Efficient Binary Segmentation Through Dense Neural Networks in a Truncated Frequency Domain

Nils Defauw
Univ. Grenoble Alpes, CEA, List
F-38000 Grenoble, France
nils.defauw_pro@posteo.net

Marielle Malfante
Univ. Grenoble Alpes, CEA, List
F-38000 Grenoble, France
marielle.malfante@cea.fr

Olivier Antoni
Univ. Grenoble Alpes, CEA, List
F-38000 Grenoble, France
olivier.antoni@cea.fr

Tiana Rakotovao
Univ. Grenoble Alpes, CEA, List
F-38000 Grenoble, France
tiana.rakotovao@cea.fr

Suzanne Lesecq
Univ. Grenoble Alpes, CEA, Leti
F-38000 Grenoble, France
suzanne.lesecq@cea.fr

Abstract—This article presents a method for binary segmentation of any type of tensor given that a dataset of such tensors with ground truth segmentations is available. The proposed method compresses input tensors through the use of the Discrete Cosine Transform (DCT) followed by a truncation of the resulting spectrums. After the compression step, a shallow dense neural network performs the segmentation entirely in the frequency domain. The method is evaluated on a common robotics environment model known as an occupancy grid map. Results exhibit a correct segmentation for an especially small computational time of 2.16 ms for the largest neural network. Moreover, the computational requirements are freely configurable by the choice of the compression factor making such a method interesting for highly constrained hardware platforms found in embedded setups.

Index Terms—segmentation, compression, frequency, DCT, neural networks

I. Introduction

Segmentation is an important perception task for robotics applications allowing the labelling of data points into classes corresponding to the objects present in the scene. Segmentation can be performed on a wide range of data such as images (where data points are pixels) or point clouds (where data points are individual points contained in the point cloud).

Robots feature two major constraints applying to segmentation algorithms: real-time execution and execution within the limited computational power of embedded hardware. Hence, perception algorithms used such as segmentation algorithms must achieve a balance between the quality of the results produced and the computational resources needed for their execution. Most current state-of-the-art segmentation algorithms rely on deep CNNs [1] (Convolutional Neural Networks) requiring the use of parallel computing architectures to be computed in real-time (e.g. Graphical Processing Units, Artificial Intelligence accelerators or specialized circuitry mimicking the neural network architecture).

This article presents an alternative approach for binary segmentation targeting low-power embedded processing units without special circuitry except for a DSP (Digital Signal Processor). The method is composed of two steps:

1) A configurable reduction of the dimension of the input through the truncation of its spectrum obtained by application of the DCT (Discrete Cosine Transform). The obtained truncated spectrum is a lossy compressed form of the original input.
2) A prediction of the truncated spectrum of a segmentation of the input through the use of a dense neural network composed of two or three layers only. The predicted spectrum is afterwards decompressed into the original domain of the input and thresholded into positive and negative classes interpreted as a segmentation of the input.

The choice of the dimension of the truncated spectrum computed from the input heavily influences the size of the subsequent dense neural network which leads to a fully configurable computational cost of the method that could be tailored specifically for individual embedded processing units.

Although the proposed method could be applied to any tensor-shaped input, it is evaluated on an environment model commonly used for robotics applications known as an occupancy grid map [2]. An occupancy grid map is a bird's eye view matrix representing the environment surrounding a robot. Each cell of an occupancy grid map represents a portion of the environment that can be occupied by an obstacle or not. To each cell is affected a probability ranging from 0 to 1 that estimates its occupancy state. Occupancy grid maps are commonly used for their ability to fuse different sensor measurements and for their low computational cost [3].

Both the quality of the predicted segmentations and the computational costs are evaluated. Used compression ratios ranging from 1/1024 to 1/32 in the method provide results ranging from 0.272 AP (Average Precision score on each component of the tensor) metric to 0.574 AP

with inference times ranging from 0.76ms to 2.16ms on CPU (Central Processing Unit) per occupancy grid map and neural network sizes ranging from 8,320 to 12,589,056 trainable weights. Moreover, these results are to our knowledge the first to present a segmentation method based on deep learning in a compressed frequency domain.

II. Background

A. Segmentation Neural Networks

Most recent segmentation algorithms rely on CNNs using as input a tensor of data and producing as output a similarly-shaped tensor containing the predicted segmentation. Segmentation networks exist for images (i.e. input tensor is the image [4], [5]), point clouds (i.e. input tensor is a voxelized form of the point cloud [6] or a spherical projection of it [7]).

Previous works propose to segment occupancy grid maps by using an image segmentation network on an RGB image which is then fused with an occupancy grid map with the help of a fusion network [8].

B. Deep Learning In The Frequency Domain

The frequency domain is mainly known for its use for lossy compression of signals such as images. In particular, the JPEG [9] (Joint Photographic Experts Group) format compresses images through the computation of the spectrum of the image with the DCT transform and the quantization of the obtained coefficients.

Transformations to frequency domain are also used with deep learning for some tasks:

- References [10], [11] propose to train CNNs on DFT-obtained (Discrete Fourier Transform) spectrums of bridge satellite images to detect cracks in the concrete ;
- Reference [12] propose multimodal neural networks using above else image spectrums to perform image translation ;
- Reference [13] propose multimodal neural networks to detect software image manipulations.

To the best of our knowledge, all neural networks using some form of frequency domain input do not predict frequency domain output. At the opposite, the method presented in this article is entirely performed in the frequency domain, from the input of the neural network to its output. We postulate that this distinctive characteristic is key to the especially low computation times observed.

III. Efficient Binary Segmentation Through Dense Neural Networks in a Truncated Frequency Domain

In the following, the input on which segmentation should be performed is a tensor indexed by d indices: $(x_{i^1,\cdots,i^d})_{1\leq i^1\leq n^1,\cdots,1\leq i^d\leq n^d}$. As a result, the tensor considered has $n^1 * \cdots * n^d$ elements. The method assumes that a dataset of input tensors with associated ground truth segmentations is available. This dataset is further divided into training, validation and testing splits.

A. Compression In The Frequency Domain Through Truncation Of Spectrums

To lower the computational costs compared to regular conventional CNNs, we propose to reduce input dimension through a compression inspired by compression formats such as JPEG. The first step is the computation of the spectrum of the input through the use of the DCT (Dicrete Cosine Transform). The obtained spectrum has the same dimension as the input tensor.

To reduce the initial dimension, the second step consists in selecting a subset of coefficients among all the coefficients of the spectrum. This action is performed through a truncation of the spectrum according to a mask.

In order to keep the most informative frequencies, we propose a method to create a truncation mask tailored specifically for the type of input tensor at hand. The truncation mask is created through the following steps:

1) Each input tensor $(x_{i^1,\cdots,i^d})_{i^1,\cdots,i^d}$ of the training split of the dataset is transformed into its spectrum $(y_{i^1,\cdots,i^d})_{i^1,\cdots,i^d}$ through application of the DCT.
2) Each spectrum is then normalized to obtain a new tensor $(\hat{y}_{i^1,\cdots,i^d})_{i^1,\cdots,i^d}$ such that:
$$\hat{y}_{k^1,\cdots,k^d} = \frac{|y_{k^1,\cdots,k^d}|}{\sum_{i^1=1}^{n^1}\cdots\sum_{i^d=1}^{n^d}|y_{i^1,\cdots,i^d}|}$$
This operation ensures that each element of the new tensor is in the range $[0,1]$ and that the sum of all its elements is equal to 1.
3) All of these normalized spectrums are then averaged into a new tensor $(\tilde{y}_{i^1,\cdots,i^d})_{i^1,\cdots,i^d}$.
4) Finally, the truncating mask retaining c frequencies is defined as the tensor M^c with c elements equal to 1 and all other elements equal to 0 for which the indices of the elements equal to 1 match the indices of the c largest values in the averaged tensor $(\tilde{y}_{i^1,\cdots,i^d})_{i^1,\cdots,i^d}$.

Finally, compressing an input tensor is done by masking the spectrum of the input tensor by using the truncating mask. This method allows keeping only c coefficients which can then be flattened into a vector of size c.

Decompression with some loss is done by projecting back the flattened vector into a d dimensional tensor by using the same mask M^c used for compression. The spectrum tensor is then used to reconstruct an input tensor with the inverse DCT.

B. Compressed Form Prediction Of Segmentation

In this subsection, we present a procedure allowing to segment an input tensor using these compression/decompression methods.

We propose to use a dense neural network to segment the input tensor. The procedure starts from the compression of the input tensor into a vector of spectral coefficients of a chosen dimension c as explained in the previous subsection. A dense neural network uses this

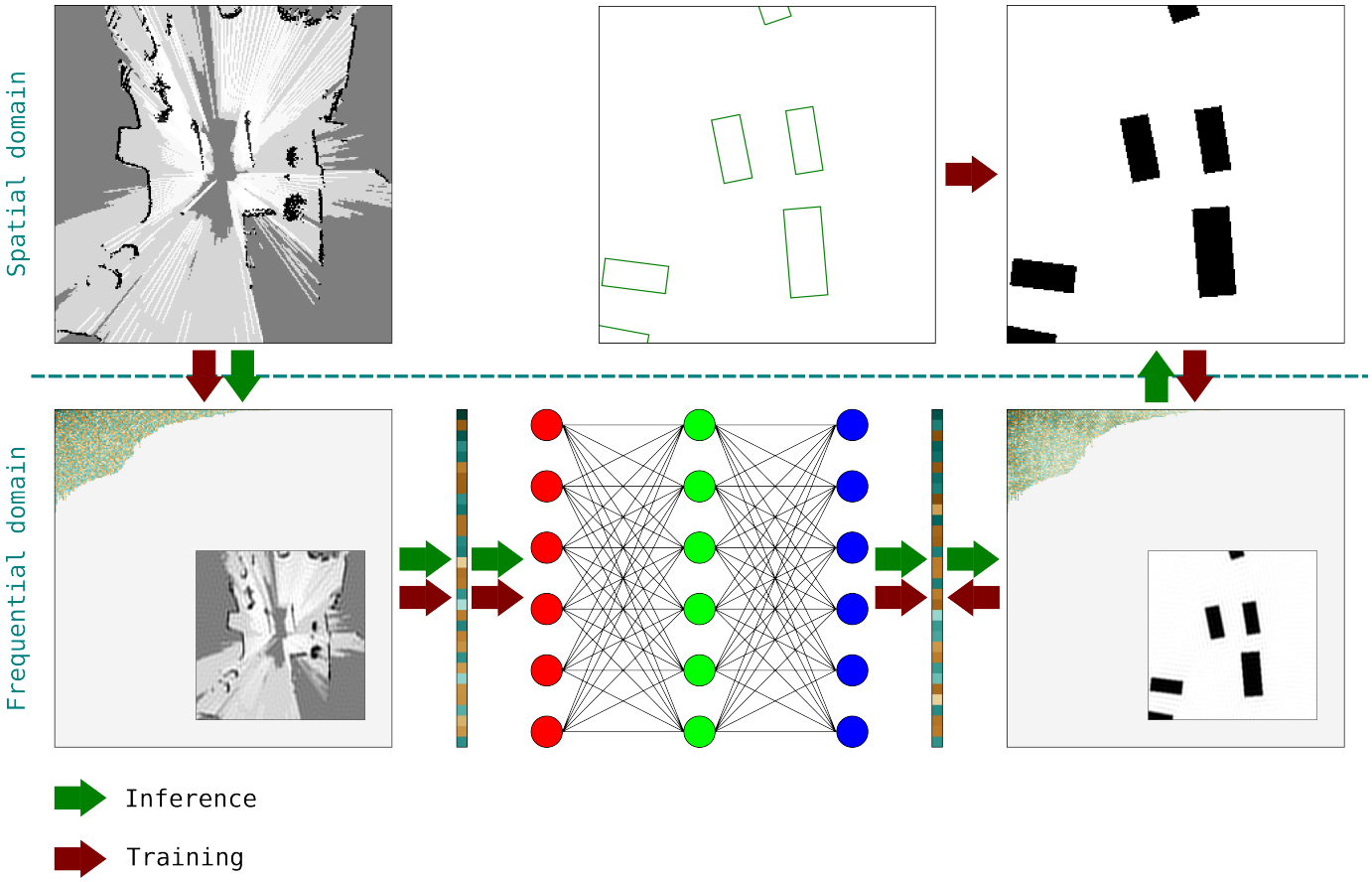

Fig. 1. A depiction of the entire binary segmentation process during the training and the inference phases.

compressed vector to predict a new vector of the same dimension c. This vector is then decompressed into a tensor of the same shape as the input tensor which can be later thresholded to obtain positive and negative classes expected to correspond to a binary segmentation of the input tensor.

The neural network is feedforward and is composed of two or three dense layers. The choice of the number of layers is detailed in the next section. All layers except the last one have a ReLU (Rectified Linear Unit) activation function. The last layer has no activation function (also designed as a linear activation function). All layers have c inputs and c outputs which leads to $c * (c + 1)$ weights per layer. As a result, the choice of the dimension of compression c has a major impact on the size of the neural network expressed in number of trainable weights.

In order to train the neural network to produce the expected output, the training split of the dataset is used. Each example of this dataset is a tuple composed of an input tensor and the ground truth binary segmentation associated. Using the compression procedure, each of these examples is transformed into the compressed form of the input tensor associated with the compressed form of the ground truth binary segmentation. These compressed examples are used as the input and the ground truth output of the neural network. The loss used to compare predicted and ground truth vectors is the Mean Squared Error and the optimizer algorithm used to update the neural network weights accordingly is Adam [14]. Finally, neural networks are trained with an early stopping procedure for which the loss is computed on the validation split after each training epoch and the training is stopped once the validation loss starts to raise.

IV. Experimental Results

The previous section has presented the generic procedure for arbitrary shaped tensors. This procedure has been evaluated on specific input tensors known as occupancy grid maps. The task evaluated is the segmentation of vehicles appearing on occupancy grid maps.

The occupancy grid maps dataset used in the present work has been created from the Waymo Open [15] automotive dataset containing 20 seconds long sequences of driving with 10Hz measurements from five LiDARs. These LiDAR measurements are transformed into occupancy grid maps composed of 256×256 cells where each cell represents a square of 10 cm $\times$ 10 cm within the environment. Each cell contains an occupancy probability ranging from 0 to 1. The Waymo Open dataset is transformed into 198,068 occupancy grid maps associated with bounding

boxes for vehicles which will be used to create ground truth segmentations. 80% of the dataset composes the training split whereas the validation and testing splits represent each 10% of the dataset. Such an occupancy grid map can be seen in the top-left corner of Fig. 1.

Truncation masks are created from the training set. A visual depiction of these masks is visible in Fig. 2. Remarkably, these masks keep higher frequencies on the horizontal scale than on the vertical scale. This behavior is believed to be caused by particularities in the occupancy grid maps used in which most edges correspond to vehicles parallel to the ego-vehicle and are thus vertical.

Fig. 2. Truncation masks obtained for occupancy grid maps.

Compared to fixed masks such as square masks keeping c elements defined as $a_{i^1,i^2} = 1 \leftrightarrow i^1 \wedge i^2 \leq \sqrt{c}$, the proposed tailored truncation masks exhibit better reconstruction with the inverse DCT when evaluated with MSE (Mean Squared Error). Table I shows this advantage for occupancy grid maps.

TABLE I
Reconstruction loss as the MSE between reconstructed occupancy grid maps and original ones with proposed masks compared to square masks.

Masks	c=64	c=256	c=1024	c=4096	c=16384
Square	0.0233	0.0182	0.0142	0.0101	0.0054
Proposed	0.0217	0.0175	0.0136	0.0096	0.0051

Regarding the segmentation itself performed in the frequency domain by the dense neural network, Fig. 3 presents results obtained on an occupancy grid map for different choices of the dimension of compression c. Notice how raising the dimension of compression c improves the result as the information provided to the neural network about the original occupancy grid map is more complete (second column of Fig. 3). The number of layers used in the neural networks is chosen in this figure to be optimal for each dimension chosen c.

Quantitatively speaking, Table II shows metrics representing the quality of segmentations with the computational costs of the entire process expressed as inference times and number of weights of the neural network.

Metrics show especially efficient methods with an inference time (which includes the compression and decompression steps) less than one millisecond for a dimension of compression c less than or equal to 1024. In terms of quality of the predicted segmentations, better results are achieved with an increase of the dimension c at the expense of the inference times and number of trainable weights.

TABLE II
Average Precision scores, computational times per occupancy grid map and number of trainable weights for the different neural networks trained with different dimensions c.

Parameters		Quality	Computational costs	
c	# layers	AP	Time	# weights
64	2	0.272	0.76ms	8,320
64	3	0.280	0.77ms	12,480
128	2	0.337	0.84ms	33,024
128	3	0.405	0.91ms	49,536
256	2	0.456	0.83ms	131,584
256	3	0.484	0.88ms	197,376
512	2	0.531	0.85ms	525,312
512	3	0.530	0.92ms	787,968
1024	2	0.561	0.90ms	2,099,200
1024	3	0.536	1.04ms	3,148,800
2048	2	0.574	1.62ms	8,392,704
2048	3	0.527	2.16ms	12,589,056

The computed Average Precision scores range from 0.272 for a dimension $c = 64$ and a two-layer network to 0.574 for a dimension $c = 2048$ and a two-layer network.

V. Prospects and Conclusion

The present article proposes a new method for segmentation which is performed by a shallow dense neural network operating entirely in a truncated (compressed) frequency domain. This method has been evaluated on the task of binary classification of vehicles on occupancy grid maps. Results exhibit the ability of the method to segment vehicles with the quality of segmentations as well as the computational costs increasing with the dimensionality of the compressed form c. Still, small dimension of compression such as $c = 512$ exhibits good segmentations for an especially small inference time less than 1ms.

We believe that this method paves the way to the use of deep learning on compressed data for embedded setups. This claim is supported by our inability to find other deep learning methods outputting in a compressed frequency domain in the scientific literature. Moreover, although evaluations have only been made with occupancy grid maps, the proposed method could be used with any tensor form input and should be evaluated on other types of data.

References

[1] B. Emek Soylu, M. S. Guzel, G. E. Bostanci, F. Ekinci, T. Asuroglu, and K. Acici, "Deep-learning-based approaches for semantic segmentation of natural scene images: A review," Electronics, vol. 12, no. 12, p. 2730, 2023.

[2] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," Computer, vol. 22, no. 6, pp. 46–57, Jun. 1989.

[3] T. Rakotovao, "Integer Occupancy Grids : a probabilistic multi-sensor fusion framework for embedded perception," phdthesis, Université Grenoble Alpes, Feb. 2017.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, ser. Lecture Notes in Computer Science, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

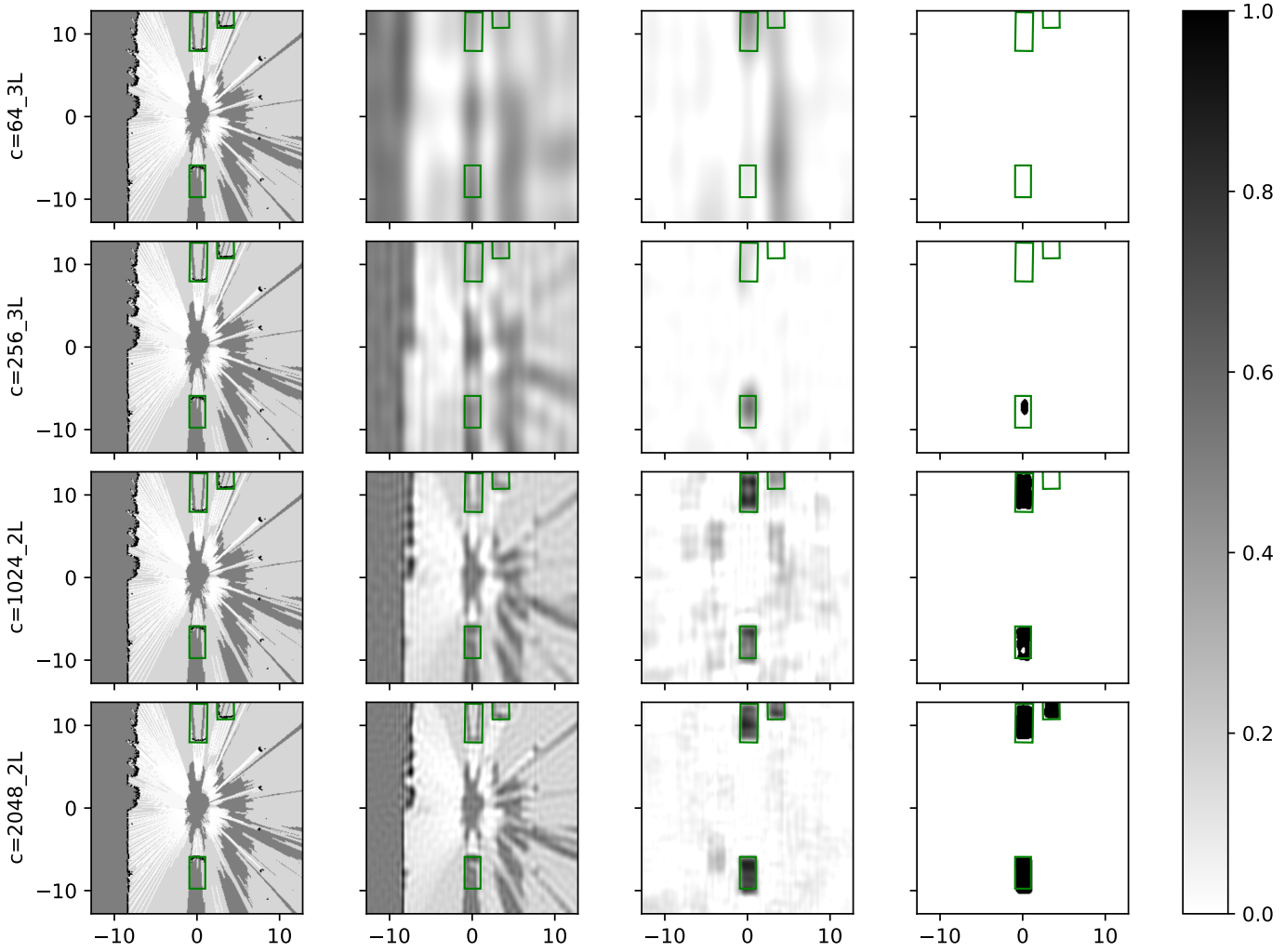

Fig. 3. Segmentations obtained with different choices of compression of dimension c. Columns, left to right: input occupancy grid map; reconstruction of an occupancy grid map from compressed form; reconstructed segmentation from the output of the neural network; binarized segmentation obtained from the third column.

[5] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 1, pp. 263–272, Jan. 2018.

[6] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sep. 2015, pp. 922–928.

[7] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nov. 2019, pp. 4213–4220.

[8] O. Erkent, C. Wolf, and C. Laugier, "Semantic Grid Estimation with Occupancy Grids and Semantic Segmentation Networks," in 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Nov. 2018, pp. 1051–1056.

[9] ISO, ISO/IEC 10918-1:1994: Information technology - Digital compression and coding of continuous-tone still images: Requirements and guidelines. Geneva, Switzerland: International Organization for Standardization, 1994.

[10] Q. Zhang, K. Barri, S. K. Babanajad, and A. H. Alavi, "Real-Time Detection of Cracks on Concrete Bridge Decks Using Deep Learning in the Frequency Domain," Engineering, vol. 7, no. 12, pp. 1786–1796, Dec. 2021.

[11] G. Kolappan Geetha and S.-H. Sim, "Fast identification of concrete cracks using 1D deep learning and explainable artificial intelligence-based analysis," Automation in Construction, vol. 143, p. 104572, Nov. 2022.

[12] M. Cai, H. Zhang, H. Huang, Q. Geng, Y. Li, and G. Huang, "Frequency Domain Image Translation: More Photo-realistic, Better Identity-preserving," in 2021 IEEE/CVF International Conference on Computer Vision (ICCV). Montreal, QC, Canada: IEEE, Oct. 2021, pp. 13 910–13 920.

[13] S. Li, S. Xu, W. Ma, and Q. Zong, "Image Manipulation Localization Using Attentional Cross-Domain CNN Features," IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 9, pp. 5614–5628, Sep. 2023.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Jan. 2017.

[15] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in Perception for Autonomous Driving: Waymo Open Dataset," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA: IEEE, Jun. 2020, pp. 2443–2451.