Coordinates Transformed Signal Compression Method (CoTSiC): A Novel Algorithm for Tele-Medicine Applications

Soham Pawar[a] and Madhav Rao[b]
International Institute of Information Technology Bangalore, Bangalore, India
{soham.pawar127, mr}@iiitb.ac.in

Keywords: Tele-Medicine, Graph-Inspired Compression, Signal Compression, Image-Based Signal Compression.

Abstract: Robust Telemedicine refers to the provision of reliable remote medical services, which primarily depends on seamless transmission of either recorded signals or video information of patients in compressed form. A wide range of physiological signals which are typically seen in the display monitor of medical instruments including ECG, Blood Pressure, Oxygen levels, EEG signals and others are beneficial if remotely transmitted through reliable channels. Conventionally, the compression techniques applied to the signals are complex and compute-intensive, making it rarely viable at the remote patients' end, where the compute infrastructure is scarcely available. To address this challenge, the paper introduces a lightweight compression algorithm specifically designed for these tele-healthcare applications. This work transforms the picture of the signal at the source into a compressed array of data points. This array is sent to the remote healthcare facility and then re-constructed into a minimalistic form of the signal. The proposed method offers a compression factor in the range of 3.87× to 2.82× for a variety of signals including EEG, ECG, and SPO_2 signals. Additionally, an acceptable SSIM of above 92.10%, and PSNR of above 40 dB is characterized for the reconstructed image of different physiological signals investigated.

1 INTRODUCTION

Continuous Health Monitoring (CHM) is emerging as a vital component of proactive healthcare (Ozkan et al., 2020; Jiang et al., 2022; Nandi and Rao, 2022; Shaikh et al., 2023; Wu et al., 2022b). CHM systems leverage wearable or implantable devices for the real-time capture, analysis, or transmission of various physiological signals (Nia et al., 2015; Penhaker, 2022; Elhosary et al., 2019). Wearable form factor devices (Faisal et al., 2022; Ma et al., 2023; Chandrasekhar et al., 2020) and implantable devices (Martínez et al., 2023; Molloy et al., 2022), have emerged and made significant progress in terms of telehealth monitoring. These CHM systems are designed to be unobtrusive yet effective, continuously collecting vital physiological data, typically including blood pressure (Nandi and Rao, 2022), oxygen levels (Nwibor et al., 2023; Cao et al., 2012), electrocardiogram (ECG) (Spanò et al., 2016), electromyogram (EMG) (Chandrasekhar et al., 2020) and electroencephalogram (EEG) signals (Dabbaghian et al., 2019; Imtiaz et al., 2019).

The advent of deep neural network (DNN)-based methodologies has revolutionized this domain. DNNs excel at extracting complex features from large data sets, enabling nuanced analysis of physiological signals (Wu et al., 2022a; Doshi et al., 2021). However, the compute-intensive nature of DNNs and frequent data transmission needs pose challenges, especially for less equipped centers (Luo et al., 2022). Available bandwidth often limits data transmission quality in CHMs. In-device signal compression offers a potential solution to these bandwidth challenges. However, traditional compression algorithms like least absolute shrinkage (LARS) (Rudelson and Vershynin, 2006), Selection operator (LASSO) (Qaisar et al., 2013), and Sparse Bayesian Learning (Mamaghanian et al., 2011) are computationally intensive and unsuitable for resource-constrained setups. This paper introduces a novel compression method and system specifically designed for CHM setups.

The CoTSiC algorithm efficiently extracts signal profiles from medical monitors and transmits pixelated positional data for the monochrome version of the image. Supplying a text stream with 2-dimensional (2D) positional data is expected to re-

[a] https://orcid.org/0009-0008-1612-803X
[b] https://orcid.org/0000-0003-2278-9148

Pawar, S. and Rao, M.
Coordinates Transformed Signal Compression Method (CoTSiC): A Novel Algorithm for Tele-Medicine Applications.
DOI: 10.5220/0013173200003911
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 18th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2025) - Volume 2: HEALTHINF*, pages 603-610
ISBN: 978-989-758-731-3; ISSN: 2184-4305
Proceedings Copyright © 2025 by SCITEPRESS – Science and Technology Publications, Lda.

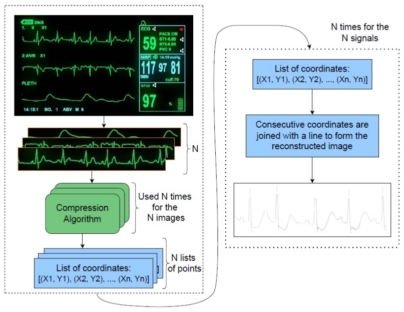
N times for the N signals
List of coordinates: [(X1, Y1), (X2, Y2), ..., (Xn, Yn)]
Consecutive coordinates are joined with a line to form the reconstructed image
N
Compression Algorithm
Used N times for the N images
N lists of points
List of coordinates: [(X1, Y1), (X2, Y2), ..., (Xn, Yn)]

Figure 1: Schematic showing the overview of the proposed Coordinates Transformed Signal Compression method (left), which operates on the medical instrument display units for CHM applications, along with the reconstructed image formed (right) by connecting the consecutive data points by a line.

duce both transmission and memory costs. Conventional CS approaches treat compression and reconstruction as separate processes, often leading to inefficiencies (Shoaib et al., 2014). Modern web cameras with wide coverage and depth are equipped to continuously capture monitor images, transferring pixelated data for remote signal reconstruction. The *CoTSiC* scheme, depicted in Figure 1, overcomes the limitation of CS methodologies where compressed signals are not readily available for direct analysis. In contrast, *CoTSiC* transforms data into a pictorial domain, where pixelated data is crucial for reconstruction. The *CoTSiC* method offers three benefits: i) Flexibility and compatibility with various platforms, including workstations, smartphones, and edge computing; ii) Independence from specific medical instruments, applicable to any diagnostic platform displaying signals; iii) Versatility across different signals, validated for ECG, SpO_2, and EEG signals.

2 PROPOSED METHOD

The proposed *CoTSiC* scheme follows six major steps which is illustrated in the Figure 2, and individual steps are further discussed.

2.1 Image Preparation

The image is blurred by employing an averaging scheme across the entire image. The kernel size of 5, configured with equal weightage to all parameters of the kernel shows that each pixel is shared equally across all 8 neighbours, leading to a blurred image. The scaling factor was applied to normalize the pixel levels according to the size chosen. This image is then converted to a grayscale image using Otsu's binariza-

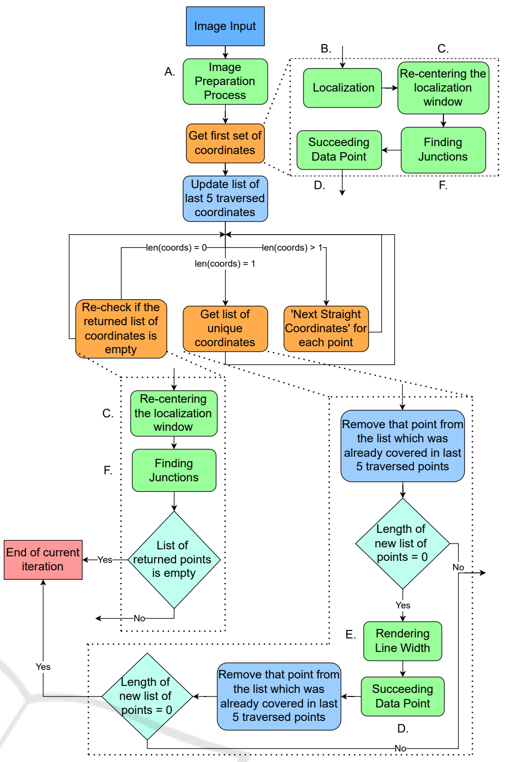
Image Input
A.
Image Preparation Process
B.
C.
Localization
Re-centering the localization window
Succeeding Data Point
Finding Junctions
Get first set of coordinates
D.
F.
Update list of last 5 traversed coordinates
len(coords) = 0
len(coords) > 1
len(coords) = 1
Re-check if the returned list of coordinates is empty
Get list of unique coordinates
'Next Straight Coordinates' for each point
C.
Re-centering the localization window
F.
Finding Junctions
List of returned points is empty
End of current iteration
Yes
No
Remove that point from the list which was already covered in last 5 traversed points
Length of new list of points = 0
No
Yes
E.
Rendering Line Width
Succeeding Data Point
D.
Remove that point from the list which was already covered in last 5 traversed points
Length of new list of points = 0
Yes
No

Figure 2: Flowchart showing the proposed *CoTSiC* method for extracting sequence of points from a given image.

tion scheme (Xue and Titterington, 2011). A boundary of 27 pixels is then added to this image and further converted into a two-dimensional (2D) array and retained.

The stored array is operated upon by the Harris Corner detection algorithm to get the list of all the corner points and the set of other points associated with each of these corner points. Post corner detection, the algorithm finds the nearest valid points from the upper-left corner of the image. This is infused as the starting position to trace the signal from a given image.

2.2 Localization

Localization attempts to find the most nearest point as mentioned above with the help of an expanding square-shaped boundary whose upper-left corner is fixed at that of the image. The expansion stops when the boundary finds a pixel above the threshold value. Figure 3 exhibits the working of a localization scheme where the progression of an enlarging square detects the features in the image. The enlarging square boundary starts with a size of 0 and keeps increasing with one unit (the active boundary is shown in blue, and the previous boundary is in yellow).

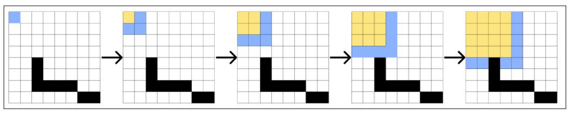
Figure 3: Progression of enlarging square boundary for finding nearest non-zero pixel for a given image.

2.3 Re-Centering the Localizing Window

This step is applied to find the best square window that maximizes the captured pixelated data, given the extracted pixel positions and size of the enlarging square. The centre of intensity (COI) is computed and the centre point is repositioned based on the computed COI. The coordinates of COI (Center of Intensity) is expressed as the weighted average of intensity along X and Y coordinates, also expressed below in the Equation 1.

$$X_{COI} = \frac{\sum_{i=1}^{n} I_i X_i}{\sum_{i=1}^{n} I_i}, Y_{COI} = \frac{\sum_{i=1}^{n} I_i Y_i}{\sum_{i=1}^{n} I_i} \quad (1)$$

This COI is computed and stored for uniquely identifying the location of different Harris corners later in the process when required. Another use of repositioning is to find the best point such that the deduced contours are uniformly distributed and are as accurate as possible. The resulting 3D plot is centered around the adjusted coordinates.

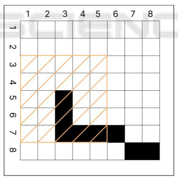
Figure 4: Temporary square in Grapher_3D module, enlarged to extract the collection of points in the square.

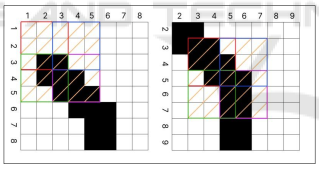
Figure 5: Representation of four quadrants for finding the direction and best-fit line with respect to the succeeding pixel.

2.4 Succeeding Data Point

This step identifies the following pixelated coordinate position, given the current center point and its direction with respect to the preceding point. This step assumes a square of fixed length defined by its center location as first input. It iterates through all the coordinates and checks whether any of the pixels are a part of Harris corner. If any Harris corners are detected, the directions and coordinates for multiple possible paths are stored. If not, then the current square under processing contains only a single path without any sharp edge, thereby a best-fit line is drawn to the

```
Input: $window\_size$: Size of the window
$center$: Center point of current window
$IMG\_F$: 2D matrix of intensities from image
Output: $point$: Center point of re-centered
        window
$sum\_ints \leftarrow 0$, $sum\_X \leftarrow 0$, $sum\_Y \leftarrow 0$;
$mid \leftarrow (window\_size - 1)/2$;
for $x, y \leftarrow 0$ to $window\_size - 1$ do
    $X\_temp \leftarrow center_x - mid + x$;
    $Y\_temp \leftarrow center_y - mid + y$;
    $value \leftarrow 255 - IMG\_F[Y\_temp][X\_temp]$;
    $sum\_X \leftarrow sum\_X + (x - mid) \cdot value$;
    $sum\_Y \leftarrow sum\_Y + (y - mid) \cdot value$;
    $sum\_ints \leftarrow sum\_ints + value$;
end
$CX\_T \leftarrow center_x + \lfloor sum\_X / sum\_ints \rfloor$;
$CY\_T \leftarrow center_y + \lfloor sum\_Y / sum\_ints \rfloor$;
return $[CX\_T, CY\_T]$;
```
Algorithm 1: Re-center Window (Block C).

next coordinate. For instance, in the Figure 5 shown, if the current location of (5,5) and the direction of 'SE' (South-East) is given, then a line extends from the current position to the succeeding coordinate (6,6) with the direction 'S' (South). Figure 5 illustrates the square size of 5×5 pixels. Here, the blue-coloured square inside the bigger shaded square is considered quadrant 1 (Q1), red is Q2, green is Q3 and pink is Q4. Hence, all coordinates and its directions are computed based on the quadrants considered. The moving square then centres to this new point, but the coordinates considered for the new position remains to be Q1, Q3 and Q4. This is because, for the previous point, the direction computed was 'SE' and hence in the new position of square, the direction of 'NW' points to Q2 and the same is not considered to find the new best-fit line. Once the slope of the best-fit line is computed, the farthest point among all from the quadrants considered with the same slope is picked for rendering the signal. This new point is then employed to compute the coordinates of a next point in the sequence such that it lies exactly at the center of the line width while rendering the signal.

```
Input: $IMG\_F, CX, CY, KER\_SIZE,$
       $LIMIT, PREV, PREV\_COORDS,$
       $IGNORE\_HARRIS$
Output: $fp\_fQS$
$opp \leftarrow \{SW : [Q2, Q3, Q4], W :$
  $[Q2, Q3], NW : [Q1, Q2, Q3], N :$
  $[Q1, Q2], NE : [Q1, Q2, Q4], E :$
  $[Q1, Q4], SE : [Q1, Q3, Q4], S : [Q3, Q4]\}$;
$s \leftarrow (KER\_SIZE - 1)/2$;
$x, y, vals, Q1, Q2, Q3, Q4 \leftarrow \emptyset$;
```

```
Q(x,y) ←
  $\begin{cases} Q1,2 & \text{if } x == CX \wedge y < CY \\ Q3,4 & \text{if } x == CX \wedge y > CY \\ Q2,3 & \text{if } y == CY \wedge x < CX\,; \\ Q1,4 & \text{if } y == CY \wedge x > CX \\ Q1,2,3,4 & \text{if } [x,y] == [CX,CY] \end{cases}$
for $Y \leftarrow (CY - s)$ to $(CY + s)$ do
    for $X \leftarrow (CX - s)$ to $(CX + s)$ do
        $value \leftarrow IMG\_F[Y][X]$;
        if $value < LIMIT$ then
            $intensity \leftarrow (255 - value)/255$;
            $x \leftarrow x \cup \{X\}, y \leftarrow y \cup \{Y\}$;
            $vals \leftarrow vals \cup \{intensity\}$;
            $v\_coords \leftarrow v\_coords \cup [X,Y]$;
            if $X == CX$ or $Y == CY$ then
                add $[X,Y,intensity]$ to
                  $Q(X,Y)$;
            else
                add $[X,Y,intensity]$ to the
                  corresponding quadrant
                  array;
            end
        end
    end
end
if $PREV$ is not empty then
    $final\_QS \leftarrow \bigcup$ elements in quadrant
      arrays present in $opp[PREV]$;
end
else
    $final\_QS \leftarrow Q1 \bigcup Q2 \bigcup Q3 \bigcup Q4$;
end
$wmean\_x\_fQS, wmean\_y\_fQS \leftarrow 0$;
$final\_QS, pp, pp\_fQS, fp\_fQS \leftarrow \emptyset$;
$bb\_f \leftarrow$ slope of best fit line for $v\_coords$;
$bb\_f \leftarrow \max(bb\_f, 10^{-7})$
 $angle\_fQS \leftarrow \arctan(bb\_f)$;
$dir\_gen\_fQS \leftarrow$ Algorithm 5$(angle\_fQS)$;
for $(x,y,val) \in final\_QS$ do
    $p1, p2 \leftarrow$ end-points of best fit line within
      the current window;
    if $(p1_x == x \vee p2_x == x) \vee (p1_y ==$
      $y \vee p2_y == y)$ then
        $pp\_fQS \leftarrow pp\_fQS \cup \{[x,y]\}$;
    end
end
```

Algorithm 2: Next Data Point (Block D).

2.5 Rendering Line Width

As shown in the Figure 6, the triangle formed from the farthest point in the vertical and horizontal direction each also highlighted by green line, aids in finding the length of the purple line (line thickness).

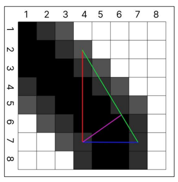
Figure 6: Illustration, depicting the estimation of line width.

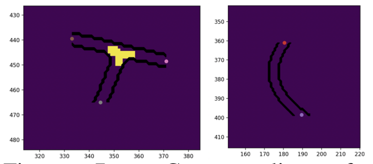
Figure 7: Contour lines for case#1 (left): Around a Harris Corner point and case#2 (right): when no coordinates for the next point are found.

```
for $point \in pp\_fQS$ do
    $angle\_curr \leftarrow \arctan\left(\frac{\text{point}_y - CY}{\text{point}_x - CX}\right)$;
    $direction \leftarrow$ Algorithm 5$(angle\_curr)$;
    if $direction \neq dir\_gen\_fQS$ then
        remove $point$ from $pp\_fQS$;
    end
end
if current window has a Harris Point then
    $fp\_fQS \leftarrow$ Algorithm 8$(point) \forall$ points in
      the window;
end
else
    forall $point \in pp\_fQS$ do
        $fp\_fQS \leftarrow$
          $fp\_fQS \cup$ Algorithm 6$(point)$;
    end
end
return $fp\_fQS$;
```

Algorithm 3: Next Data Point (Block D) - Continuation.

```
Input: $angle$
Output: $dir$
$dir \leftarrow$ 'E';
if $angle \in [0.125, 0.375] \cup [-0.875, -0.625]$ then
    $dir \leftarrow$ 'SE';
end
else if $angle \in [0.375, 0.625] \cup [-0.625, -0.375]$
  then
    $dir \leftarrow$ 'S';
end
else if $angle \in [0.625, 0.875] \cup [-0.375, -0.125]$
  then
    $dir \leftarrow$ 'SW';
end
return $dir$;
```

Algorithm 4: Direction Pointing Function.

2.6 Finding Junctions

The *Succeeding Data* step presents several points defining different paths for junction points, and their corresponding directions in case the region under study comprises of Harris corners. The contour lines for that region is found initially. A Pre-processing step is applied to ensure that the thickness of the line

```
Input: $point, beta\_bar, img\_f, limit, length$
Output: $fp$
$possible \leftarrow \emptyset$;
$xc, yc \leftarrow point$;
$l\_xc, u\_xc \leftarrow xc$;
$l\_yc, u\_yc \leftarrow yc$;
while $img\_f[yc][l\_xc - 1] <$
  $limit \wedge |xc - l\_xc + 2| \leq length$ do
    $l\_xc \leftarrow l\_xc - 1$;
end
while $img\_f[yc][u\_xc + 1] <$
  $limit \wedge |u\_xc - xc + 2| \leq length$ do
    $u\_xc \leftarrow u\_xc + 1$;
end
do the same decrement and increment for $l\_yc$
  and $u\_yc$ in the $y$-direction;
$range\_of\_points \leftarrow \emptyset$;
for $x \leftarrow l\_xc$ to $u\_xc$ do
    for $y \leftarrow l\_yc$ to $u\_yc$ do
        add $[x, y]$ to $range\_of\_points$;
    end
end
$b \leftarrow |u\_yc - l\_yc|$, $a \leftarrow |u\_xc - l\_xc|$;
compute the line perpendicular (pink) to the
  hypotenuse (green) using the construction in 6
  and $a, b$;
$p1, p2 \leftarrow$ end-points of the line with intensity
  $< limit$;
$fp \leftarrow$ mid-point of $p1$ and $p2$;
return $fp$;
```

Algorithm 5: Find Maximum Perpendicular Point (Block E).

```
Input: $curr\_points$: List of points for which the
       ends of the path are to be calculated
Output: $points$: Coordinates of re-centered
        window
$contours \leftarrow$ get contour lines for $curr\_points$;
$end\_points, points \leftarrow \emptyset$;
for $lines \in contours$ do
    add end points of $lines$ to $end\_points$;
end
$hull \leftarrow$ convex hull generated from points in
  $end\_points$;
$group \leftarrow$ pairs of points from $hull$, each from
  consecutive contour lines, ensuring closest
  possible points;
for $g \in group$ do
    add the mid-point of points in $g$ to $points$;
end
return $points$;
```

Algorithm 6: Finding Junction Ends (Block F).

lies completely inside the assumed square and there is a difference in the intensities of the adjacent pixels for the contour lines. The coordinates of the end-points for each contour line in the outermost set is noted and re-arranged such that when drawn in the sequence, these points form a convex polygon. The coordinates are re-arranged by first finding the mean of the corresponding X and Y values in the coordinate system, also referred to as centroid. Then the polar angles of each of the points with respect to its centroid is compared and sorted. Figure 7 depicts contour lines for two cases when the Harris corner is detected and otherwise. In the left image, this method was used because a junction was found through the Harris corner detection process. The Harris coordinates are represented by yellow color. The colored dots at the end are the points computed for generating contour lines.

In each of the images depicted in the Figure 8, the circles indicate that among all the contour lines, the one which lies on the outermost boundary is selected. These are chosen to form a convex polygon, and are

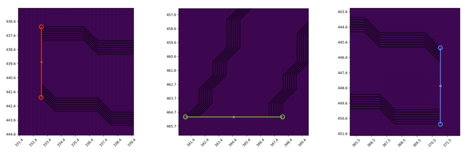

Figure 8: Pairing of end-points of consecutive contour lines in the outermost set of contours.

further paired to form corresponding paths. For example, in Figure 8, the pairs (upper-red, upper-blue), (lower-blue, right-green) and (left-green, lower-red) correspond to the end-points of the three contour lines. For each pair, the mean point is calculated and shown in Figure 7. Once the list of coordinates are available, the points that have crossed the boundary of the paths are excluded. The boundary points are just the lines connecting the end-points forming the pair. Care is taken to prevent from traversing the same path for more than once. The crossed boundary lines are found out by scanning the list of the last 5 traversed points and comparing their relative positions with each of the boundary lines. The boundary line is identified by zero crossing algorithm where the product of successive traversed coordinates emerges to negative, and the corresponding coordinates of the point and the direction is removed from the computed list.

3 EVALUATION

The advantages of this proposed *CoTSiC* method is the possibility of reconstructing images to any possible size. SSIM metric requires both original and reconstructed image of same size. Hence for different sizes, both images are first resized to a common dimension. This image is scaled down to the desired dimension and then supplied to the proposed method for extracting the reconstructed image. It was evident from experimental evaluation that as the common dimension of the signal image increases, the SSIM improves. The SSIM score for the original dimension is reported to be 92.14%, with a PSNR of 44.72 dB, which is acceptable (Setiadi, 2021).

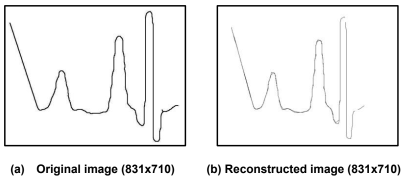

Figure 9: Input image vs reconstructed images.

Figure 10 (a) shows the input ECG signal, and the image of the reconstructed signal. The reconstructed image and the original image have the same dimensions exhibiting SSIM of 97.08%, and PSNR of 48.75 dB. Figure 11 shows the theta and delta components of the original EEG signal, and the image of the reconstructed signal respectively. Figure 10 (b) shows the original SpO2 signal with motion artifact, and its corresponding reconstructed signal. The reconstructed and the original image have the same dimensions exhibiting SSIM of 95.23%, and PSNR of 43.12 dB. Similarly, SPO$_2$ signal with no motion artifact generates SSIM of 97.3%, and PSNR of 46.53 dB. Table 1 compares the SSIM and PSNR values between images post JPEG compression and images formed after reconstruction from the data points generated by the proposed method for input images covering different physiological signals. Table 2 summarizes SSIM for all the investigated signals, and its compression benefits. The compression factor is estimated as the ratio of file size of original image and the corresponding reconstructed image. The compression factor in the range of 3.87× to 2.82× is characterized for the proposed *CoTSiC* method for four different signals with SSIM and PSNR of more than 92.10%, and 40 dB PSNR respectively.

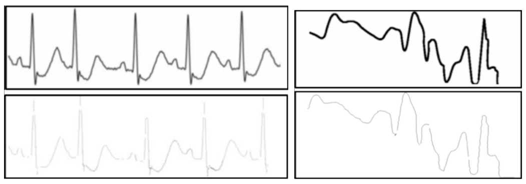

Figure 10: (a) ECG (left) and (b) SpO$_2$ (right) signals: input (top) reconstructed signals (bottom).

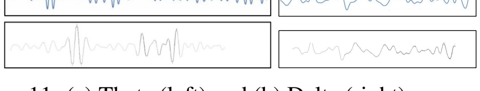

Figure 11: (a) Theta (left) and (b) Delta (right) components of an EEG Signal: input (top) reconstructed signal (bottom).

4 COMPLEXITY ANALYSIS

The complexity of image preparation process is expressed as:

$$\underbrace{O(M\times N\times k^2)}_{\text{Term 1}}+\underbrace{O(N^2)}_{\text{Term 2}}+\underbrace{O(W^2)}_{\text{Term 3}}+\underbrace{O(W^2)}_{\text{Term 4}}+\underbrace{O(W^2)}_{\text{Term 5}}$$

In this Equation , M and N are the input image dimensions and k is the blur kernel size in Term 1 and Term 2. k is a constant and thus can be ignored from the term. In Term 3, Term 4 and Term 5, W represents the window size which is used for estimating the data points. In Term 5, W^2 is required since all the points in the window are required for calculating the junction end-points. Following this, the equation can be written as:

$$\underbrace{O(M\times N)}_{\text{Term 1}}+\underbrace{O(N^2)}_{\text{Term 2}}+\,3\underbrace{O(W^2)}_{\text{Term 3}}$$

Thus, the complexity of the image pre-processing portion of the whole algorithm is $O(M\times N)$. For the recursive part of the algorithm, there are three branches possible. For each iteration, the cost function is stated in the Equation 2.

$$T(P_i)=\sum_{k}^{\{p_j\}}(O(W^2_{[P_i,P_k]})+T(P_k)) \qquad (2)$$

Here, a *path* is defined as the signal waveform along which the shifting window of dimension $W\times W$ pixels, traverses. This window estimates the next coordinates based on only that portion of the path which lies inside the window, thus leading to the greedy nature of the algorithm. $O(W^2_{[P_i,P_k]})$ represents the time complexity of these operations where P_i and P_k are the start and end points of the portion of path currently inside the window. $T(P_i))$ represents the time complexity corresponding for the path with P_i as the start point. For case len(coords = 1), $\{p_j\}=i+1$. The equation simplifies to $T([P_i,P_n])=2O(W^2_{[P_i,P_{i+1}]})+T(P_{i+1})$ For case len(coords = 0), the equation simplifies to $T([P_i,P_n])\approx 2O(W^2_{[P_i,P_{i+1}]})$ For case len(coords $>$ 1), $\{p_j\}=i+1$, the time complexity is further simplifies to $T(P_i)=\sum_k^{\{p_j\}}(O(W^2_{[P_i,P_k]})+T(P_k))$ Here $\{p_j\}$ represents the set of start points for the new branching paths. All these cases are referred to Figure 12.Complexity analysis of the overall recursion is done by considering the worst-case for the input image. This is where the physiological signal image contains as many peaks as possible and covers the most portion of the image frame. The closer the arms of the peaks (Figure 13) are, the more number of peaks are accommodated in the image. Experimentally, it was found that the minimum distance between

Table 1: SSIM and PSNR metrics comparison for different images post JPEG compression and those reconstructed from the proposed CoTSiC method.

Signal Generator	**Reconstructed Image**	**Resized Dimensions**	**SSIM (%)**	**PSNR (db)**
ECG (173 KB), 2100 × 2826	Proposed (56 KB), 2826 × 2826	2100 × 2826	97.10	48.41
		2826 × 2826	97.26	48.46
	JPEG Best (234KB), 2100 × 2826	2100 × 2826	99.99	71.38
	JPEG Least (100KB), 2100 × 2826	2100 × 2826	99.55	44.91
EEG (224KB), 1341 × 4626	Proposed (69KB), 1341 × 4626	1341 × 4626	95.41	57.31
		2313 × 2313	93.95	50.80
	JPEG Best (325KB), 1341 × 4626	1341 × 4626	98.53	44.25
	JPEG Least (121KB), 1341 × 4626	1341 × 4626	98.53	44.25
SpO2 (82KB) 294 × 1707	Proposed (11KB), 294 × 1707	2904 × 1707	81	36.90
	JPEG Best (82KB), 294 × 1707	294 × 1707	99.94	60.35
	JPEG Least (18KB), 294 × 1707	294 × 1707	98.02	39.90

Table 2: SSIM estimated for the CoTSiC method generated re-constructed physiological signals.

Signal	SSIM Score	PSNR	Compression Ratio (same dimensions)
Sample signal (Fig. 9)	92.14%	44.72dB	2.82
ECG	97.08%	48.75dB	3.39
Theta EEG	95.02%	58.8dB	3.5
Delta EEG	92.86%	40.71dB	3.87
SpO2 with Motion Artifact	95.23%	43.12dB	3.28
SpO2 with no Motion Artifact	97.3%	46.53dB	3.45

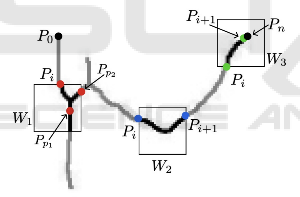

Figure 12: Schematic of physiological signal with windows W_1, W_2, and W_3 for the three $T([P_i, P_n])$ conditions stated above. P_0 and P_n are the start and end points.

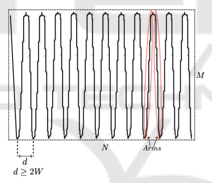

Figure 13: Worst case image input $(M \times N)$, where the peak height is the image height M.

the base of the arms of a peak should be at least $2 \times W$. This was tested with different line widths of the signal. Assuming an approximate straight arm and using simple trigonometry, for the worst case, the number of peaks and arm length is expressed as $n_{peaks} = N/d$, and $l_{arm} = M \times \sqrt{1 + \cancelto{0}{(W/M)^2}}$ respectively. Since $W \ll M$, the *path* length of the whole waveform is written as:

$$l_{path} = 2n_{peaks}l_{arm} \approx \frac{N}{d}(2M) = \frac{2MN}{d} = kMN \quad (3)$$

Grouping all constants, the number of new positions taken by the window while traversing the *path* is:

$$n_{opers} = l_{path}/W = (kMN)/W$$

Considering time complexity of window operations, the total time complexity is:

$$n_{opers} \cdot O(W^2) = \frac{kMN}{W} \cdot O(W^2) = O(MNW) \quad (4)$$

Considering W as a constant, the complexity becomes $O(M \times N)$ Hence the overall time complexity of the proposed method is deduced to $2O(M \times N) = O(M \times N)$, better than LASSO and Sparse Bayesian Learning algorithms Where M and N are the dimensions of the input image.

5 CONCLUSIONS

This study presents a novel signal compression scheme leveraging image processing to convert image(s) of signals from medical instrument displays into arrays of data points, requiring minimal additional setup for tele-healthcare. By avoiding machine learning algorithms, the method supports low-resource edge computing, making it viable for resource-constrained environments. Future work includes optimizing the algorithm by processing images in smaller blocks to reduce computational overhead and developing a mobile application compatible with a wide range of devices, enhancing accessibility and practicality for telemedicine.

REFERENCES

Cao, Z., Zhu, R., and Que, R.-Y. (2012). A wireless portable system with microsensors for monitoring respiratory diseases. *IEEE Transactions on Biomedical Engineering*, 59(11):3110–3116.

Chandrasekhar, V., Vazhayil, V., and Rao, M. (2020). Design of a real time portable low-cost multi-channel surface electromyography system to aid neuromuscular disorder and post stroke rehabilitation patients. In *2020 42nd Annual International Conference of the*

IEEE Engineering in Medicine & Biology Society (EMBC), pages 4138–4142.

Dabbaghian, A., Yousefi, T., Fatmi, S. Z., Shafia, P., and Kassiri, H. (2019). A 9.2-g fully-flexible wireless ambulatory eeg monitoring and diagnostics headband with analog motion artifact detection and compensation. *IEEE Transactions on Biomedical Circuits and Systems*, 13(6):1141–1151.

Doshi, R., Sankar, A. R., Nagaraj, K., Vazhayil, V., Nagaraj, C., and Rao, M. (2021). Eeg driven autonomous injection system for an epileptic neuroimaging application. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 1480–1486.

Elhosary, H., Zakhari, M. H., Elgammal, M. A., Abd El Ghany, M. A., Salama, K. N., and Mostafa, H. (2019). Low-power hardware implementation of a support vector machine training and classification for neural seizure detection. *IEEE Transactions on Biomedical Circuits and Systems*, 13(6):1324–1337.

Faisal, A. I., Mondal, T., Cowan, D., and Deen, M. J. (2022). Characterization of knee and gait features from a wearable tele-health monitoring system. *IEEE Sensors Journal*, 22(6):4741–4753.

Imtiaz, S. A., Iranmanesh, S., and Rodriguez-Villegas, E. (2019). A low power system with eeg data reduction for long-term epileptic seizures monitoring. *IEEE Access*, 7:71195–71208.

Jiang, W., Majumder, S., Kumar, S., Subramaniam, S., Li, X., Khedri, R., Mondal, T., Abolghasemian, M., Satia, I., and Deen, M. J. (2022). A wearable tele-health system towards monitoring covid-19 and chronic diseases. *IEEE Reviews in Biomedical Engineering*, 15:61–84.

Luo, X., Liu, D., Huai, S., Kong, H., Chen, H., and Liu, W. (2022). Designing efficient dnns via hardware-aware neural architecture search and beyond. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(6):1799–1812.

Ma, C., Wang, Z., Zhao, L., Long, X., Vullings, R., Aarts, R. M., Li, J., and Liu, C. (2023). Deep learning-based signal quality assessment in wearable ecg monitoring. In *2023 Computing in Cardiology (CinC)*, volume 50, pages 1–4.

Mamaghanian, H., Khaled, N., Atienza, D., and Vandergheynst, P. (2011). Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes. *IEEE Transactions on Biomedical Engineering*, 58(9):2456–2466.

Martínez, S., Veirano, F., Constandinou, T. G., and Silveira, F. (2023). Trends in volumetric-energy efficiency of implantable neurostimulators: A review from a circuits and systems perspective. *IEEE Transactions on Biomedical Circuits and Systems*, 17(1):2–20.

Molloy, A., Beaumont, K., Alyami, A., Kirimi, M., Hoare, D., Mirzai, N., Heidari, H., Mitra, S., Neale, S. L., and Mercer, J. R. (2022). Challenges to the development of the next generation of self-reporting cardiovascular implantable medical devices. *IEEE Reviews in Biomedical Engineering*, 15:260–272.

Nandi, P. and Rao, M. (2022). A novel cnn-lstm model based non-invasive cuff-less blood pressure estimation system. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 832–836.

Nia, A. M., Mozaffari-Kermani, M., Sur-Kolay, S., Raghunathan, A., and Jha, N. K. (2015). Energy-efficient long-term continuous personal health monitoring. *IEEE Transactions on Multi-Scale Computing Systems*, 1(2):85–98.

Nwibor, C., Haxha, S., Ali, M. M., Sakel, M., Haxha, A. R., Saunders, K., and Nabakooza, S. (2023). Remote health monitoring system for the estimation of blood pressure, heart rate, and blood oxygen saturation level. *IEEE Sensors Journal*, 23(5):5401–5411.

Ozkan, H., Ozhan, O., Karadana, Y., Gulcu, M., Macit, S., and Husain, F. (2020). A portable wearable tele-ecg monitoring system. *IEEE Transactions on Instrumentation and Measurement*, 69(1):173–182.

Penhaker, M. (2022). Biomedical engineering in the 21st century and in the future. In *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 000011–000012.

Qaisar, S., Bilal, R. M., Iqbal, W., Naureen, M., and Lee, S. (2013). Compressive sensing: From theory to applications, a survey. *Journal of Communications and Networks*, 15(5):443–456.

Rudelson, M. and Vershynin, R. (2006). Sparse reconstruction by convex relaxation: Fourier and gaussian measurements. In *2006 40th Annual Conference on Information Sciences and Systems*, pages 207–212.

Setiadi, D. R. I. M. (2021). Psnr vs ssim: imperceptibility quality assessment for image steganography. *Multimedia Tools and Applications*, 80:8423–8444.

Shaikh, M. R., Rao, M., and Subramaniam, G. (2023). A novel thermal imaging based transfer-learning model to estimate blood pressure. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5.

Shoaib, M., Lee, K. H., Jha, N. K., and Verma, N. (2014). A 0.6–107 µw energy-scalable processor for directly analyzing compressively-sensed eeg. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 61(4):1105–1118.

Spanò, E., Di Pascoli, S., and Iannaccone, G. (2016). Low-power wearable ecg monitoring system for multiple-patient remote monitoring. *IEEE Sensors Journal*, 16(13):5452–5462.

Wu, D., Li, S., Yang, J., and Sawan, M. (2022a). neuro2vec: Masked fourier spectrum prediction for neurophysiological representation learning.

Wu, D., Yang, J., and Sawan, M. (2022b). Bridging the gap between patient-specific and patient-independent seizure prediction via knowledge distillation. *Journal of Neural Engineering*, 19(3):036035.

Xue, J.-H. and Titterington, D. M. (2011). t -tests, f -tests and otsu's methods for image thresholding. *IEEE Transactions on Image Processing*, 20(8):2392–2396.