., 2022) and metal pipe welding defect classification using Mobile Net (Moshayedi et al., 2022) are also proposed in the literature.

However, while these projects demonstrate the potential of using embedded boards like Raspberry Pi to deploy camera-based defect detection systems, there are several limitations to the current approaches. Many of these robots operate in a semi-automatic manner, where video feeds are transmitted for later analysis due to the resource-intensive nature of the models available today. The existing solutions often require substantial computational power for real-time defect detection, rendering them impractical for deployment on low-powered IoT devices.

This paper addresses limitations in current defect detection systems by combining the YOLO (You Only Look Once) object detection framework with the Efficient Channel Attention Mechanism (ECAM). This fusion reduces model complexity and size, making it suitable for low-powered IoT devices like Raspberry Pi. YOLO's real-time capabilities enable prompt defect identification, while ECAM enhances feature representation without significant computational overhead. This optimized synergy allows for efficient real-time defect detection on underwater robots, ensuring timely maintenance and streamlined inspection operations.

3 METHODOLOGY

Utilizing deep learning object detection for identifying pipe defects improves inspection and industrial automation. However, accurately classifying multiple defects with low inference times poses a challenge, especially compared to larger models. This work employs an enhanced YOLOv7-tiny version for object detection, chosen based on application requirements. YOLOv7 is known for its accuracy and real-time detection speed, but there's a trade-off between speed and accuracy (Huang et al., 2017). In our application, minimizing this trade-off is crucial for efficient real-time detection.

3.1 The YOLOv7 Models

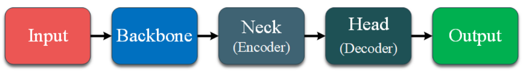

Figure 2: Simplified structure of YOLO algorithm.

YOLO models, including the latest YOLOv7, are single-stage object detectors with a pipeline illustrated in Figure 2. YOLOv7 enhances images using ELAN in the backbone, where extracted features are fused and mixed in the neck before detection in the head. Incorporating focal loss in YOLOv7 improves small object detection by adjusting loss weights. It also processes higher resolution images. YOLOv7 tiny, a lightweight variant, maintains the base model's backbone with ELAN but has fewer convolutional layers, retaining the original architecture without reducing channel width.

3.2 Proposed Model Architecture

This paper suggests a novel pipe defect detection algorithm and uses YOLOv7 Tiny as the base model. This method makes three novel contributions:

- Reducing ELAN structure by having fewer convolution layers before concatenation to reduce complexity and hence inference speed.
- Implementation of ECAM at the end of the backbone layer with an additional convolution layer to compensate for the loss of layers, hence increasing model accuracy.
- The addition of ECAM to the head of the network before the concatenation with the P5 and P4 layers of the backbone also contributes to the accuracy improvement when channel width is reduced.

3.2.1 Reduced ELAN

The backbone of the Attentive YOLO employs reduced ELAN (R-ELAN) where the layers have been reduced from ELAN, maxpooling layers and the PReLU (parametric linear unit) activation function for faster training and reducing inference speed. The network structure of the reduced ELAN is shown in Figure 3.

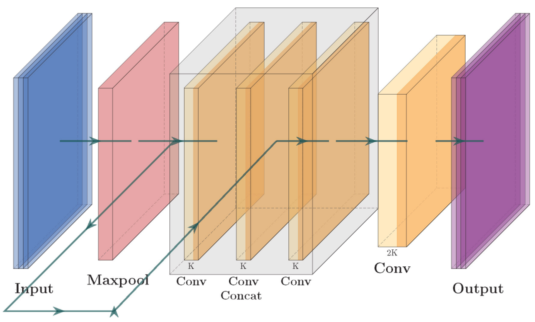

Figure 3: Reduced ELAN module (R-ELAN).

As shown in Figure 3, R-ELAN has three convolutional blocks. Afterwards, layers from the previous are passed through a maxpool layer, concatenated, and then passed through another convolutional block for R-ELAN (Figure 3) and then passed to the next layer shown here as output.

3.2.2 Efficient Channel Attention (ECA) Attention Mechanism

In Efficient Channel Attention, all the channels of the image are passed through the global average pool operation after the image input. Next, channel weights are produced by using a 1x1 convolution of size K_{ac}. After calculating the corresponding probabilities of the channels, it is compared to the original input image (Wang et al., 2020), (Wang et al., 2022).

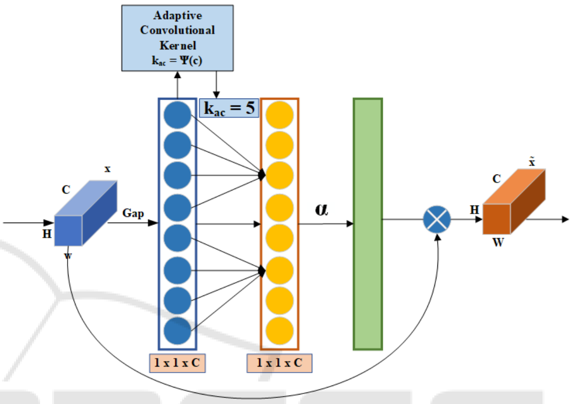

Figure 4: ECA Mechanism Structure.

The input characteristics are multiplied together and then used as the input for the next layer. Equation 1 shows how this method uses function adaptation to find the K_{ac} value, and its proportionality to the dimension of the channel - C:

$$
\begin{aligned}
C &= \phi(K_{ac}) = 2^{(\lambda K_{ac} - b)} \\
K_{ac} &= \psi(C) = \left|\frac{\log_2(C)}{\lambda} + \frac{b}{\lambda}\right|_{odd}
\end{aligned} \tag{1}
$$

where, $\lambda = 2$, $b = 1$ and K_{ac} accepts nearest odd value

Despite being lightweight in design, ECA can eliminate unnecessary data and concentrate on valuable semantic details within feature maps. This is achieved without reducing the dimensions of the data. Figure 4 shows the structure of the ECA Mechanism.

3.2.3 CBS Layers, Loss Function and Activation Function

Each convolutional block shown in the figures has a convolutional layer at the beginning and an activation function at the end, with a batch normalization operation in the middle. The YOLOv7 loss function comprises three components: the bounding box loss function (e.g., CIoU loss Equation 3), the objectness loss function and the classification loss function, both being BCEWithLogitsLoss (BCE-Loss) Equation 2 because of its efficient handling of imbalanced classes.

$$\text{BCE-Loss}(x,t) = -t \cdot \log(\sigma(x)) - (1-t) \cdot \log(1-\sigma(x)) \quad (2)$$

where, x represents the input logits or predicted values, t is the target values or labels and α is the sigmoid function, which maps the logits to probabilities between 0 and 1.

$$\text{CIoU Loss} = 1 - \text{IoU} + \frac{\mathrm{d}^2(\mathrm{c}_1, \mathrm{c}_2)}{\mathrm{r}^2} + \alpha \cdot v \quad (3)$$

where IoU is the Intersection over Union, $d(c_1, c_2)$ represents the distance between the centres of the predicted and ground truth bounding boxes, r^2 represents the squared diagonal length of the smallest enclosing box, α is a hyperparameter that balances the impact of the enclosing box term and finally v represents an additional penalty term that encourages more accurate bounding box regression.

$$\begin{aligned} pReLU(x) &= x \text{ if } x \geq 0 \\ &= \alpha x \text{ else, where } \alpha \text{ need to be trained} \end{aligned} \quad (4)$$

PReLU, as shown in Equation 4, is a variant of the Leaky ReLU function that allows for the alpha parameter to be learned during training.

3.3 Final Model

As shown in Figure 5, we add an ECA block using the ECAM with the convolutional block before and after at the end of the backbone. After every upsample in the neck of the network, an ECAM has been added to capture the features for final detection. Besides, we have several max-pooling layers before each R-ELAN. The initial part of the YOLOv7 tiny model is a simplified version of the SPPCSPC (CSPNet with Spatial Pyramid Pooling Layer) block from the base model. Compared to the original YOLOv7 tiny model, instead of regular convolutional blocks, we have used RepConv in the head of the network. RepConv uses 3x3 convolution, 1x1 convolution, and identity connection in one convolutional layer (Ding et al., 2021) except the identity connection is skipped.

4 EXPERIMENTATION AND RESULTS

For training and initial testing, we used a system running Ubuntu 20.04 LTS equipped with two Nvidia RTX 3090Ti with 24 GB memory GPU. For the runtime environment and framework, python 3.9 and pytorch 2.0.1 are used, respectively. The training parameters were kept identical to the original YOLOv7 model.

4.1 Dataset

It paper uses an open source data set (solinas xml to txt, 2023) consisting of 1872 images gathered from CCTV footage from inside the used water pipes in the sewage system having variable inside diameters from 2.94 inches to 5.94 inches. The dataset contained eight classes, namely root blockage (rb), encrustation (en), ferrule (fr), joint (jt), pipe surface damage (pd), shape deformation (sd), slug accumulation (sa) and stone or obstacles (st) with train/test/val split being 80/10/10.

4.2 Experimental Results

We evaluated several models including YOLOv5 by training them on the dataset mentioned in section 4.1. Two sections are shown in the Table 1. Two approaches were explored: first, maintaining full channel width in each layer and second, adjusting the width by multiplying it with a factor inherent in the YOLOv7 network. The latter option, the lowest feasible multiple was chosen to minimize model size and computation while enhancing accuracy and speed in defect detection.

Table 1: Ablation Experiments Results.

Method	**mAP$_{50}$**	**mAP$_{.50:9.5}$**	**Size (MB)**	**Time (ms)**	**Para (M)**
Previous Generation of YOLO Model					
YOLOv5-l	*0.98*	*0.70*	*92.9*	*7.0*	*46.1*
YOLOv5-m	*0.983*	*0.69*	*42.0*	*4.4*	*20.0*
YOLOv5-s	*0.979*	*0.68*	*13.9*	*3.4*	*7.03*
YOLOv5-n	*0.98*	*0.68*	*3.9*	*3.5*	*1.7*
Full Channel Width [YOLOv7]					
YOLOv7	*0.971*	*0.643*	*74.9*	*5.7*	*37.2*
YOLOv7-Tiny	*0.975*	*0.658*	*12.3*	*3.0*	*6.02*
YOLOv7-Tiny + CBAM	*0.959*	*0.551*	*10.34*	*4.8*	*5.03*
Proposed Method	*0.973*	***0.634***	***10.26***	***2.3***	***4.9***
1/3rd Model Depth + 1/4th Channel Width [YOLOv7]					
YOLOv7	*0.951*	*0.628*	*5.1*	*3.5*	*2.3*
YOLOv7-Tiny	*0.93*	*0.58*	*1.0*	*2.8*	*0.38*
YOLOv7-Tiny + CBAM	*0.956*	*0.58*	*0.88*	*4.3*	*0.32*
Proposed Method	***0.962***	***0.572***	***0.87***	***2.2***	***0.31***

As shown in Table 1, the original YOLOv7 performs well but demands powerful hardware due to its larger size. The larger YOLOv7 and Yolov5 do not run efficiently on the Raspberry Pi system. YOLOv7 achieves 0.951 mAP_{50} with reduced depth and 5.1 MB size, with an inference time of 3.5 ms. YOLOv7-tiny gets 0.93 mAP_{50} and 1.0 MB size, but with less inference speed. With CBAM, YOLOv7 tiny reaches 0.956 mAP_{50} but slower inference, unsuitable for the Raspberry Pi. Our proposed model, Attentive YOLO

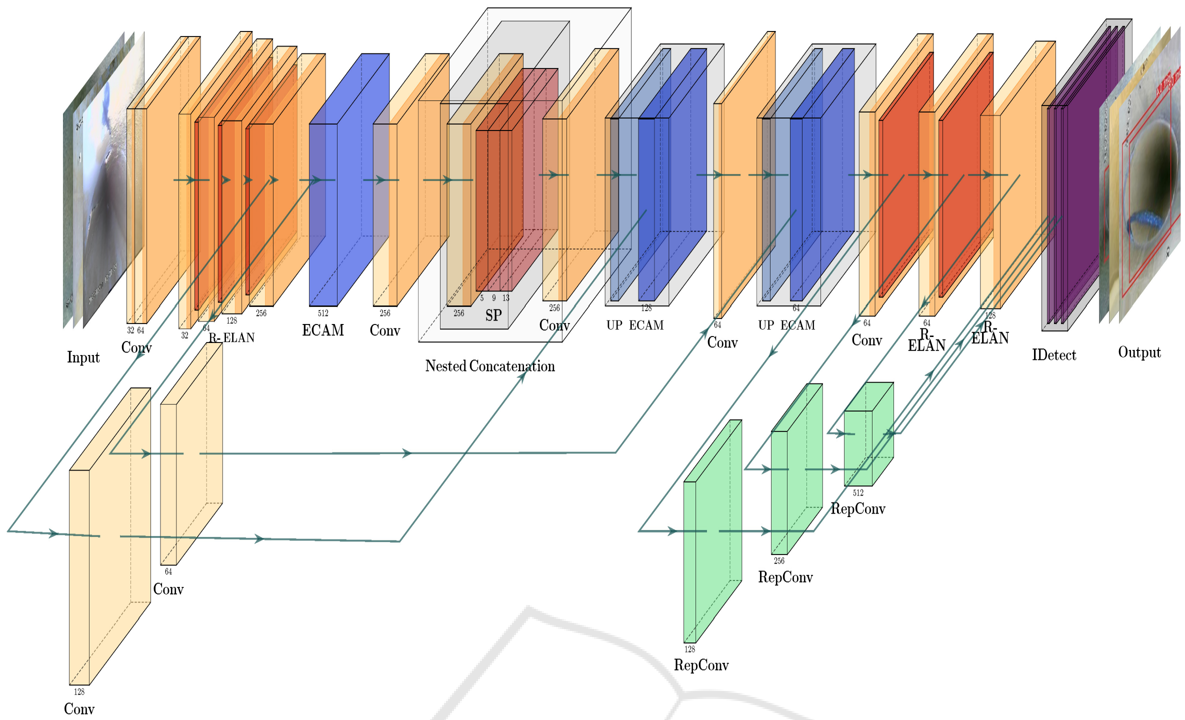

Figure 5: Architecture of the proposed Attentive YOLO.

(YOLOv7-Tiny + ECAM) gets 0.962 mAP_{50}, similar size, yet significantly faster inference than the base models, i.e. a favorable speed-accuracy tradeoff.

In the Table 2, the accuracies of different classes of the dataset are shown. Compared to the original models, our proposed model, Attentive YOLO performs consistently in both full-channel and reduced-channel mode.

Table 2: Model Performance on water pipeline image dataset (solinas xml to txt, 2023).

Model	rb	en	fr	jt	pd	sd	sa	st
Previous Generation of YOLO Model								
YOLOv5-l	0.995	0.929	0.966	0.995	0.995	0.995	0.995	0.977
YOLOv5-m	0.995	0.929	0.965	0.995	0.995	0.995	0.995	0.993
YOLOv5-s	0.995	0.913	0.949	0.995	0.995	0.995	0.995	0.991
YOLOv5-n	0.995	0.928	0.965	0.995	0.995	0.995	0.995	0.99
Full Channel Width [YOLOv7]								
YOLOv7	0.996	0.918	0.972	0.979	0.995	0.996	0.996	0.975
YOLOv7-Tiny	0.996	0.908	0.972	0.917	0.995	0.996	0.996	0.98
YOLOv7-Tiny+CBAM	0.996	0.912	0.975	0.996	0.996	0.997	0.996	0.991
Proposed Method	0.995	0.897	0.956	**0.995**	0.995	0.996	0.996	**0.992**
Reduced Model Depth and Channel Width [YOLOv7]								
YOLOv7	0.995	0.892	0.971	0.786	0.995	0.995	0.995	0.981
YOLOv7-Tiny	0.933	0.821	0.93	0.928	0.904	0.995	0.992	0.954
YOLOv7-Tiny+CBAM	0.995	0.876	0.902	0.995	0.902	0.995	0.995	0.985
Proposed Method	**0.995**	**0.875**	0.927	**0.976**	**0.956**	0.995	0.995	**0.975**

As the table shows, as the model depth is reduced compared to the original model (YOLOv7-Tiny), accuracy has reduced significantly. Adding CBAM certainly increases the accuracy; however, the performance, as shown in Table 3 deployed on a Raspberry Pi Model 3B+ is reduced. With Attentive YOLO, accuracy is nearly consistent with the full-depth models, keeping performance the same too.

Table 3: Performance of Model on the Raspberry Pi.

Model	Min FPS	AVG FPS	Max FPS	Inference Time (ms)
Previous Generation of YOLO Model				
YOLOv5-l	NULL	NULL	NULL	NULL
YOLOv5-m	0.02	0.04	0.05	20307
YOLOv5-s	0.12	0.129	0.134	7773
YOLOv5-n	0.22	0.29	0.35	3481
Full Channel Width [YOLOv7]				
YOLOv7	NULL	NULL	NULL	NULL
YOLOv7-Tiny	0.16	0.23	0.30	4259
YOLOv7-Tiny-CBAM	0.19	0.28	0.33	3493
Proposed Method	**0.22**	0.28	**0.35**	**3385**
Reduced Model Depth and Channel Width [YOLOv7]				
YOLOv7	0.19	0.26	0.35	3716
YOLOv7-Tiny	0.53	0.87	1.36	946
YOLOv7-Tiny-CBAM	0.40	0.89	1.39	902
Proposed Method	**0.53**	**0.95**	1.18	**878**

As it shows in Table 3, the proposed Attentive YOLO performs efficiently w.r.t. inference speed and FPS, considering it runs on a single-chip computer Raspberry-Pi system. The initial YOLOv7 model encounters runtime issues, causing device crashes, while the YOLOv7-Tiny model exhibits sluggish performance with inference times nearing 4 seconds per frame. In contrast, our proposed model, Attentive YOLO with R-ELAN, demonstrates substantial enhancements tailored to our use case, achieving approximately 3.3 seconds per frame for the full-depth model. This translates to nearly 1 frame per second (FPS) or an efficient 0.9 milliseconds of inference time on average, surpassing the performance of CBAM with reduced width in the model.

Figure 6A shows the simulated setup of the device with a USB webcam connected to Raspberry-Pi system. Figure 6B shows the prediction results after being run. For the real-world simulation, the video of the pipe was directly fed to the Raspberry Pi as a file and via webcam.

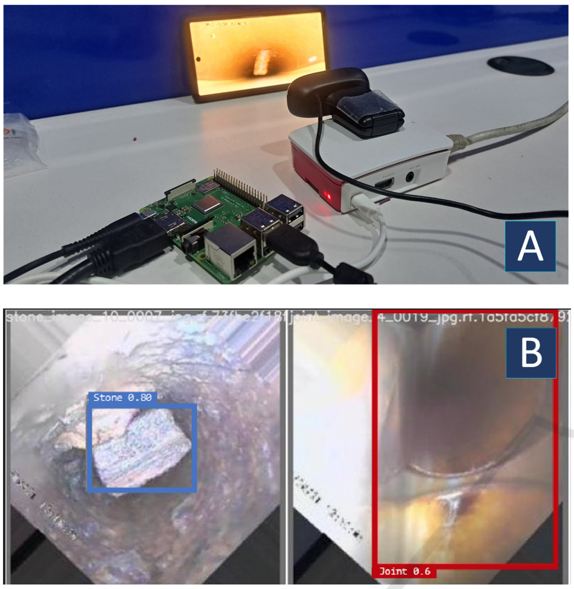

Figure 6: Attentive-YOLO Experimental Setup & Results.

The test images have been augmented (rotated, cropped, flipped etc.) in several ways to challenge the model. Inference times for the webcam and direct video feed are averaged and shown in Table 3.

5 CONCLUSION AND FUTURE WORK

In this paper, we address the challenge of inspecting and repairing water pipes by leveraging computer vision techniques for non-destructive testing by proposing a modified version of the YOLOv7 Tiny model, incorporating the R-ELAN with ECAM into the base architecture. This efficient model is suitable for deployment on small-scale computing devices like the Raspberry Pi in IoT and robotics applications like pipeline inspection robots. The performance evaluation demonstrates that our proposed model, Attentive YOLO, outperforms the base YOLOv7 and YOLOv7-Tiny models on a single-chip computer regarding inference speed and Frames Per Second. The base YOLOv7 model fails to run on the device, leading to system crashes, while the YOLOv7-Tiny model exhibits slow inference times, taking nearly 4 seconds per frame. In contrast, the attentive YOLO achieves an average inference time of approximately 0.9 seconds per frame on the Raspberry Pi for the full-depth model, corresponding to nearly 1 FPS on average on the shallow model while retaining considerable accuracy compared to state-of-the-art models. Future work can focus on refining the model further, exploring additional optimizations, and evaluating its performance in real-world pipeline inspection scenarios to ensure its practical applicability, scalability and implementation in a pipe inspection robot.

REFERENCES

Ab Rashid, M. Z., Yakub, M. F. M., bin Shaikh Salim, S. A. Z., Mamat, N., Putra, S. M. S. M., and Roslan, S. A. (2020). Modeling of the in-pipe inspection robot: A comprehensive review. *Ocean Engineering*, 203:107206.

Aliff, M., Hanisah, N. F., Ashroff, M. S., Hassan, S., Nurr, S. F., and Sani, N. S. (2022). Development of underwater pipe crack detection system for low-cost underwater vehicle using raspberry pi and canny edge detection method. *International Journal of Advanced Computer Science and Applications*, 13(11).

Bastian, B. T., N, J., Ranjith, S. K., and Jiji, C. (2019). Visual inspection and characterization of external corrosion in pipelines using deep neural network. *NDT & E International*, 107:102134.

Bondada, V., Pratihar, D. K., and Kumar, C. S. (2018). Detection and quantitative assessment of corrosion on pipelines through image analysis. *Procedia Computer Science*, 133:804–811.

Chen, K., Li, H., Li, C., Zhao, X., Wu, S., Duan, Y., and Wang, J. (2022). An automatic defect detection system for petrochemical pipeline based on cycle-gan and yolo v5. *Sensors*, 22(20):7907.

De Masi, G., Gentile, M., Vichi, R., Bruschi, R., and Gabetta, G. (2015). Machine learning approach to corrosion assessment in subsea pipelines. In *OCEANS 2015-Genova*, pages 1–6. IEEE.

Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., and Sun, J. (2021). Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13733–13742.

Duran, O., Althoefer, K., and Seneviratne, L. D. (2007). Automated pipe defect detection and categorization using camera/laser-based profiler and artificial neural network. *IEEE Transactions on Automation Science and Engineering*, 4(1):118–126.

Gašparović, B., Lerga, J., Mauša, G., and Ivašić-Kos, M. (2022). Deep learning approach for objects detection in underwater pipeline images. *Applied Artificial Intelligence*, 36(1):2146853.

Gašparović, B., Mauša, G., Rukavina, J., and Lerga, J. (2023). Evaluating yolov5, yolov6, yolov7, and yolov8 in underwater environment: Is there real improvement? In *2023 8th International Conference on*

Smart and Sustainable Technologies (SpliTech), pages 1–4. IEEE.

Guo, W., Zhang, X., Zhang, D., Chen, Z., Zhou, B., Huang, D., and Li, Q. (2022). Detection and classification of pipe defects based on pipe-extended feature pyramid network. *Automation in Construction*, 141:104399.

Halfawy, M. R. and Hengmeechai, J. (2014). Automated defect detection in sewer closed circuit television images using histograms of oriented gradients and support vector machine. *Automation in Construction*, 38:1–13.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311.

Kalita, N. (2023). Assam: Water pipeline connecting Kharguli reservoir explodes, several injured.

Korlapati, N. V. S., Khan, F., Noor, Q., Mirza, S., and Vaddiraju, S. (2022). Review and analysis of pipeline leak detection methods. *Journal of Pipeline Science and Engineering*, page 100074.

Mangayarkarasi, N., Raghuraman, G., and Kavitha, S. (2019). Influence of computer vision and iot for pipeline inspection-a review. In *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, pages 1–6.

Moshayedi, A. J., Khan, A. S., Yang, S., and Zanjani, S. M. (2022). Personal image classifier based handy pipe defect recognizer (hpd): Design and test. In *2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)*, pages 1721–1728. IEEE.

Nabizadeh, E. and Parghi, A. (2023). Automated corrosion detection using deep learning and computer vision. *Asian Journal of Civil Engineering*, pages 1–13.

Prema Kirubakaran, A. and Murali Krishna, I. (2018). Pipeline crack detection using mathematical morphological operator. *Knowledge Computing and its Applications: Knowledge Computing in Specific Domains: Volume II*, pages 29–46.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.

Saranya, R., Daniel, J., Abudhahir, A., and Chermakani, N. (2014). Comparison of segmentation techniques for detection of defects in non-destructive testing images. In *2014 International Conference on Electronics and Communication Systems (ICECS)*, pages 1–6. IEEE.

Shaik, N. B., Benjapolakul, W., Pedapati, S. R., Bingi, K., Le, N. T., Asdornwised, W., and Chaitusaney, S. (2022). Recurrent neural network-based model for estimating the life condition of a dry gas pipeline. *Process Safety and Environmental Protection*, 164:639–650.

Shaikat, A. S., Hussein, M. R., and Tasnim, R. (2021). Design and development of a pipeline inspection robot for visual inspection and fault detection. In *Proceedings of Research and Applications in Artificial Intelligence: RAAI 2020*, pages 243–253. Springer.

Sharma, R. R. (2021). Gas leakage detection in pipeline by svm classifier with automatic eddy current based defect recognition method. *Journal of Ubiquitous Computing and Communication Technologies (UCCT)*, 3(03):196–212.

Sheng, Z., Chen, H., and Qi, Z. (2023). Cbam-based method in yolov7 for detecting defective vacuum glass tubes. In *Proceedings of the 2023 2nd Asia Conference on Algorithms, Computing and Machine Learning*, pages 413–418.

Silva, W., Lopes, R., Zscherpel, U., Meinel, D., and Ewert, U. (2021). X-ray imaging techniques for inspection of composite pipelines. *Micron*, 145:103033.

Situ, Z., Teng, S., Liao, X., Chen, G., and Zhou, Q. (2023). Real-time sewer defect detection based on yolo network, transfer learning, and channel pruning algorithm. *Journal of Civil Structural Health Monitoring*, pages 1–17.

solinas xml to txt (2023). kapilproject dataset. https://universe.roboflow.com/solinas-xml-to-txt/kapilproject. visited on 2023-06-17.

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., and Hu, Q. (2020). Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11534–11542.

Wang, Y., Wang, H., and Xin, Z. (2022). Efficient detection model of steel strip surface defects based on yolo-v7. *IEEE Access*, 10:133936–133944.

Wu, W., Liu, Z., and He, Y. (2015). Classification of defects with ensemble methods in the automated visual inspection of sewer pipes. *Pattern Analysis and Applications*, 18:263–276.

Yan, X. and Song, X. (2020). An image recognition algorithm for defect detection of underground pipelines based on convolutional neural network. *Traitement du Signal*, 37(1).

Yang, M.-D. and Su, T.-C. (2009). Segmenting ideal morphologies of sewer pipe defects on cctv images for automated diagnosis. *Expert Systems with Applications*, 36(2):3562–3573.

Zhang, J., Liu, X., Zhang, X., Xi, Z., and Wang, S. (2023). Automatic detection method of sewer pipe defects using deep learning techniques. *Applied Sciences*, 13(7):4589.

Zhong, X., Peng, X., Yan, S., Shen, M., and Zhai, Y. (2018). Assessment of the feasibility of detecting concrete cracks in images acquired by unmanned aerial vehicles. *Automation in Construction*, 89:49–57.