Two-dimensional DFT with sliding and hopping windows for edge map generation of road images

Amit Kr. Vishwakarma | N. Sukumar | P. Sumathi

Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand 247 667, India

Correspondence
P. Sumathi, Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand 247 667, India.
Email: psumifee@iitr.ac.in

Abstract
An edge map generation technique based on two-dimensional discrete Fourier transform with sliding and hopping windows is proposed for road images to detect the lanes and road markings. A $2 \times 2$ sliding/hopping window DFT with bin indices $(k_1, k_2) = (0, 1)$ for horizontal edge detection and $(k_1, k_2) = (1, 0)$ for vertical edge detection has been proposed. The 2-D SDFT/HDFT-based edge detector has been proved to be more efficient for lane and road marking images in comparison with conventional edge detectors and cellular neural network based edge detectors. In the presence of noise and various signal to noise ratio conditions, the horizontal and vertical edges have been efficiently recovered with good Pratt figure of merit without applying any pre-processing, noise removal process, and post processing techniques. The PFOM was found to be quite stable with wide threshold range for various noise levels. The consistent performance of the proposed edge detector is proved with MSE and PSNR determination of detected edge images. Moreover, the proposed 2-D SDFT/HDFT-based edge detector performs well in developing edge maps for real-time road videos. The system-on-chip implementation of the 2-D SDFT/HDFT edge detector on Cyclone IV FPGA chip is also carried out for detecting the lane and road markings.

1 | INTRODUCTION

Vision based measurement (VBM) is being proposed and used in various automated applications such as advanced driver assistance systems (ADAS) and intelligent transportation systems [1, 4]. Camera-based vehicle instrumentation is becoming popular to analyze the intentions and state of a driver, and detect potential driver errors to significantly reduce car accidents [2]. The ADAS integrates a number of different functions such as lane change assist, forward and rear collision warning, park assist, and blind spot detection [3].

Edge detection is an important process for the lane position detection, tracking systems, advanced driver assistant systems, and intelligent transportation systems [4]. Lane detection is a two-step process, in which the first step is detecting the lanes and then fit to the parametric curve. The challenges involved in vision-based lane detection include the lack of clarity of lane markings, light reflections, illumination, shadows, poor visibility due to bad weather conditions and high noise levels. Due to the aforementioned conditions, the lane detection algorithms fail to distinguish lane lines from the road in an image [5]. This research work focuses on the first step to detect the lanes through developing edge maps. The performance of the lane detection process depends on the successful edge characterization on the road images. An edge in an image connects the boundary between two different regions. Edge provides sufficient information with reduced data size which is suitable for the image analysis for most of the applications. The image intensity shows abrupt changes at edges and the magnitude of derivative of the image intensity function is high at the location of edges [6]. A number of methods for edge detection involving different approaches could be found in literature. These are (i) statistical methods, (ii) difference methods, and (iii) curve fitting [6]. The conventional edge detectors such as Sobel [7], Canny [8], and Prewitt [9] are popular and employed for feature extraction of lane and road marking images. The classical methods such as Sobel and Prewitt detectors use the first directional derivative to determine the edges [6]. However, images with strong
noise cannot be handled by classical edge detectors. In [10], the fuzzy reasoning-based algorithm of the Fuzzy Sobel method is introduced which automatically obtains four threshold values, and applies fuzzy reasoning for edge enhancement. The edges extracted by this method are very clear and provides better representation for image edges and object contours. An improvement is proposed in Sobel edge detector with soft-threshold wavelet technique [11] for the removal of noise. The problems encountered in FPGA implementation of Sobel algorithm has been solved and improved through gradient calculation template with increased processing speed [12]. The zero-crossing edge detectors of another category use the second derivative along with Laplacian operator. A Laplacian of Gaussian filter is used as an edge detector for texture analysis of glass production [13]. LoG method computes the second derivative in the image and then find zero-crossing points. Since the second derivative of the image is used for edge detection, this filter is very sensitive to noise. To overcome this problem, firstly the noise should be reduced by applying Gaussian smooth filter and then Laplacian filter must be implemented to the image. The Canny detector is a Gaussian edge detector, is one of the most popular edge detectors in the literature and it has been widely used in many applications. Although the Gaussian detectors exhibit relatively better performance, they are computationally much more complex than classical derivative based edge detectors [6]. An improvement in Canny edge detection algorithm [14] is proposed towards improving the sensitivity of noise and loss of weak edge information. The concept of gravitational field intensity was introduced to replace image gradient. Two adaptive threshold selection methods based on the mean standard deviation of image gradient were used to enhance the noise suppression. A robust edge detection algorithm using multiple threshold approach (B-Edge) is proposed [15] to cover limitations of edge connectivity and thickness. The widely used Canny edge operator uses two thresholds and still witnesses a few gaps for optimal results. To handle the loopholes of the Canny edge operator, this algorithm uses the simulated triple thresholds that target the prime issues of the edge detection such as image contrast, effective edge pixels selection, errors handling, and similarity to the ground truth. Apart from the conventional edge detectors, the Hough transform is a much applied algorithm in lane detection and tracking systems. The computationally efficient hierarchical additive Hough transform is proposed for detecting only the straight lanes [16]. An efficient method for reliably detecting road lanes based on spatio-temporal images is proposed using Hough transform [17]. The cellular neural network (CNN) is a novel class of information processing system which was proposed in [18]. Its two-dimensional inputs and output make it suitable for image processing [19]. The edge detection based on CNN is proposed in [6] with differential evolution algorithm to obtain edges of digital images. A template learning set-up has been proposed and resultant cloning template detects the edges; however, the noisy images are not considered for edge detection. Another edge detection method based CNN is proposed in [20] for noisy images. The edge detection is carried out with two steps; in the first step, noise is removed with a set of CNN cloning templates and in the second step, the resultant noise removed images are given as inputs to the CNN with another set of cloning templates. Moreover, the improved results of edge detection is shown with salt and pepper noise of different levels. The Gaussian noise is not considered for the evaluation. A CNN-based noise removal technique for color images have been proposed with hybrid linear matrix inequality and swarm optimization procedure [21]. The cloning templates were designed for salt and pepper noise, speckle noise, and Gaussian noise, and noise removal has been evaluated for color images. These templates can be used to remove the noise efficiently. However, for edge detection, an another set of templates need to be designed. A bias feed CNN model has been proposed for glass defect detection in [22]. Bias input is a single number value in traditional CNN algorithm which is converted into a bias template and used to balance the brightness level of the image. Through the contribution from bias input, background reflections and negative effects arising from transparency are decreased. These cloning templates are obtained through particle swarm optimization. Generally, for the CNN models, there are three major template design methods used. They are: (i) intuitive way, (ii) template learning, and (iii) direct template derivation. The first requires intuitive thinking of the designer. It leads to quick results in simple cases; however, it does not guarantee to obtain the desired template. Moreover, designers need to conduct lots of experiments in both image processing and array dynamics. The second design method, the template learning, is extensively studied and a popular field of the CNN research. Almost all classical neural network training methods have been applied to the CNN structure. The third method can be applied when the desired function is exactly determined. The design methods depend on the particular template class [23]. In general, the edge detection is performed as a linear filtering process which convolutes the image pixel and kernel. In 1D sense, discrete Fourier transform (DFT) is a linear filter that performs the correlation between signal samples and basis function to extract the fundamental signal and harmonics. The 1D sliding DFT is computationally efficient as tuned filter to extract the particular fundamental and harmonic of interest from the input signal [24, 25]. Moreover, the instantaneous values of fundamental and harmonic can be extracted from the filter operation [26]. In 2D, similar to classical edge detectors, the 2D sliding DFT performs correlation between image pixel and kernel, in which correlation is the convolution of image pixel and 180° rotated kernel [27]. The 2D SDFT can be tuned to extract only edge frequencies from the image and reject all other frequencies including high noise levels. The pre-processing and post-processing can be completely eliminated through the 2D Sliding DFT/Hopping DFT edge detector, since the sliding DFT is a tuned 2D filter which can sharply tuned to extract the edges. The existing edge detectors requires pre-processing like Gaussian filtering in Canny and post-processing such as thinning in Canny, Sobel and Prewitt edge detectors. The CNN requires noise removal process as a separate process before edge detection. Moreover, it involves the difficult process of designing the cloning templates for a particular application. The performance of the edge detectors in the presence of noise is
poor for the traditional edge detectors. Hence, the existing edge detectors necessitates the following:

(i) pre-processing,
(ii) filtering,
(iii) noise removal,
(iv) post-processing,
(v) high computational complexity,
(vi) more resource utilization.

To this end, an edge detector is proposed based on 2D SDFT/HDFT and applied for road marking images in order to create an edge map. These edge maps can be employed in lane detection, tracking systems, and driver assistant systems. The proposed edge map generation method performs well for the road images. The performance is also verified for the known ground truth images corrupted with noise. The proposed method yields consistent performance for the noisy images with good PFOM, MSE, and PSNR. The SDFT based edge detector yields medium entropy for the detected edges compared to studied edge detectors. This paper is organized as follows: The proposed edge detection technique based on 2D SDFT/HDFT is described in Section 2. The performance evaluation and comparison of the proposed method with existing well-known edge-detectors and CNN-based edge detectors are discussed in Section 3. The FPGA implementation of the proposed edge detector is also presented in Section 3. Some important remarks are given in Section 4.

2 PROPOSED EDGE DETECTION TECHNIQUE

The block diagram representation of proposed edge detection methodology is shown in Figure 1. As a first step in edge detection, the 2D SDFT is performed on input image with $2 \times 2$ window as shown in Figure 2.

2.1 2D sliding DFT

The two-dimensional DFT is determined on a fixed-size window of pixels $N \times N$, which is regularly updated with new pixels and the oldest ones are discarded. Consider the $N \times N$ window is moved in the horizontal direction on the image row by row: The $(x,y)^{th}$ pixel of an input image is represented by $f(x,y)$ and $(k_1,k_2)^{th}$ bin of the $N \times N$ DFT at pixel position $(x,y)$ is $F_{x,y}(k_1,k_2)$ [28], which is represented by

$$F_{x,y}(k_1,k_2) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f_{\hat{x}+m,\hat{y}+n} W_N^{-k_1 m} W_N^{-k_2 n},$$

(1)

where $\hat{x} = x - N + 1$, $\hat{y} = y - N + 1$, $W_N = e^{2\pi i / N}$, and $k_1, k_2 = 0, 1, ..., N-1$. For the pixel $(x + 1, y)$, the 2D-DFT is represented as,

$$F_{x+1,y}(k_1,k_2) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f_{\hat{x}+m,\hat{y}+n} W_N^{-k_1 m} W_N^{-k_2 n},$$

(2)

Let $m+1 = p$ and shifting the limits of summation in (2),

$$F_{x+1,y}(k_1,k_2) = W_N^{k_1} \sum_{p=0}^{N-1} \sum_{m=0}^{N-1} f_{\hat{x}+p+m,\hat{y}+n} W_N^{-k_1 p} W_N^{-k_2 n}$$

$$+ \sum_{m=0}^{N-1} f_{\hat{x}+p+m,\hat{y}+n} W_N^{k_1} W_N^{-k_2 n}$$

$$- \sum_{m=0}^{N-1} f_{\hat{x}+m,\hat{y}+n} W_N^{k_1} W_N^{-k_2 n}.$$
Equation (3) can be rewritten in terms of current pixel \((x, y)\) and previous pixel \((x-1, y)\) as

\[
F_{x,y}(k_1, k_2) = \sum_{n=0}^{N-1} f_{x-n,j} W_{N}^{n k_1} W_{N}^{n k_2}.
\]

(4)

Let \(d_{x,y} = f_{x-N-1,j} + f_{x-1,j} - f_{x-N-1,j} - f_{x-1,j} + s\) and 1-D DFT of \(d_{x,y}\) is

\[
D_{x,y}(k_2) = \sum_{n=0}^{N-1} d_{x-N+1} W_{N}^{-n k_2}.
\]

Using (5), Equation (4) can be written as

\[
F_{x,y}(k_1, k_2) = W_{N}^{k_1} F_{x-N,y}(k_1, k_2) + W_{N}^{k_2} D_{x,y}(k_2)\]

(6)

\(D_{x,y}(k_2)\) can be determined by 1-D sliding DFT [25]

\[
D_{x,y}(k_2) = W_{N}^{k_2} \left[ D_{x,y-1}(k_2) + d_{x,y} - d_{x,y-N} \right].
\]

(7)

Replacing \(D_{x,y}(k_2)\) in (6),

\[
F_{x,y}(k_1, k_2) = W_{N}^{k_1} \left[ F_{x-N,y}(k_1, k_2) + W_{N}^{k_2} D_{x,y-1}(k_2) \right. \\
\left. + d_{x,y} - d_{x,y-N} \right].
\]

(8)

Equation (8) confirms that the 2D sliding DFT at position \((x, y)\) can be directly computed from the position \((x-1, y)\) using 1D sliding DFT. For \(2 \times 2\) window, Equation (8) becomes

\[
F_{x,y}(k_1, k_2) = W_{2}^{k_1} \left[ F_{x-1,y}(k_1, k_2) + W_{2}^{k_2} D_{x,y-1}(k_2) \right. \\
\left. + d_{x,y} - d_{x,y-2} \right].
\]

(9)

For detecting horizontal edges

\[
F_{x,y}(0, 1) = W_{2}^{k_1} \left[ F_{x-1,y}(0, 1) + W_{2}^{k_2} D_{x,y-1}(1) \right. \\
\left. + d_{x,y} - d_{x,y-2} \right].
\]

(10)

For detecting vertical edges

\[
F_{x,y}(1, 0) = W_{2}^{k_1} \left[ F_{x-1,y}(1, 0) + W_{2}^{k_2} D_{x,y-1}(0) \right. \\
\left. + d_{x,y} - d_{x,y-2} \right].
\]

(11)

The 1D DFT is the correlation of samples and basis functions, whose frequencies of \(k\) cycles within \(N\) samples width [24]. Correlation is either the convolution of flipped samples and basis function or the convolution of samples and flipped basis function. In the proposed 2D SDFT technique, the image pixel and sliding window are correlated. The horizontal (refer (10)) and vertical edges (refer (11)) have been detected by kernels, obtained from basis function, moving on the image as shown in Figure 2 as shaded pixels. The 2D SDFT performs linear filtering, which is equivalent to the convolution of 180° rotated kernel and image pixel in horizontal and vertical directions. After computing the 2D SDFT on each pixel of the image, each pixel is compared with a threshold to determine edge map of the image. Threshold is chosen as \(4 \times \) mean of \(F_{base-dir}\).

### 2.2 2D hopping DFT

The 1D hopping DFT [29] computes the \(N\)-point DFT at time index \(n\) using the precalculated \(N\)-point DFT at time index \(n-L\), where \(L\) is time hop. The HDFT can adjust the time hop between successive DFT outputs. For 2D HDFT, using (5) and (6)

\[
F_{x,y}(k_1, k_2) = W_{N}^{k_1} \left[ F_{x-N,y}(k_1, k_2) \right. \\
\left. + \sum_{p=0}^{N-1} d_{x,y+p} W_{N}^{-k_2 p} \right].
\]

(12)

To obtain 2-D HDFT \(F_{x,y}(k_1, k_2)\) with \(L\) time hop \(F_{x-L,y}(k_1, k_2), F_{x-1,y}(k_1, k_2)\) is repeatedly applied \(L\) times in \(F_{x,y}(k_1, k_2)\), that is, Equation (12). The resultant expression of 2-D HDFT is

\[
F_{x,y}(k_1, k_2) = W_{L}^{k_1} \left[ F_{x-L,y}(k_1, k_2) \right. \\
\left. + \sum_{p=0}^{N-1} d_{x,y+p} W_{N}^{-k_2 p} \right] + \sum_{p=0}^{L-1} W_{L}^{k_1} \sum_{p=0}^{N-1} d_{x,y+p} W_{N}^{-k_2 p} + \cdots \\
\left. + W_{N}^{(L-1)k_1} \sum_{p=0}^{N-1} d_{x,y+p} W_{N}^{-k_2 p} \right].
\]

(13)

Equation (13) can be written as

\[
F_{x,y}(k_1, k_2) = W_{N}^{k_1} \left[ F_{x-L,y}(k_1, k_2) \right. \\
\left. + \sum_{p=0}^{L-1} \sum_{p=0}^{N-1} d_{x,y+p} W_{N}^{(p-L+1)k_1} W_{N}^{-k_2 p} \right].
\]

(14)

The second term of Equation (14) is 1D DFT. Let

\[
D_{x,y}(k_2) = \sum_{p=0}^{L-1} \sum_{p=0}^{N-1} d_{x,y+p} W_{N}^{(p-L+1)k_1} W_{N}^{-k_2 p}.
\]

(15)
Equation (15) can be written in the form of 1-D SDFT as follows:

\[
D_{x,y}(k_2) = \frac{W_N}{k_2} \left[ D_{x,y-1}(k_2) + \sum_{p=0}^{L-1} d_{x-p,y} W_N^{(p-L+1)k_1} \right] \\
- \sum_{p=0}^{L-1} d_{x-p,y-N} W_N^{(p-L+1)k_1}
\]

\[
= \frac{W_N}{k_2} \left[ D_{x,y-1}(k_2) + \sum_{p=0}^{L-1} (d_{x-p,y} - d_{x-p,y-N}) W_N^{(p-L+1)k_1} \right].
\]

Let \( e_{x,y} = d_{x,y} - d_{x,y-N} \). Hence, Equation (16) can be written as

\[
D_{x,y}(k_2) = \frac{W_N}{k_2} \left[ D_{x,y-1}(k_2) + \sum_{p=0}^{L-1} (e_{x-p,y}) W_N^{(p-L+1)k_1} \right].
\]

The 1D SDFT of Equation (17) can be expressed in terms of 1D HDFT by repeating \( L \) times of SDFT as

\[
D_{x,y}(k_2) = \frac{W_N}{k_2} \left[ D_{x,y-L}(k_2) + \sum_{p=0}^{L-1} (e_{x-p,y-L}) W_N^{(p-L+1)k_1} \right]
+ \frac{W_N}{k_2} \left[ \sum_{p=0}^{L-1} (e_{x-p,y}) W_N^{(p-L+1)k_1} \right].
\]

Equation (18) can be rewritten as

\[
D_{x,y}(k_2) = \frac{W_N}{k_2} \left[ D_{x,y-L}(k_2) + \right.
+ \left. \sum_{p=0}^{L-1} \sum_{q=0}^{L-1} (e_{x-p,y-q}) W_N^{(p-q-L+1)k_1} \right].
\]

The 2D HDFT equation can be expressed as

\[
F_{x,y}(k_1, k_2) = \frac{W_N}{k_1} \left[ F_{x-L,y}(k_1, k_2) + D_{x,y}(k_2) \right].
\]

For \( L = 2 \) and \( N = 4 \), Equation (20) can be represented as

\[
F_{x,y}(k_1, k_2) = \frac{W_N}{k_2} \left[ F_{x-2,y}(k_1, k_2) + D_{x,y}(k_2) \right].
\]

For horizontal edges,

\[
F_{x,y}(0, 1) = \frac{W_N}{k_1} \left[ F_{x-2,y}(0, 1) + D_{x,y}(1) \right].
\]

For vertical edges,

\[
F_{x,y}(1, 0) = \frac{W_N}{k_1} \left[ F_{x-2,y}(1, 0) + D_{x,y}(0) \right].
\]

From Equation (1), the equivalent kernel of 2D SDFT and 2D HDFT for \( 2 \times 2 \) and \( 4 \times 4 \) size windows are shown in Figures 3 and 4, respectively.

### 3 | IMPLEMENTATION

The performance of the proposed edge detector is evaluated for different dataset of road images and compared with the well-known edge detectors such as Canny, Sobel, Prewitt, and CNN. The performance for the noisy images are determined for various noise level. The qualitative analysis has been carried out by computing the PFOM, entropy, MSE, and PSNR.

#### 3.1 | 2D SDFT based edge detector

The performance of the proposed 2D SDFT based edge detector is evaluated with road marking images from ROad MARkings (ROMA) image database [30]. The images of ROMA database possessing the following details have been considered for evaluating the proposed 2D SDFT based edge detector. Camera: Balser A101CP, image size: 1280 × 1024, Pixel size: 6.7 micro-meter × 6.7 micro-meter; all are color images and they are converted to gray scale for evaluation. The input images used for edge detection, edge map developed through the proposed method (refer Figure 1), overlaying the original image and the detected edges, red in color are shown in Figure 5. The detected lanes, lane markings, trees, and objects could be seen with clarity in the edge map. The performance of this edge detector is compared with the conventional edge detection algorithms such as Canny, Sobel, and Prewitt edge detectors. The comparison of the output of each edge detector for ROMA images are shown.
in Figures 6 and 7. It could be observed that the proposed edge
detector performs better on the road marking images. The pro-
posed method is also evaluated with RoadMarking dataset used
in [31]. The edge map comparison of each method are shown
in Figures 8 and 9 with input images of different lighting con-
ditions. The information extracted in the edge map can be used
for lane detection and lane position detection in tracking and
driver-less cars for decision making process. Furthermore, video
sequence has been tested for the proposed edge detector and
the algorithm could successfully create the edge map for the
video frames.

3.2 Cellular neural network (CNN)

CNN is a locally interconnected analog processor array
arranged to a regular 2D grid. The model of a 2D CNN is
composed of basic processing units called cells. Each cell is
FIGURE 6 Comparison of 2D sliding DFT-based edge detector with conventional edge detectors: Input images from Database ROMA

connected to its neighboring ones; therefore, only the adjacent cells can interact directly with each other [6]. For a CNN array with M rows and N columns on a 2D grid, the dynamics of each cell can be described by the state equations [20],

\[
\dot{x}_{ij}(t) = -x_{ij}(t) + \sum_{C(k,l)\in N_r(i,j)} A(i,j,k,l)y_{kl}(t) + \sum_{C(k,l)\in N_r(i,j)} B(i,j,k,l)u_{kl}(t) + T_{ij}
\]

\[
y_{ij}(t) = f(x_{ij}(t)) = \frac{1}{2}(|x_{ij}(t) + 1| - |x_{ij}(t) - 1|),
\]

where \(i = 1,2,3 \ldots M, \, j = 1,2,3 \ldots N, \, 1 \leq k \leq M, \, 1 \leq l \leq N; \, x_{ij}(t), \, u_{ij}(t), \, y_{ij}(t)\) are the state, the input and the output of the \((i,j)\)-th cell in the grid. The initial condition of \(x_{ij}(0) = 0\) and static input \(u_{ij} \leq 1\). \(A(i,j,k,l), \, B(i,j,k,l)\) denote the connection templates from cell \(C(k,l)\) to cell \(C(i,j)\); \(T_{ij}\) represents the bias of \((i,j)\)-th cell in the grid. Equation (24) shows that the state and the output of each cell are affected by the inputs and outputs of its neighboring cells. Moreover, for each cell \(C(i,j)\), the following set \(N_r(i,j)\), named \(r\) neighborhood, can be defined as

\[
N_r(i,j) = C(k,l) := \max(|k - i|, |l - j|) \leq r,
\]

where \(r\) denotes the neighborhood radius of each cell which is a positive integer, and the pairs \((i,j)\) and \((k,l)\) are the indices which express the position of cells, that is, the rows and the columns of the generic cell and its neighboring ones in the grid, respectively. The cells which belong to the \(r\)-neighborhood of \(C(i,j)\) are arranged in a maximum \((2r+1) \times (2r+1)\) grid whose central element coincides with \(C(i,j)\). For \(r = 1\), a space-invariant CNN with a \(3 \times 3\) neighborhood is defined uniformly by a string of 19 real numbers, called the CNN gene, that is, they include a uniform threshold \(T\), a feedback template \(A\) consisting of nine feedback synaptic weights and a control cloning template \(B\) consisting of nine control synaptic weights. The most challenging point of CNNs applications is to find the satisfactory feedback template \(A\), control template \(B\), and bias \(T\). The 19 real numbers together with initial condition and static input can completely determine the dynamical properties of the CNN.

The uncoupled CNN template chosen for edge detection of road images are

\[
A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad B = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}; \quad T = -0.5.
\]

Figure 10 shows the performance of CNN edge detector by using above mentioned templates for the ROMA Database images used in Figures 6 and 7. Similarly, the performance of CNN is compared with the proposed 2D edge detector for the
road images used in Figures 8 and 9 from the DataSet. These edge-detected road images are shown in Figure 11. The 2D SDFT performs better than CNN and it was observed that in the CNN output, the edge detected road images, certain noises were introduced.

3.3 Performance evaluation

3.3.1 Pratt’s figure of merit

Pratt’s figure of merit (PFOM) was proposed by Pratt for accuracy assessment of extracted edges [32]. It represents the deviation of an actual edge point \( (I_d) \) from the ideal edge \( (I) \) and it is defined as

\[
R = \frac{1}{\max(I, I_d)} \sum_{i=1}^{I_d} \frac{1}{1 + \delta e^2(i)},
\]

where \( e(i) \) is distance between \( I_d \) and \( I \), \( \delta \) is scaling constant (usually 1/9).

3.3.2 Performance of the proposed edge detector with and without noise

The PFOM is determined for the images, whose ideal edges are known in prior without adding any noise, and compared with the conventional Canny, Sobel, and Prewitt edge detectors (refer Figure 12(a)). The proposed method performs better than other techniques with a PFOM of 0.9545. Similarly, the comparison of performance evaluation with different SNR of 40 to 10 dB is shown in Figure 12(b). These input images have been tested for edge detection with CNN; the results are shown in Figure 13. The bias is chosen as \(-5\) for these noisy images, whereas \( A \) and \( B \) templates are retained as such. The proposed 2D SDFT edge detector performs well in the presence of noise. The threshold is chosen to be 0.4 for other conventional edge detectors. In case of Canny edge detector, the width of Gaussian filter \( (\sigma) \) is chosen for maximum PFOM. The Canny edge detector involves the multiple stages in the edge detection process. These are: (i) noise reduction by Gaussian filter, (ii) Sobel kernel to find edge gradient, (iii) non-maxima suppression to remove any unwanted pixels, and (iv) hysteresis thresholding. The Sobel and Prewitt edge detectors involves thinning process in the edge map generation. CNN performs well for the image without noise and 40 dB noise with only edge detection templates; however, the PFOM decreases from 30 to 10 dB noise cases. The detected edges of noise image of 10 dB is noticeably noisy and PFOM is poor. The edge detection of noisy images may be handled differently with CNN in two steps. They are: (i) first, the noise removal templates need to be designed and apply to CNN to remove the noise. (ii) Secondly, the edge detection templates need to be applied on the noise removed image. Hence, designing templates may be a great challenge in the case of CNN.
Moreover, the conventional edge detectors did not demand separate step to remove the noise, whereas CNN may require that important procedure; otherwise, it fails to detect the edges. In comparison with these edge detectors, the proposed 2D SDFT/HDFT-based edge detector performs better without involving any filtering or thinning or noise removal process. Furthermore, the range of threshold has been found for each edge detectors, except CNN, without noise and with noise cases for the input image shown in Figure 12(a),(b). The simulation has been performed on input image to find the range of threshold for which PFOM is retained at maximum value. The simulation results have been shown in Figure 14(a)–(f) for without noise and variation of noise for 40 to 10 dB, respectively. It can be observed from Figure 14(a) that the Canny demands the $\sigma$ variation with threshold. The variation of both $\sigma$ and threshold are varied to obtain maximum PFOM as shown in Figure 15. The Sobel and Prewitt edge detector threshold range is the same value for the cases of with and without thinning process. In these edge detectors, the PFOM is found to be good when thinning process is applied for edge detection; without thinning process, the PFOM was found to be poor. The proposed edge detector based on 2D SDFT/HDFT performs equally well with Canny edge detector for the noise input image. It could be noticed from Figure 14(a)–(f) that 15 dB noise is tolerable with the proposed edge detector, below which the performance is degraded and threshold range cannot be obtained.

3.3.3 Entropy

The content of an information that is present in an image is measured through entropy function [33].

$$H(I) = -\sum_{i} p_i \log p_i,$$  \hspace{1cm} (28)

where $I$ represents an image and $p_i$ is the rate of recurrence of pixels with intensity $i$. Shannon’s entropy states that lesser value of $H(I)$ designates information with low content; however, on the other hand, too high value of $H(I)$ indicates the presence of noise and double edges. Therefore, $H(I)$ should always lie in the middle of both extremes for best results. The entropy of 14 images from ROMA DataBase and 14 images from RoadMarking DataSet are calculated and the results are plotted in Figure 16(a), (b). The proposed SDFT edge detector possesses medium entropy compared to the Canny, Sobel,
Prewitt, and CNN edge detectors. The low entropy indicates less edges are detected, whereas the high entropy indicates that the double edges are detected from the image. The attained entropy difference values are less than Canny but higher than Prewitt’s and Sobel’s and relies on nearby Canny in some scenarios, which state that the entropy difference values fall in between the two extremes. The CNN does not show a consistent entropy either in the medium level or high. In Figure 16(a), for some images, the entropy is medium. For other images, it is high evidencing the double edge detection. Figure 16(b) shows that the CNN-based edge detection has high entropy and hence the double edge detection for selected road images. The performance comparison of the proposed method with the conventional and CNN edge detectors are given in Table 1.

### 3.3.4 MSE and PSNR

The mean-square error (MSE) and peak signal-to-noise-ratio (PSNR) are calculated for the proposed 2D SDFT edge detector. A comparison is made with other conventional edge detectors and CNN. The MSE and PSNR [20] are given by

\[
MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i,j) - K(i,j)]^2,
\]

\[
PSNR = 10 \log_{10} \left( \frac{R^2}{MSE} \right),
\]

where \(M \times N\) is the size of the image, \(I(i,j)\) is ideal edge and \(K(i,j)\) is the detected edge, and \(R\) is the maximum possible pixel value of the detected edge. The calculated MSE and PSNR are listed in Table 2. Moreover, the MSE and PSNR can be determined only for those images for which the ground truth is known. It could be noticed that the proposed 2D SDFT performs invariably well in the presence of noise, whereas the other methods show variation in performance in the presence of noise. Though the CNN performance is good for certain cases, it is entirely depending on the bias and degrades with noise. For bias \(T = -5\) case, the PFOM is not good, but the MSE and PSNR are good up to noise level 15 dB. For bias \(T = -0.5\), which is used for testing the road images, the performance is good up to the noise level of 40 dB and then it degrades.

### 3.4 FPGA Implementation

The proposed edge detection algorithm is implemented in FPGA. The Altera DE2-115 FPGA board of clock frequency 50 MHz with Cyclone IV 4CE115F29 FPGA device is used for implementing the algorithm in real-time. The proposed edge detector blocks are developed in simulink environment and through HDL-coder 3.1, the equivalent verilog codes are generated and downloaded into FPGA. The PC and FPGA board communication is established by USB-blaster. The SDRAM of size 8-bit 307200 (640 × 480) bytes is used to store a frame of a video from the camera. The edge detected video is brought out to the real-world through DAC-VGA connector, and finally the edge detected video is displayed in the TFT monitor. The proposed edge detector algorithm is made available as system-on-chip device by programming the FPGA through active serial programming, so that the algorithm is available permanently on the chip. Figure 17 shows experimental set-up used for capturing the indoor environment and detected edge map displayed with a monitor in red lines, Altera FPGA board with a camera and its interface. The utilization summary of the 2D SDFT method with 2 × 2 window on Cyclone IV FPGA,
DE2-115 is as follows: Total logic elements 39,697 / 114,480 (35%), total combination functions 39,435 / 114,480 (34%), dedicated logic registers 38,282 / 114,480 (33%), total memory bits 59,344 / 3,981,312 (1%), embedded multiplier 9-bit elements 3 / 532 (<1%). The proposed edge detector algorithm based on 2D SDFT and 2D HDFT are implemented in Altera DE2i-150 to accommodate both algorithms for comparison.
TABLE 1  Comparison of the proposed edge detector with the existing edge detectors [15, 20]

| Edge detector   | Technique                        | Advantages                                                                 | Disadvantages                                                                 |
|-----------------|----------------------------------|----------------------------------------------------------------------------|------------------------------------------------------------------------------|
| 2-D SDFT/HDFT   | Sliding Window DFT               | Good noise performance; Accurate; Preprocessing and post processing are absent as in Canny, Sobel and Prewitt; Detection of edges and orientation; Good localization; Good response | SNR below 10 dB, the threshold does not hold a range.                        |
| Canny           | Gaussian                         | Improved SNR due to the presence of Gaussian filter; Good localization; Good response; Accurate | False zero crossing; slow and complex; extensive computations                |
| Sobel           | First order differential based   | Simple; Detection of edges and orientation                                  | Sensitive to noise; Inaccurate                                               |
| Prewitt         | First order differential based   | Simple; Detection of edges and orientation                                  | Sensitive to noise; Inaccurate                                               |
| CNN             | Cloning templates; first-order non-linear differential equation defining the dynamics via neighborhood cell. | Improved SNR with certain bias values. | Design of templates for noise removal and edge detection as two separate steps. |

TABLE 2  Comparison of MSE and PSNR of different edge detectors

| Methods   | Without noise | 40 dB noise | 30 dB noise | 20 dB noise | 15 dB noise | 10 dB noise |
|-----------|---------------|-------------|-------------|-------------|-------------|-------------|
| MSE       |               |             |             |             |             |             |
| 2D SDFT   | 0.0133        | 0.0133      | 0.0133      | 0.0133      | 0.0134      | 0.0135      |
| Canny     | 0.0230        | 0.0134      | 0.0135      | 0.0135      | 0.0133      | 0.0136      |
| Sobel     | 0.0133        | 0.0136      | 0.0136      | 0.0139      | 0.0134      | 0.0141      |
| Prewitt   | 0.0133        | 0.0138      | 0.0140      | 0.0138      | 0.0137      | 0.0140      |
| CNN (T = −5) | 0.000133 | 0.000133     | 0.0110      | 0.0024      | 0.0072      | 0.0413      |
| CNN (T = −0.5) | 0        | 0.000000953 | 0.0219      | 0.2352      | 0.3289      | 0.3815      |
| PSNR      |               |             |             |             |             |             |
| 2D SDFT   | 18.7506       | 18.7506     | 18.7506     | 18.7506     | 18.7362     | 18.7038     |
| Canny     | 16.3785       | 18.7362     | 18.6860     | 18.6860     | 18.7506     | 18.6646     |
| Sobel     | 18.7506       | 18.7362     | 18.6646     | 18.6860     | 18.7362     | 18.5010     |
| Prewitt   | 18.7506       | 18.6012     | 18.5456     | 18.6768     | 18.7182     | 18.5456     |
| CNN (T = −5) | 38.7506 | 38.7506      | 37.3006     | 26.4700     | 21.4053     | 15.8429     |
| CNN (T = −0.5) | ∞    | 60.2048      | 16.5972     | 6.2863      | 4.8295      | 4.1852      |

The resource utilization summary of the chip Cyclone IV GX EP4CGX150DF31C7 FPGA is as follows: SDFT 4 x 4 window size: Total logic elements 48,784 / 149,760 (33%), Total registers 47268, Total pins 360/508 (71%), total memory bits 51,152 / 6,635,520 (< 1%), embedded multiplier 9-bit elements 8 / 720 (1%). HDFT 4 x 4 window size: Total logic elements 100,228 / 149,760 (67%), Total registers 98535, Total pins 360/508 (71%), total memory bits 51,152 / 6,635,520 (< 1%), embedded multiplier 9-bit elements 8 / 720 (1%).

3.4.1  Real-time road videos

The proposed edge detector is tested with the following environments for edge detection. (i) Indoor environment; (ii) in the corridor; (iii) in the outdoor on the road. In the first case, the video is recorded in indoor environment by slowly moving the camera in our laboratory space and all the edges of the room could be detected with both 2-D SDFT/HDFT detector. In the second case, the video is recorded in the corridor of the department with varying lighting conditions. Light in the corridor switches on/off automatically by sensing the presence of person. A person walking towards the camera could be detected with all the edges in the corridor with poor and better lighting conditions. In the third case, the video is recorded for outdoor environment in real-time road conditions; the lane markings and vehicles are detected by this proposed method.

4  CONCLUSION

The 2D SDFT/HDFT is used for developing edge maps of road images in real-time. The suitability of this edge detector is explored for lane, road markings and object detection in road images and videos. The proposed edge map generation technique performs better in comparison with the conventional Canny, Sobel, and Prewitt edge detectors without demanding any filtering and thinning processes in the presence of noise.
REFERENCES

1. Shirmohammadi, S., Ferrero, A.: Camera as the instrument: The rising trend of vision based measurement. IEEE Trans. Instrum. Meas. Mag. 17(3), 41–47 (2014)

2. Beauchemin, S.S., et al.: Portable and scalable vision-based vehicular instrument for the analysis of driver intentionality. IEEE Trans. Instrum. Meas. 61(2), 391–401 (2012)

3. Milind Patole, S., et al.: Automotive radars: A review of signal processing techniques. IEEE Signal Process. Mag. 34(2), 22–35 (2017)

4. McCall, J.C., Trivedi, M.M.: Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation. IEEE Trans. Intell. Transp. Syst. 7(1), 20–37 (2006)

5. Lee, C., Moon, J.-H.: Robust lane detection and tracking for real-time applications. IEEE Trans. Intell. Transp. Syst. 19(12), 4043–4048 (2018)

6. Basturk, I., Gunay, E.: Efficient edge detection in digital images using a cellular neural network optimized by differential evolution algorithm. Expert Syst. Appl. 36, 2645–2650 (2009)

7. Sobel, I., Feldman, G.: A 3x3 isotropic gradient operator for image processing. Presented at the Stanford Artificial Project in 1968, unpublished but often cited. In: Duda, R., Hart, P. (eds.) Pattern Classification and Scene Analysis, pp. 271–272, John Wiley and Sons, New York (1973)

8. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Machine Intell. PAMI-8(6), 679–698 (1986)

9. Prewitt, J.M.S.: Object enhancement and extraction. In: Lipkin, B., Rosenfeld, A. (eds.) Picture Processing and Psychopictorics, pp. 75–149,Academic Press, New York (1970)

10. El-Khamy, S.E., Lotfy, M., El-Yamany, N.: A modified fuzzy Sobel edge detector. In: Proceedings of IEEE 17th National Radio Science Conference, pp. 1–9.IEEE, Piscataway, NJ (2000)

11. Gao, W., et al.: An improved Sobel edge detection. In: Proceedings of 3rd International Conference on Computer Science and Information Technology (ICCSIT), pp. 67–71.IEEE, Piscataway, NJ (2010)

12. Zhang, K., et al.: An improved Sobel edge algorithm and FPGA implementation. In: Proceedings of the 8th International Congress on Information and Communication Technology, vol. 131, pp. 243–248 (2018)

13. Ozturk, S., Akdemir, B.: Comparison of edge detection algorithms for texture analysis on glass production. Procedia-Social Behav. Sci. 195, 2675–2682 (2015)

14. Rong, W., et al.: An improved CANNY edge detection algorithm. Proc. IEEE Int. Conf. Mechatron. Automat. 2(2), 577–582 (2014)

15. Mittal, M., et al.: An efficient edge detection approach to provide better edge connectivity for image analysis. IEEE Access 7, 33240–33255 (2019)

16. Satzoda, R.K., et al.: Hierarchical additive hough transform for lane detection. IEEE Embed. Syst. Lett. 2(2), 23–26 (2010)

17. Jung, S., Youn, J., Sull, S.: Efficient lane detection based on spatiotemporal images. IEEE Trans. Intell. Transp. Syst. 17(1), 289–295 (2016)

18. Chua, L.O., Yang, L.: Cellular neural networks: Theory. IEEE Trans. Circuits Syst. 35, 1257–1272 (1988)

19. Chua, L.O., Yang, L.: Cellular neural networks: Applications. IEEE Trans. Circuits Syst. 35, 1273–1290 (1988)

20. Li, H., et al.: Edge detection of noisy images based on cellular neural networks. Commun. Nonlinear Sci. Numer. Simulation 16, 3746–3759 (2011)

21. Su, T.-J., Cheng, J.-C., Lin, Y.-J.: Cellular neural network based on hybrid linear matrix inequality and particle swarm optimization for noise removal or color image. Adv. Mater. Res. 422, 771–774 (2011)

22. Ozturk, S., Akdemir, B.: Novel BiasFeed cellular neural network model for glass defect inspection. In: Proceedings of 2016 International Conference on Control, Decision and Information Technologies (CoDIT), pp. 366–371 (2016)

23. Zarandy, A.: The art of CNN template design. Int. J. Circ. Theo. Appl. 27, 5–23 (1999)

24. Lyons, R.: Understanding Digital Signal Processing. Prentice Hall PTR, New Jersey (2001)

25. Jacobsen, E., Lyons, R.: An update to the sliding DFT. IEEE Signal Process. Mag. 21(1), 110–111 (2004)

26. Sumathi, P., Janakiraman, P.A.: Integrated phase-locking scheme for SDFT-based harmonic analysis of periodic signals. IEEE Trans. Circuits and Syst. II: Express Briefs 55(1), 51–55 (2008)

27. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Pearson Education, London (2011)

28. Park, C.S.: 2D discrete Fourier transform on sliding windows. IEEE Trans. Image Process. 24(3), 901–907 (2015)

29. Park, C.S., Ko, S.J.: The hopping discrete Fourier transform. IEEE Signal Process. Mag. 31(2), 135–139 (2014)

30. Veit, T., et al.: Evaluation of road marking feature extraction. In: Proceedings of 11th IEEE Conference on Intelligent Transportation Systems, 2008, Beijing, China

31. Wu, T., Ranganathan, A.: A practical system for road marking detection and recognition. In: IEEE Intelligent Vehicles Symposium, 2012, Spain, pp. 25–30

32. Pratt, W.K.: Digital Image Processing. John Wiley & Sons, New York (1991)

33. Verma, O.P., Agrawal, N., Sharma, S.: An optimal edge detection using modified artificial bee colony algorithm. Proc. Nat. Acad. Sci., India Sect. A, Phys. Sci. 86(2), 157–168 (2016)

34. Qiu, T., Yan, Y., Lu, G.: An Autoadaptive Edge-Detection Algorithm for glass defect inspection. In: 2016 International Conference on Control, Decision and Information Technologies (CoDIT), pp. 366–371 (2016)