Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-018-0984-z

ORIGINAL RESEARCH

FPGA implementation of elephant recognition in infrared images to reduce the computational time

N. M. Siva Mangai[1] · P. Karthigaikumar[2] · Shilu Tresa Vinod[1] · D. Abraham Chandy[1]

Received: 4 December 2017 / Accepted: 11 August 2018

Abstract
Object recognition is a challenging task in image processing and computer vision. In this paper, segmentation, feature extraction and classification methods are done for elephant recognition. Thresholding based segmentation technique is used for image segmentation and k-NN classifier is used for object recognition based on the shape features of the segmented image. Infrared elephant images are considered for experimentation. The database created by us for this type of object recognition includes elephant, bear, horse, pig, tiger, and cow and lion images. The recognition rate is calculated for performance evaluation. However, implementing such algorithms on software consumes more time as image sizes and bit depths grow larger. Hence this paper aims at hardware implementation of elephant recognition to reduce the computational time. The proposed hardware is prototyped inVirtex-4 xc4vlx25 FPGA using Xilinx System Generator (XSG) tool. The hardware/software co-simulation feature allows the input and output to be displayed on Matlab window while the processing is done through FPGA. The results indicate that when the category is elephant and if the recognition status is "yes", recognition rate is 100%. If the category is not an elephant and if the recognition status is "no", recognition rate is still 100% also indicates, the approach is successful in elephant recognition and the computation of segmentation algorithm and shape feature extraction (area, centroid, equivdiameter) in hardware reduces the computational time of elephant recognition by 89.65% as compared to software computation.

Keywords Elephant recognition · Thresholding based segmentation · Feature extraction · K-NN classifier · FPGA implementation

1 Introduction

Asians and African elephants are the largest land animals, requiring large area for their habitat (Van Aarde et al. 2008; Santiapillai et al. 2010). Therefore, their living area are not officially confined to a particular territory because of its complexity.

Elephants are in threat because of poaching and deforestation by humans for settlement, industrial purpose (Douglas-Hamilton 2008; Lemieux and Clarke 2009) and habitat loss.

India is native to 60% of Asian elephant population, most of which lives near to human settlements in mountainous forests. Southern India hosts half of its population numbering to 6300 elephants (Kumara et al. 2012). Population increase of humans leading to more agricultural and industrial activities decreases the needs of elephants such as grassing area and acute shortage of water. Hence, there has been severe human–elephant conflict (HEC) which threatens both of their livings. As elephants are in search of food entering agricultural fields and humans in turn agitating creates huge conflict (Hoare and Du Toit 1999; Pinter-Wolman 2012). This happens especially in the vicinity of human inhabitation near forests (Sugumar and Jayaparvathy 2013). To track these groups of elephants becomes difficult because of their size and changing behaviour.

By the growth of the technology and advanced living world, a lot of research has been performed on the visual analysis of human beings and human-related events. The automated analysis of elephants has been widely used in the area of object recognition (Zeppelzauer 2013). Elephant recognition in images is an interesting and one of the difficult

✉ P. Karthigaikumar
p.karthigaikumar@gmail.com

[1] Department of ECE, Karunya University, Coimbatore, Tamilnadu, India

[2] Department of ECE, Karpagam College of Engineering, Coimbatore, Tamilnadu, India

tasks in object recognition, which can play an important role to solve the human–elephant conflict. Animals are among the most difficult objects for classification and recognition. An object recognition system very much depends on segmentation, feature extraction and decision making process (Kamavisdar et al. 2013). Generally, in monitoring applications the camera is usually fixed and the background is mostly static. In such a scenario we can easily learn a background model and identify objects of interest by detecting changes to the background.

The main significance of this work is image datasets containing elephants in different poses, sizes, in groups or as individuals. The detection of elephants is especially hard because their skin does not exhibit a salient texture pattern and thus lacks in distinctive visual features (Ardovini et al. 2008). Infrared (IR) images are used because during the night time the object cannot be seen through the nacked eye. So, the image perception is based on the object (animal) temperature. The object may be either hot or cold, which depends upon the objects internal and external temperatures. Brightness of the infrared image depends on temperature of object. If it is cold, less brightness in the IR image, otherwise more brightness in the IR images. For this purpose variety of IR cameras are available to capture the object image. The captured infrared images are processed to detect any presence of the object in the field of view. The object is separated from the background and is then processed and compared with the specific reference feature set (Suen et al. 1998). If the object is determined to be an elephant, then the recognition status is ‘yes’, otherwise ‘no’.

Implementation of the image processing algorithm in a general purpose processors (GPPs) is inefficient as it is not been able to keep up with the pace at which data are growing. Real-time image processing is difficult to achieve on a serial processor (Chikkali and Prabhushetty 2011). This is due to several factors such as the large data set represented by the image and the complex operations which may need to be performed on the image (Hwang et al. 2010). One of the methods to accelerate data analysis and to overcome the limitation of GPPs is the use of hardware in the form of field programmable gate arrays (FPGAs). FPGAs are often used as implementation platforms for image processing applications because their structure is able to exploit spatial and temporal parallelism. Hence, the objective of this paper is to implement the elephant recognition algorithm in FPGA to reduce the computation time. But however it cannot be used as a standalone model as image pre-processing for FPGA implementation is done using CPU.

The rest of this paper is organised as follows. Section 2 describes the methodology used for elephant recognition. Section 3 presents the implementation process of elephant recognition in FPGA, which reduces the computational time as compared with the software. Section 4 deals with the experimental results and discussions and Sect. 5 lists the conclusion and future work.

1.1 Related works

There are many techniques for automatic detection of elephants such as satellite tracking using global positioning system (GPS) (Douglas-Hamilton et al. 2005) and systems based on vibration sensors in the ground (Sugumar and Jayaparvathy 2013) which are not applicable for bigger population and are costly too. Today, techniques such as acoustic and visual monitoring are used to sample populations and to obtain reliable estimates of species presence and, potentially, abundance which are cost and detection effective (Wrege et al. 2010; Blumstein et al. 2011). Dabarera and Rodrigo (2010) proposed appearance based recognition algorithms for identification of elephants. By providing the front face image of an elephant the system gives the result based on vision algorithm whether the elephant is a new one or already recognized one (Goswami et al. 2012) by comparing the tusk, ear lobe parameters of a captured elephant can calculate its population. But in reality its difficult to capture the front face of an elephant.

Sugumar and Jayaparvathy (2013) have given an analytical model for surveillance and tracking of elephant herds using a three-state Markov chain. This study was conducted in the dense forest region of western ghats in Coimbatore region. Sugumar and Jayaparvathy (2014) using a real time imaging they presented the intrusion of elephants in the border (Zeppelzauer and Stoeger 2015) acoustic and visual domain were used to detect the elephants. Visual detectors can detect even during day time and do not require vocalization of elephants. There are various methods available to detect object and identify it in the field of image processing (Kumar and Parameswaran 2013). The earlier works in (Ramanan et al. 2006; Hannuna et al. 2005; Lahiri et al. 2011) specifies one skin feature in detecting wild animals in forest. The skin texture helps in identifying the animal easily. One feature alone will not help rather both color and skin texture helps in the identification of different elephants (Sugumar and Jayaparvathy 2014, Wang et al. 2011). In image processing after the filtering is done, various edge detection mechanisms could be used in image processing for the identification of the desired objects. Various of these mechanisms are explored by Muthukrishnan and Radha (2011). Mechanisms like edge based, sobel, canny,roberts, prewitt methods specifically for elephants were discussed (Shaukin 2015).

Image segmentation was found to be the best one for recognising an elephant from the background. As this method process only with image, it will not affect both elephant and human beigns. It can be calssified as thresholding (Suen et al. 1998), template matching, region growing, edge detection (Chikkali and Prabhushetty 2011) and clustering.

Clustering technique has been employed in various fields among which *k*-means clustering is widely used (Hussain et al. 2012). *K*-NN classifier (Hussain and Seker 2012) the most trustworthy and with less errors is used for classification. Mangai et al. (2015) implemented a clustering-based image segmentation approach for elephant recognition method. The experiment and analysis is done using 23 different IR images. All those images gave the proper segmentation results using *k*-means clustering technique with $k=2$. Shape features are extracted using the *K*-NN classifier. Elephants are accurately detected for $K=3$ and 5.

Many automated systems which are the result of scientific advancement are discussed: the authors (Arivazhagan and Ramakrishnan 2010) presented a paper on the conservation of elephants with the focus on connectivity of elephant's habitat. The study was done in southern part of the Indian sub-continent. Olivier et al. (2009) discussed a method to detect the elephants using the dung. With the dung decay rates and distance sampling techniques, they have detected and estimated the population size, age group in Southern Mozambique region. Fernando et al. (2012) proposed a solution by direct observations to avoid conflict. Balch et al. (2006) proposed RFID system for tracking. It has a drawback of short range detection and it has low update rate of the locations.

Prabu (2016) used seismic sensors for detection. Recently, automated vision-based method was proposed to detect elephants (Ramesh et al. 2017).

2 Methodology

The block diagram shown in Fig. 1 depicts the various stages of automated elephant recognition and the detailed description of each one is given in this section.

The input images are infrared images that may contain different animals with static background. The animals may be single or in group doing different actions like eating, moving, etc. The conversion of infrared (IR) image into grayscale image is accomplished in the pre

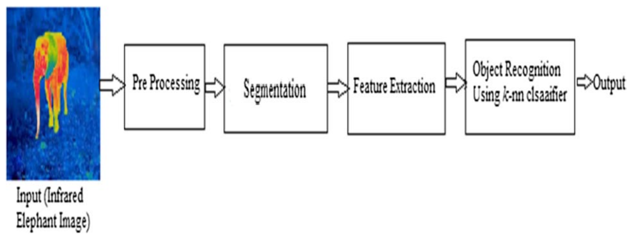

Pre Processing
Segmentation
Feature Extraction
Object Recognition Using k-nn clsaaifier
Output
Input (Infrared Elephant Image)

Fig. 1 Various stages of automated elephant recognition

processing stage. Subsequently, thresholding based segmentation method is used to segment the object of interest. Next, different shape based feature are extracted from the detected object. Finally, object recognition is done using *k*-NN classifier based on the extracted features. The detailed description of each stage is given in the following subsections.

2.1 Image segmentation by thresholding

First step of object recognition is segmenting the image, so that the pixels of same characteristics belong to one group. Whole pixels are arranged into distinct groups at the end of segmentation It has been extensively used in many areas, such as pattern recognition, machine learning and statistics. The segmentation algorithms used in our methodology is thresholding based segmentation. The objective of image binarization is to divide an image into two groups, foreground or object, and background. In image processing applications, the gray level values assigned to an object are different from the gray level values of the background. Therefore thresholding can be considered as an effective way to separate foreground and background. The output of a thresholding process is a binary image which is obtained by assigning pixels with values less than the threshold with zeros and the remaining pixels with ones.

Let us consider image f of size $M \times N$ (M rows and N columns) with L gray levels in the range $[0, L-1]$. The gray level or the brightness of a pixel with coordinates (i, j) is denoted by $f(i, j)$. The threshold, T, is a value in the range of $[0, L-1]$. The thresholding technique determines an optimum value for T from the histogram of the input image, so that:

$$g(i,j) = \quad 0 \text{ for } f(i,j) < T \quad 1 \text{ for } f(i,j) > T \tag{1}$$

The data's available may not be useful in their collected form unless they are computationally processed to extract meaningful results. For the application of object recognition, features are to be extracted from the segmented image. These features will determine in which class the object belongs to. The classification is done using K-NN classifier (Hussain and Seker 2012) which is the most robust to noisy training data and accurate classification method. It is effective when the training data is large. A storage database having the characteristics of the particular object searching for is maintained and the segmented image is compared with the database stored. If the segmented image matches the characteristics stored in the database, the output for the object recognition is positive, else it is negative. Positive represents that the object is present in the image and negative, if the object is not present.

2.2 Feature extraction

The various shape features extracted from the segmented object(s) in the work are major axis length, minor axis length, eccentricity, orientation, convex area, equivdiameter, solidity, extend, area and centroid. The detailed description of the same is given below.

- *Major axis length* The longest line that can be drawn through the object connecting the two farthest points in the boundary is called as major axis. If the major axis end points are $(x_{1,} y_{1)}$ and $(x_{2,} y_2)$, then the major axis length is calculated as,

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

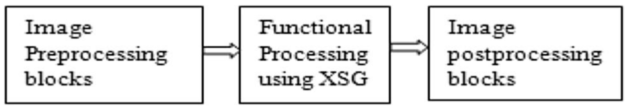


Image Preprocessing blocks
Functional Processing using XSG
Image postprocessing blocks

Fig. 2 Various stages of FPGA implementation of image processing using XSG

- *Minor axis length* The object width can be calculated using the minor axis. Minor axis is the line that can be drawn through the object while maintaining perpendicularity with the major axis
- *Eccentricity* This feature determines the elongatedness of the object. The eccentricity is the ratio of the distance between the foci of the object 'C' and its major axis length 'D', which is defined as,

$$\text{Eccentricity} = \frac{C}{D} \quad (3)$$

- *Orientation* The orientation the object is determined by using the angle between major axis *x*-axis of the image described as,

$$\tan^{-1}\left(\frac{(y_2 - y_1)}{(x_2 - x_1)}\right) \quad (4)$$

- *Convex area* The number of pixels in the convex hull (smallest polygon that encompasses) the object.
- *Equivdiameter* It specifies the diameter of a circle with same area as the detected object, which is computed as,

$$\text{Equiv diameter} = \sqrt{\frac{4 \times \text{area of an object}}{\pi}} \quad (5)$$

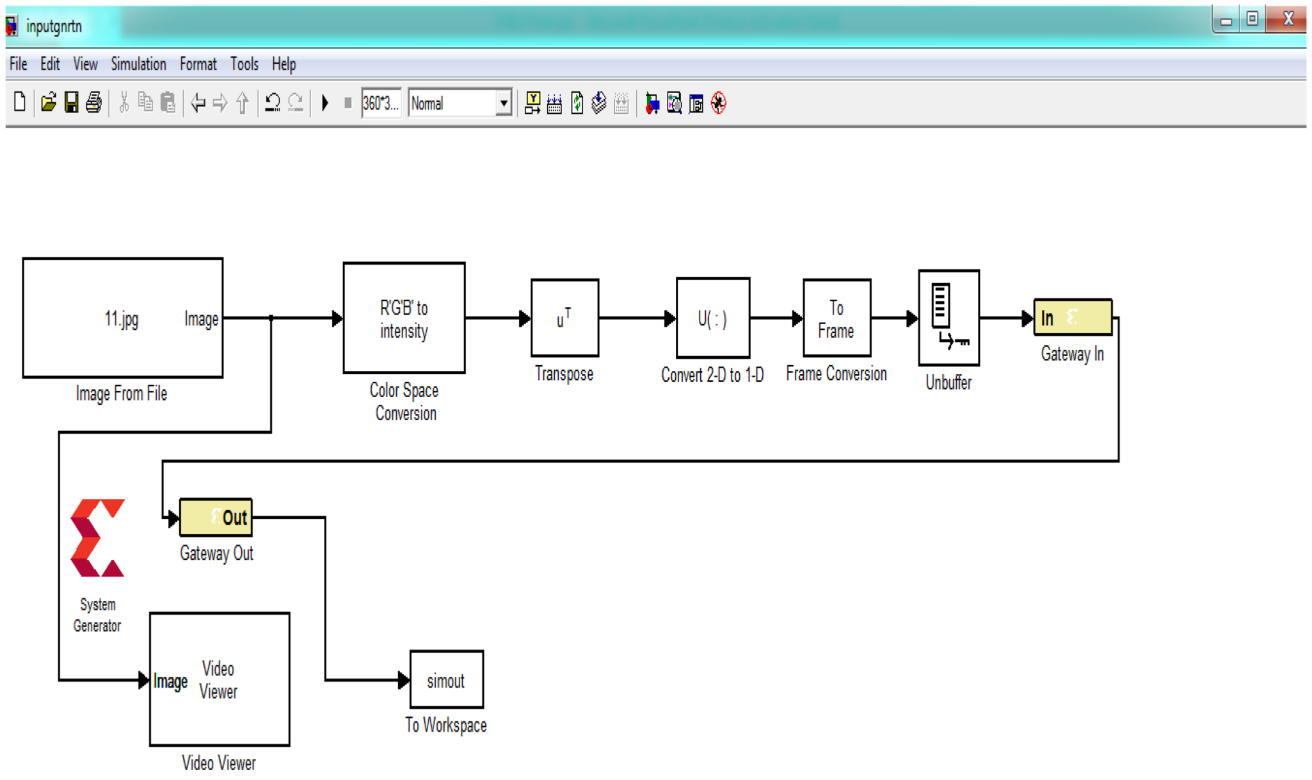


Fig. 3 Simulink blocks for image pre-processing

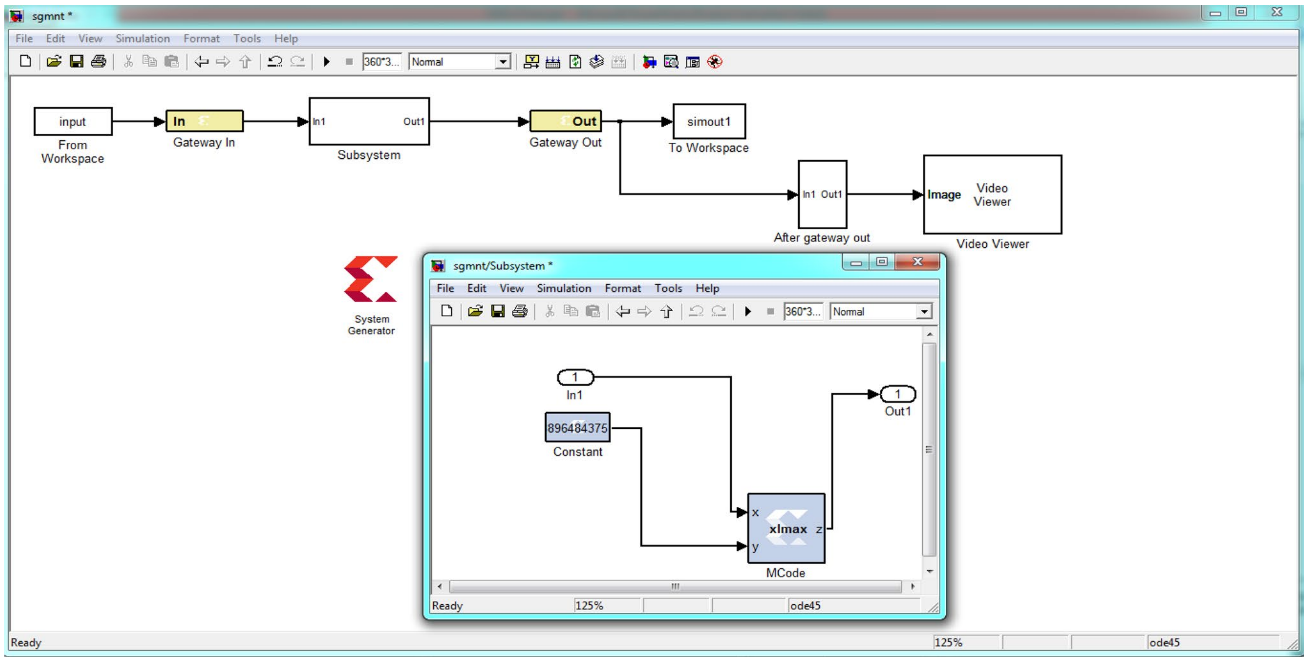


Fig. 4 XSG blocks for image segmentation using thresholding

where, area is the total number of pixels of the object.

- *Solidity* The ratio of the area of the object to the convex area given as,

$$\text{Solidity} = \frac{\text{area}}{\text{convex area}} \tag{6}$$

- *Extent* The ratio of the total number of pixels in the object region to the pixels in the minimum bounding box (smallest rectangle that enclose the object). It is expressed as,

$$\text{Extent} = \frac{\text{area}}{\text{area of the minimum bounding box}} \tag{7}$$

- *Area* is a scalar; the actual number of pixels in the region.
- *Centroid* is a scalar that specifies the centre of mass of the region. The first element of Centroid is the horizontal coordinate (or x-coordinate) of the centre of mass, and the second element is the vertical coordinate (or y-coordinate).

It is to be noted that the above features for the segmented object in our work are computed using the matlab command '*regionprops*'.

2.3 Recognition

Object recognition identifies the object from a set of known labels. The recognition analysis is done with the help of *k-NN* classifier. Animals like elephant, pig, deer, lion, tiger, cow, bear, horse and dog are classified into various classes like class 1, class 2, class 3, class 4, class 5, class 6, class 7, class 8 and class 9 respectively. If the object belongs to class 1, then the recognition will be "detected object is elephant". If not "detected object is not an elephant".

The algorithm for *K*-nn classifier is as follows:

- *Step 1* Calculate the distance between the feature set extracted from the detected object and the feature dataset, i.e., training vectors using Euclidean distance measure.
- *Step 2* Select k closest training sets based on the calculated distance measures
- *Step 3* Recognize the object based on the class of the majority training set selected.

3 FPGA implementation

The need to process the image with less computational time, lead to the implementation of image processing algorithm in hardware level, which offers parallelism and thus significantly reduces the processing time (Saegusa and Maruyama 2007; Hussain et al. 2012). The drawback of most of the

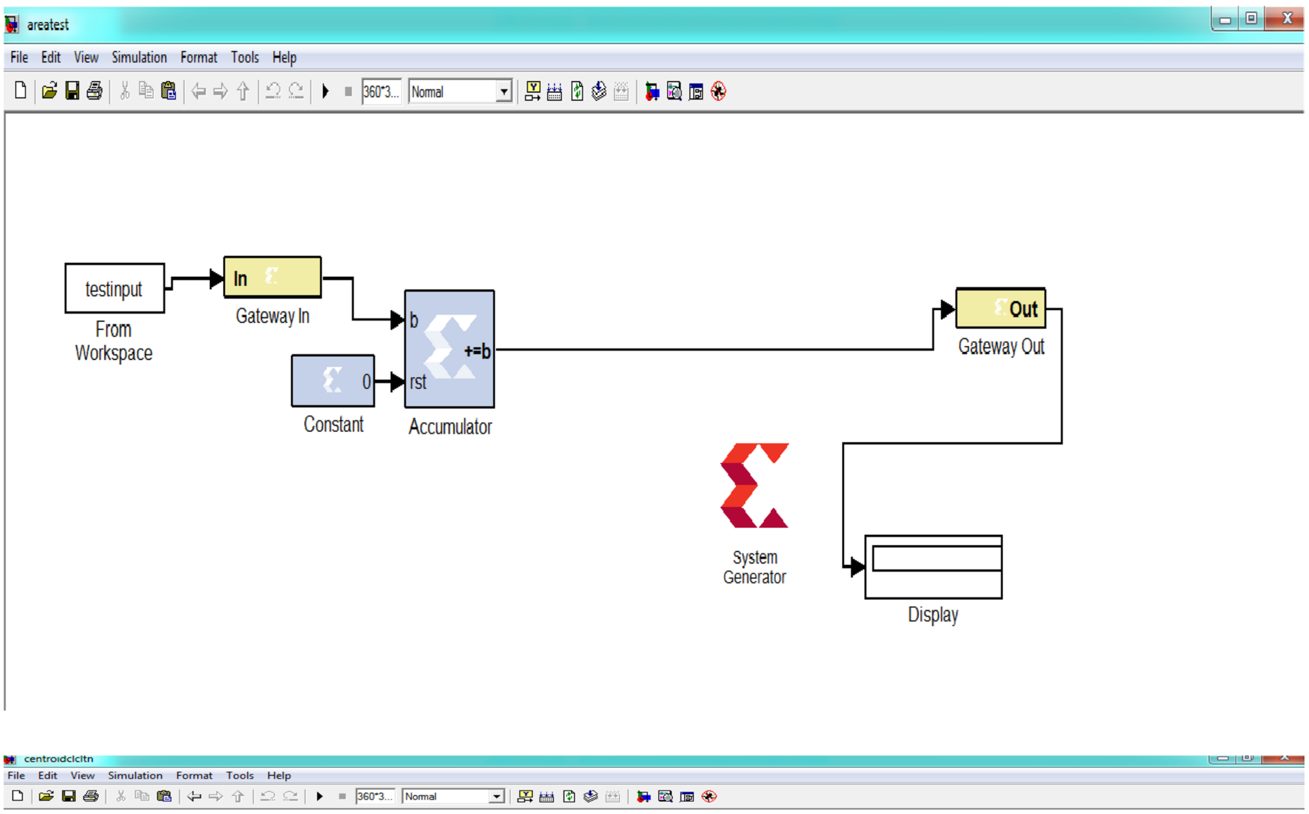


Fig. 5 XSG blocks for area feature extraction

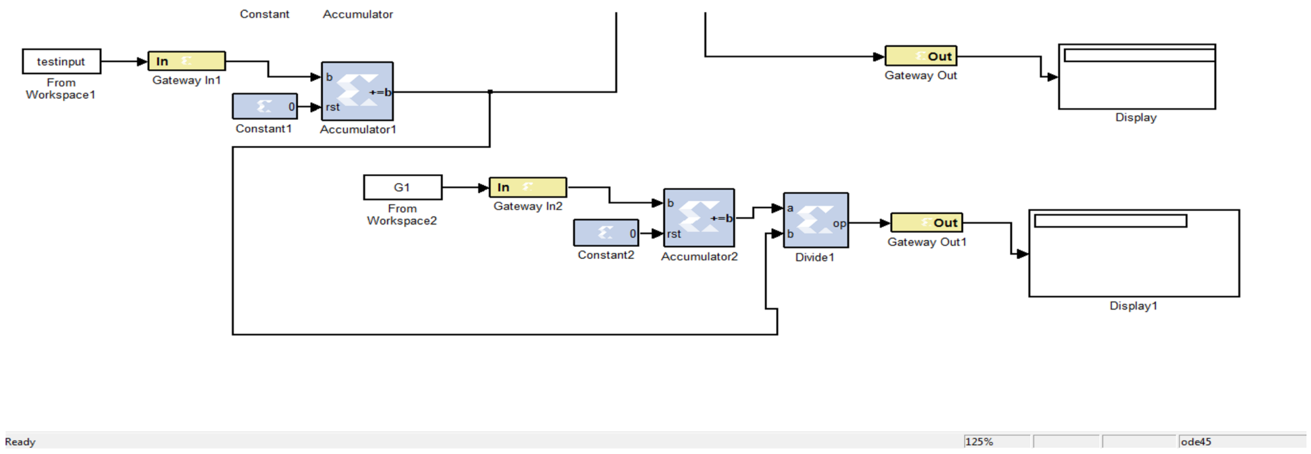


Fig. 6 XSG blocks for centroid feature extraction

methods is that they use a high level language for coding. This objective lead to the use of Xilinx System Generator, a tool with a high level graphical interface under the Matlab, Simulink based blocks which makes it very easy to handle with respect to other software for hardware description (Christe et al. 2011). Xilinx System Generator is an extension of Simulink and consists of models called 'Xilinx Blocks' which are mapped into architectures, entities, signs, ports and attributes which scripts file to produce synthesis in FPGAs, HDL simulation and development tools (Ladgham et al. 2012).

Figure 2 shows the entire operation of FPGA implementation of image processing using Simulink and Xilinx blocks. It goes through three phases as:

- Image pre-processing blocks

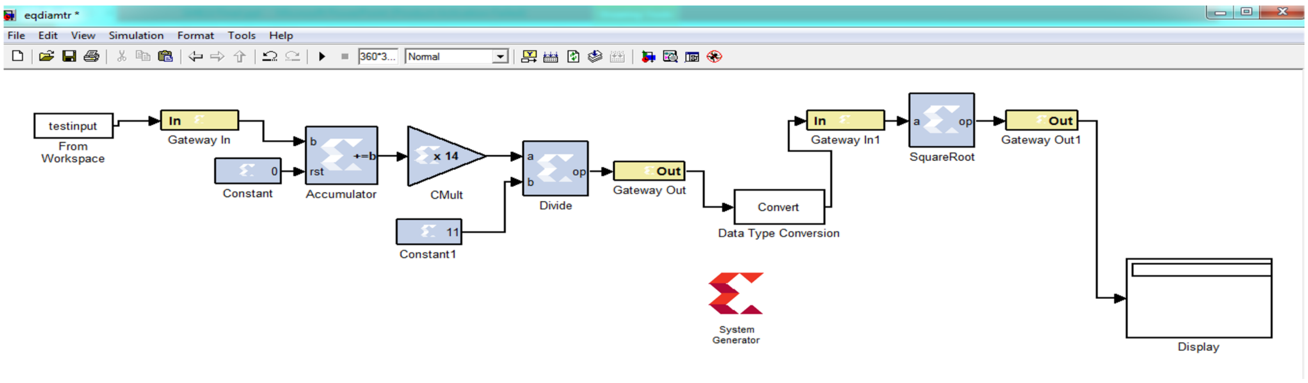


Fig. 7 XSG blocks for Equiv diameter feature extraction

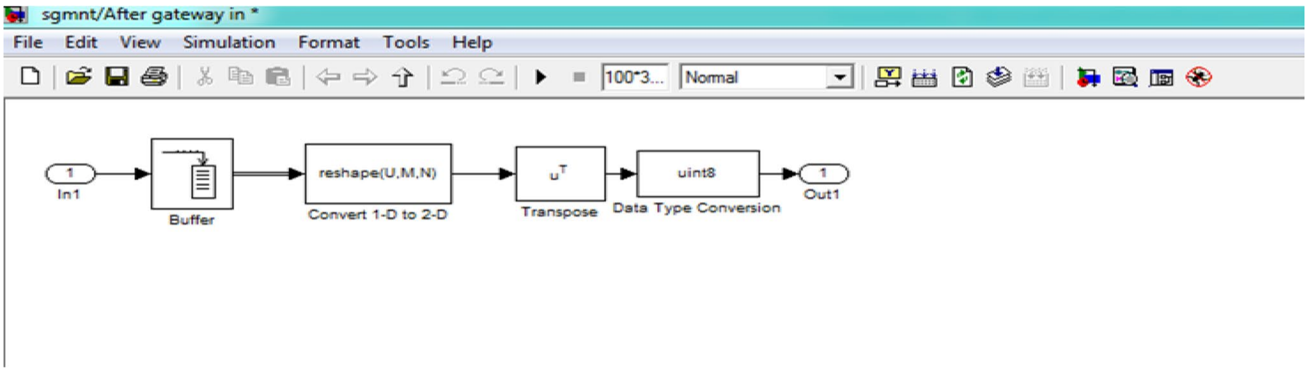


Fig. 8 Simulink blocks for image post-processing

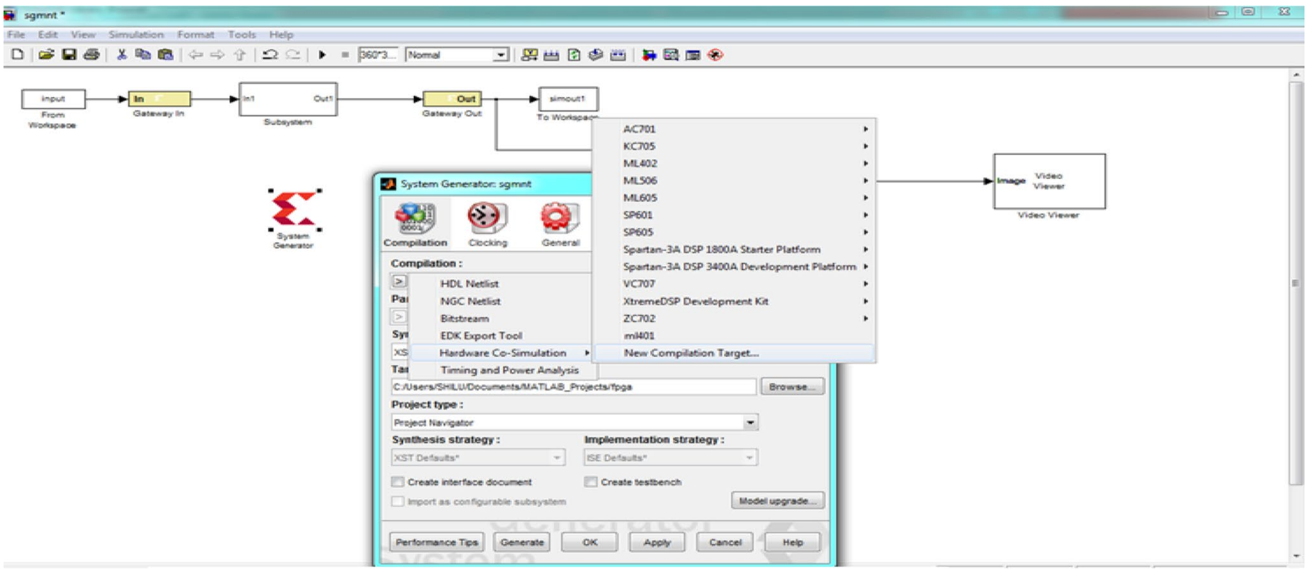
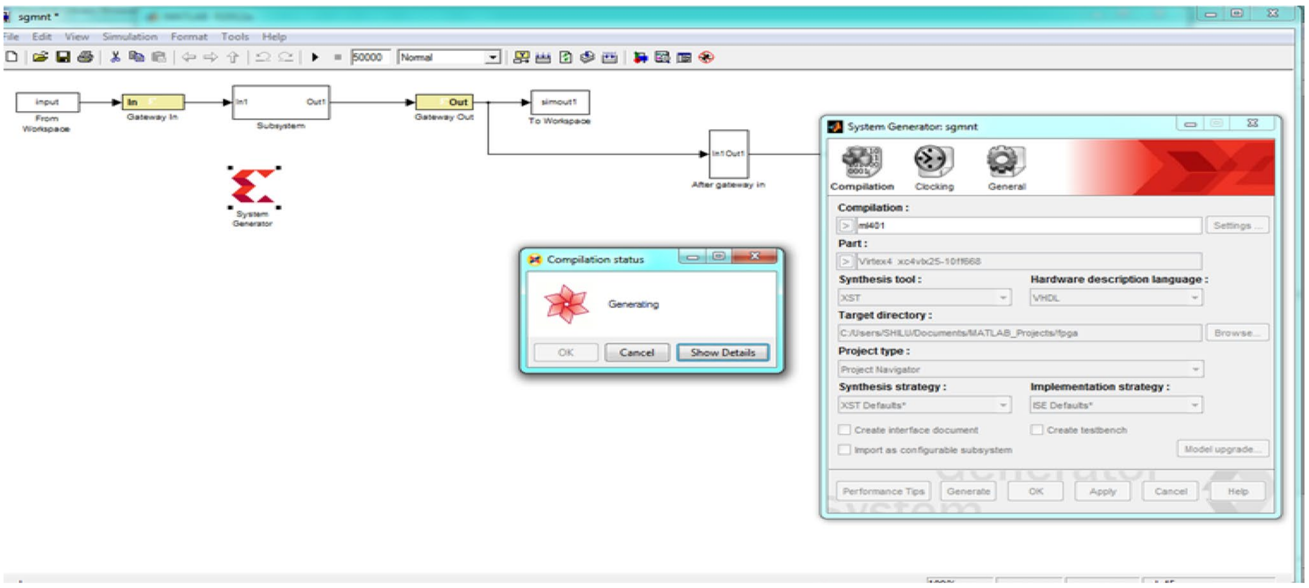
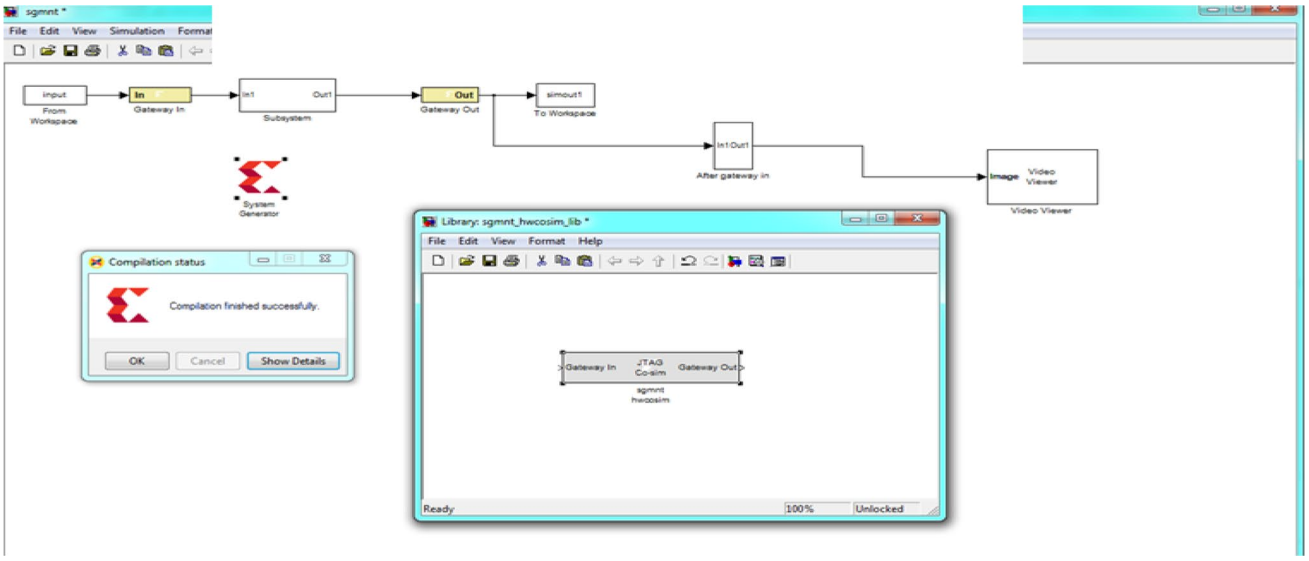


Fig. 9 Hardware co-simulation—compiling a new board



Compilation : ml401
Part : Virtex4 xc4vlx25-10ff668
Synthesis tool : XST
Hardware description language : VHDL
Target directory : C:/Users/SHILU/Documents/MATLAB_Projects/fpga
Project type : Project Navigator
Synthesis strategy : XST Defaults*
Implementation strategy : ISE Defaults*
Create interface document
Create testbench
Import as configurable subsystem

Fig. 10 Building the design netlist



Compilation finished successfully.

Fig. 11 Netlist building complete

- Functional processing using XSG
- Image post-processing blocks

3.1 Image pre-processing block-sets

In the software level simulation using Simulink block-sets alone, where the image is used as a two-dimensional (2D) arrangement such as M×N, there is no need for any image pre-processing, but at hardware level this matrix must be an array of one dimension(1D), namely a vector, where it requires image pre-processing. Simulink blocks for image pre-processing are shown in Fig. 3. All the image pre-processing is done on a CPU.

Moreover, the methods require considerable processing. Although these methods have good performance, they are not generally suitable for hardware implementation. The basic requirement for the thresholding method is its adaptability and efficiency. It should also have the least dependency on image pre-processing. A Xilinx block for thresholding technique is given in Fig. 4.

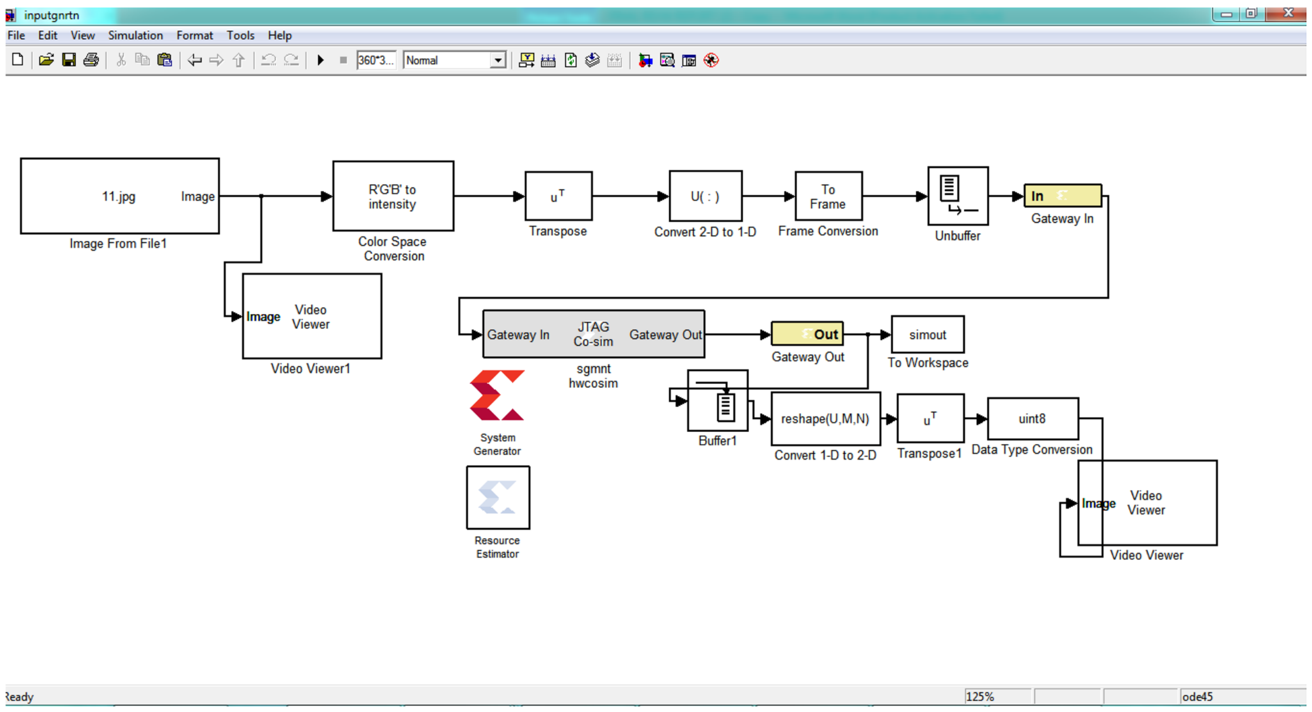


Fig. 12 Modified design for hardware co-simulation

Table 1 Experimental datasets

Category	Classes	No. of images
Elephant	1	7
Pig	2	1
Deer	3	3
Lion	4	1
Tiger	5	1
Cow	6	3
Bear	7	1
Horse	8	5
Dog	9	2

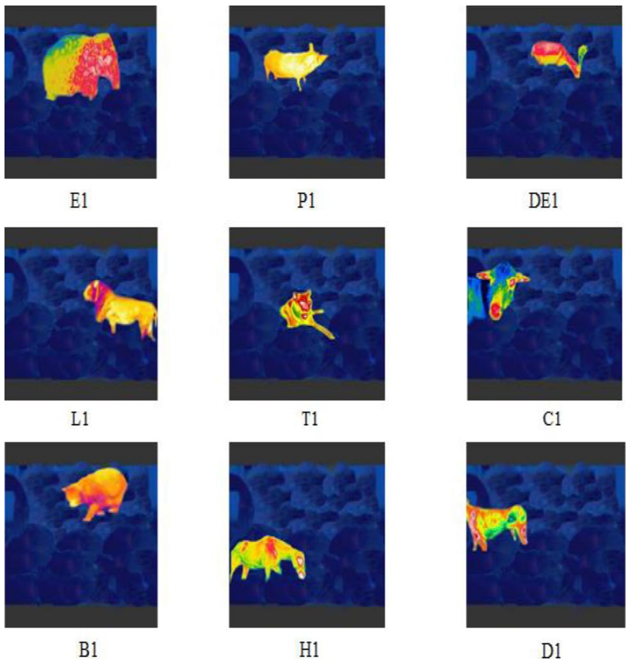


Fig. 13 Sample infrared images

3.2 Functional processing using XSG

System Generator is a DSP design tool from Xilinx that enables the use of the Math Works model-based Simulink design environment for FPGA design. Xilinx blocksets for image segmentation and thereby feature extraction are made for generating corresponding HDL code.

3.2.1 Image segmentation by thresholding using XSG

For the hardware implementation, the effectiveness of a thresholding method is considered in terms of parameters such as speed and complexity. These become very important in hardware image processing applications. All of the high-ranked cluster based techniques have to compute some image features, such as maximum/minimum gray level values, or variance of image, before the segmentation. Therefore an image must be pre-processed pixel by pixel. For

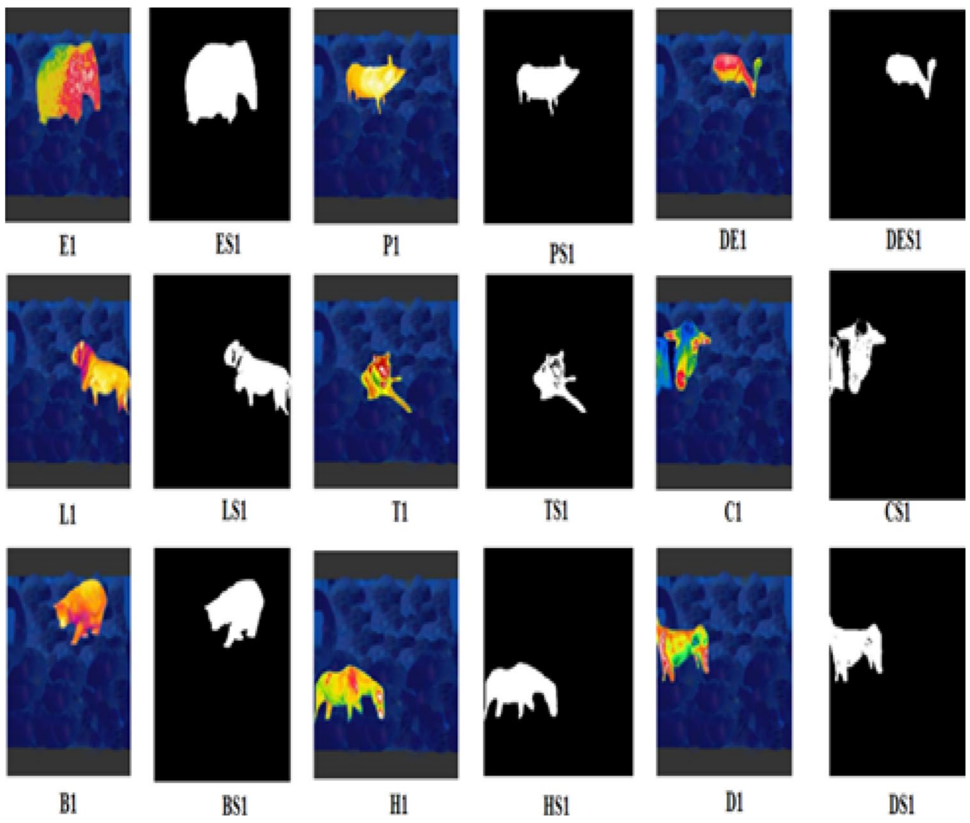


Fig. 14 Segmentation output for software processing

Table 2 Sample feature set-software

Sample images	Class	Major axis	Minor axis	Eccentricity	Orientation	Convex area	Equiv diameter	Solidity	Extent
E1	1	11.4502	7.7403	0.7369	– 29.499	68.000	8.8129	0.8971	0.6162
P1	2	0.1474	0.0658	0.0009	0.0044	9.8370	0.0927	0.0007	0.0005
DE1	3	0.1425	0.0644	0.0009	– 0.0212	7.5290	0.0793	0.0007	0.0004
L1	4	0.1504	0.0715	0.0009	– 0.0111	9.8770	0.0961	0.0007	0.0005
T1	5	0.1280	0.0765	0.0008	– 0.0270	7.9940	0.0713	0.0005	0.0003
C1	6	9.1936	3.4593	0.9265	– 10.683	25.000	5.4115	0.9200	0.6389
B1	7	0.0218	0.0122	0.0001	0.0009	2.1935	0.0147	0.0001	0.0001
H1	8	27.2617	7.2092	0.9644	– 75.656	161.000	13.6343	0.9068	0.5615
D1	9	4.0000	4.0000	0.0000	0.0000	12.0000	3.9088	1.0000	0.7500

these methods a large processing overhead is present. In these techniques complex computational processes, such as logarithms, are also required.

3.2.2 Feature extraction using XSG: area

Area is a scalar that gives the actual number of on pixels in the image. Figure 5. gives the Xilinx blocks for extracting area of the object in the binary image.

3.2.3 Feature extraction using XSG: centroid

Centroid is a scalar that specifies the centre of mass of the region. The first element of Centroid is the horizontal coordinate (or x-coordinate) of the centre of mass, and the second element is the vertical coordinate (or y-coordinate). Xilinx blocks for extracting the centroid of the object in the image is given in Fig. 6.

3.2.4 Feature extraction using XSG: Equiv

3.2.4.1 Diameter Equiv diameter is the scalar that specifies the diameter of a circle with the same area as the region.

Table 3 Recognition results software

Database	Recognition status
1	Yes
2	Yes
3	Yes
4	Yes
5	Yes
6	Yes
7	Yes
8	No
9	No
10	No
11	No
12	No
13	No
14	No
15	No
16	No
17	No
18	Yes
19	No
20	No
21	No
22	No
23	Yes
24	No

Table 4 Verification results-software

IR database images	Category	Recognition status
1	Elephant	Yes
2	Elephant	Yes
3	Elephant	Yes
4	Elephant	Yes
5	Elephant	Yes
6	Elephant	Yes
7	Elephant	Yes
8	Pig	No
9	Deer	No
10	Deer	No
11	Deer	No
12	Lion	No
13	Tiger	No
14	Cow	No
15	Cow	No
16	Cow	No
17	Bear	No
18	Horse	Yes
19	Horse	No
20	Horse	No
21	Horse	No
22	Horse	No
23	Dog	Yes
24	Dog	No

Computed as sqrt (4 × area/pi). This property is supported only for 2-D input label matrices. Figure 7. gives the Xilinx blocks for extracting the equiv diameter of the object in the image.

3.3 Image post-processing block-sets

For post-processing it uses a Buffer block which converts scalar samples to frame output at lower sampling rate, followed by a 1D to 2D (matrix) format signal block, finally a sink is used to display the output image back in the monitor, utilizing the Simulink block-sets. Image post-processing block-sets are shown in Fig. 8.

Table 5 Class wise recognition rate software

Class\category	Recognition rate (%)
1\Elephant	100
2\Pig	100
3\Deer	100
4\Lion	100
5\Tiger	100
6\Cow	100
7\Bear	100
8\Horse	80
9\Dog	50

3.4 Hardware-software co-simulation

Usually several issues may arise when the model is transformed into hardware. System generator provides several methods to transform the models built using Simulink into hardware. One of these methods is called hardware co-simulation.

Hardware co-simulation enables building a hardware version of the model and using the flexible simulation environment of Simulink we can perform several tests to verify the functionality of the system in hardware. Hardware co-simulation supports FPGAs from Xilinx on boards that support JTAG or ethernet connectivity. Steps involved are:

Table 6 Recognition rates with different K values

Class\category	K = 3 (R%)	K = 5 (R%)	K = 7 (R%)
1\Elephant	100	100	85.71
2\Pig	100	0	100
3\Deer	100	100	100
4\Lion	100	100	100
5\Tiger	100	100	100
6\Cow	100	100	100
7\Bear	100	100	100
8\Horse	80	60	60
9\Dog	0	0	100
Average (%)	86.66	73.33	93.96

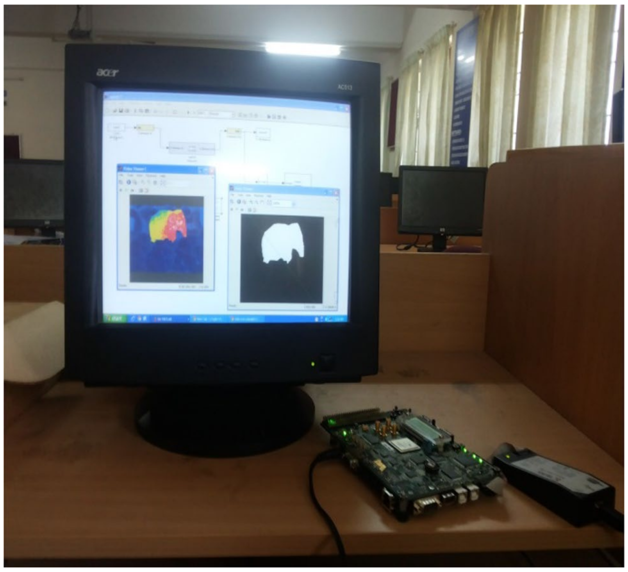


Fig. 15 Hardware co-simulation for segmentation using XSG

- Configure system generator for hardware co-simulation—compiling a new board as shown in Fig. 9.
- Generation of the hardware co-simulation block is shown in Fig. 10
- Hardware co-simulation block and synthesis report generation is shown in Fig. 11.
- Modifying the design for hardware co-simulation is given in Fig. 12.

Advantage of this methodology is that, it does not require HDL code to be written.

Table 7 Sample feature set-hardware

Sample images	Class	Area	Centroid	Equiv diameter
E1	1	18,286	181.38 125.512	152.55
P1	2	6119	148.498 121.474	88.248
DE1	3	4023	211.6809 107.1570	71.55
L1	4	8248	277.4975 171.1372	102.457
T1	5	3369	169.8070 180.0887	65.4814
C1	6	6295	63.139 136.647	89.508
B1	7	8727	220.977 96.5658	105.39
H1	8	8177	86.1721 222.6131	102.015
D1	9	6836	67.1623 158.2254	93.275

Table 8 Recognition results-hardware

Database	Recognition status
1	Yes
2	Yes
3	Yes
4	Yes
5	Yes
6	Yes
7	Yes
8	No
9	No
10	No
11	No
12	Yes
13	No
14	Yes
15	No
16	No
17	No
18	No
19	No
20	No
21	No
22	No
23	No
24	No

Table 9 Verification results-hardware

Database	Category	Recognition status
1	Elephant	Yes
2	Elephant	Yes
3	Elephant	Yes
4	Elephant	Yes
5	Elephant	Yes
6	Elephant	Yes
7	Elephant	Yes
8	Pig	No
9	Deer	No
10	Deer	No
11	Deer	No
12	Lion	Yes
13	Tiger	No
14	Cow	Yes
15	Cow	No
16	Cow	No
17	Bear	No
18	Horse	No
19	Horse	No
20	Horse	No
21	Horse	No
22	Horse	No
23	Dog	No
24	Dog	No

Table 10 Class wise recognition rates-hardware

Class\category	Recognition rate (%)
1\Elephant	100
2\Pig	100
3\Deer	100
4\Lion	0
5\Tiger	100
6\Cow	66.66
7\Bear	100
8\Horse	100
9\Dog	100

4 Results and discussion

In this section, recognition performance for various animals like elephant, bear, horse, pig, tiger, deer, cow and lion are presented and discussed. The outputs at segmentation, feature extraction and recognition in both software and hardware processing are presented in detail. The device utilization summary of hardware implementation and the delay comparison of computation in software and hardware are discussed in detail. Software implementation results are obtained using Matlab R2012a and hardware implementation results using Matlab Simulink R2012a and Xilinx System Generator using Xilinx ISE Design Suits 14.4 and prototyped in Virtex-4 FPGA board. 4.1 experimental datasets.

Input infrared images for the experiment include objects in a fixed background. The database include infrared images of nine classes of elephant, pig, deer, lion, tiger, cow, bear, dog and horse. Detailed information about the number of images in each class is tabulated in Table 1. Fig. 13 shows the sample infrared images from each class.

The datasets are categorised into classes from 1 to 9. 24 images are chosen as the training data. The category and the corresponding classes with the number of images in each class is shown in Table 1. One sample image from each class is given in Fig. 14. E1, P1, DE1, L1, T1, C1, B1, H1 and D1 corresponds to first sample image of categories elephant, pig, deer, lion, tiger, bear, horse and dog respectively.

4.1 Software implementation results

The Matlab implementation results for image segmentation, feature extraction and object recognition are shown in this section. The entire algorithm is run on a CPU.

4.1.1 Segmentation results

The segmentation output for sample images from each class are shown in Fig. 14. The segmented outputs are based on the thresholding process.

4.1.2 Feature extraction

Table 2 presents eight different shape-based features i.e., major axis length, minor axis length, eccentricity, orientation, convex area, equiv diameter, solidity and extent that are extracted from the sample images under each category of animals, i.e., elephant, pig, deer, lion, tiger, cow, bear, horse and dog. The feature extraction values are extracted from the segmented object.

4.1.3 Recognition results

Table 3 shows the recognition status of each IR image. Recognition is done for the elephants. From the recognition results it is observed that when the IR database images are given as inputs, the images 1–7, 18 and 23 are detected as elephant hence the recognition status is "yes" and all other animals are not detected as elephants and hence the recognition status is "no". For verifying the results, the category

Table 11 Device utilization summary

Resources	Total	Segmentation	Feature extraction		
			Area	Centroid	Equiv diameter
No of slices	10,752	1	27	49	17
No of slice FFs	21,504	1	40	97	33
No of 4 i/p LUTs	21,504	1	11	96	32
No of IOs		18	1	122	130
No of bonded IOBs	448	13	1	121	129
No of GCLKs	32	1	3	1	1

Table 12 Computational time analysis between software simulation and hardware simulation for image processing

Techniques	Computational time in software simulation (s)	Computational time in hardware simulation (ns)
Segmentation	0.118	4.178
Feature extraction		
Area	2.257	4.661
Centroid		3.223
Equiv diameter		3.223
Total computational time	2.375	15.285

of the input IR images are also taken into consideration and the verification results are shown in Table 4. It is found that whenever the input IR image is an elephant the recognition status is "yes", hence elephant is recognised. The recognition rate has been evaluated and is shown in Table 5. Recognition rate is calculated from Table 4 as, when the category is elephant and if the recognition status is "yes", recognition rate is 100%.

If the category is not an elephant and if the recognition status is "no", recognition rate is 100%. A low performance shown for certain categories due to its smaller training set included in K-NN classifier. The classification and recognition results shown in Table 4 make use of K = 3. Observations are made by changing the value of K to 5 and 7 and the recognition results are calculated. These are then compared with the results obtained with K = 3.

Recognition rates with different K values are given in Table 6. On taking the average, and considering the entry of all other animal categories, K value of 7 is preferred since it has given an average recognition rate of 93.96%. But it is observed that for K = 3 and 5, elephants are perfectly detected. While for K = 7 the recognition rate for elephant is reduced to 85.71%. Hence K = 3 is chosen for recognition since the number of neighbours is less as compared to K = 5. This helps in the reduction of computation time and complexity of the algorithm.

4.2 Hardware implementation results

The hardware implementation results are obtained by hardware co-simulation using Xilinx System Generator (XSG). Results for image segmentation using thresholding technique, feature extraction and object recognition using K-NN classifier are shown in this section.

4.2.1 Segmentation results

The FPGA board used is Virtex-4 xc4vlx25-10ff668. The board is connected to the computer using JTAG for hardware co-simulation. Figure 15 shows the hardware co-simulation result for segmentation.

Table 13 Computational time analysis of object recognition in software simulation and software–hardware co-simulation

Techniques	Computational time in software simulation (s)	Computational time in software–hardware co-simulation
Segmentation	0.118	4.178 ns
Feature extraction		
Area	2.257	4.661 ns
Centroid		3.223 ns
Equiv diameter		3.223 ns
Object recognition using K-NN classifier	0.274	0.274 s
Total computational time	2.649	0.274 s

4.2.2 Feature extraction

From the segmented image, area, centroid and Equiv diameter features are extracted using XSG. Sample feature set extracted for each class is shown in Table 7.

4.2.3 Recognition results

The recognition results obtained from K-NN classifier are given in Table 8. Here the recognition status is set to 'yes' if an elephant is present in the image else 'no'. From the recognition results it is observed that 1–7, 12 and 14 are detected as elephant and all other animals are not detected as elephants. For verifying the results, the corresponding images are taken for verification and the results are shown in Table 8.

The verification results are shown in Table 9. The classification and recognition results shown in Table 8 make use of K = 3. The class wise recognition rates are evaluated and are shown in Table 10. For elephant the recognition rate is 100%. A very low performance shown for certain categories is due to its smaller training set included in the k-NN classifier. The recognition rate differs for lion, cow, horse and dog when recognized in software and hardware is due to the difference in the shape features extracted from the segmented image.

4.2.4 Device utilization summary and computation time analysis

The resource utilization of Virtex-4 FPGA when segmentation and feature extraction algorithm are computed is shown in Table 11.

Table 12 details about the computation time taken for processing segmentation and feature extraction (area, centroid, equivdiameter) by software and hardware. The results clearly indicate that hardware processing consumes time in nanoseconds while software processing consumes time in seconds. From Table 13, it is observed that the entire process of elephant recognition is done using software–hardware co-simulation. Thus the total computational time taken to process the object recognition algorithms in software is 2.649 s and software–hardware co-simulation is 0.274 s. The computation time delay is reduced by 89.65% in software–hardware co-simulation as compared to software simulation. Hardware processing means the entire algorithm run on a FPGA. Based on the FPGA architecture, the computation time differs. But the objective of this work is to prove that irrespective of the FPGA architecture, the computation time is less as compared to implementing on CPU.

5 Conclusions and future work

In this paper, thresholding based segmentation and k-NN classifier has been done for the elephant recognition of the segmented infrared image by prototyping in Virtex-4 xc4vlx25-10ff668 FPGA board. The effectiveness of our approach and implementation is evaluated by calculating the recognition rate and computational time. The results indicate that the approach is capable of recognizing the elephants from infrared database images and when implemented in hardware using software–hardware co-simulation method, the computation time reduces by 89.65% as compared to software simulation. In future, recognition rate of animals other than elephant may be improved and FPGA implementation of k-NN classifier can be done to further reduce the computational time. Analysis is done for 24 images and it can be extended for more number of images. FPGA implementation of K-NN classifier can also be done in future to achieve more decrease in the computational time.

References

Ardovini A, Cinque L, Sangineto E (2008) Identifying elephant photos by multi-curve matching. Pattern Recognit 41:1867–1877

Arivazhagan C, Ramakrishnan B (2010) Conservation perspective of Asian elephants (*Elephas maximus*) in Tamil Nadu, Southern India. Int J Biol Technol (IJBT) 1:15–22

Balch T, Dellaert F, Feldman A, Guillory A, Isbell C, Khan Z, Pratt S, Stein A, Wilde H (2006) How multi-robot systems research will accelerate our understanding of social animal behaviour. Proc IEEE 94:1445–1463

Blumstein DT, Mennill DJ, Clemins P, Girod L, Yao K et al (2011) Acoustic monitoring in terrestrial environments using microphone arrays: applications, technological considerations and prospectus. J Appl Ecol 48:758–767

Chikkali PS, Prabhushetty K (2011) FPGA based image edge detection and segmentation. Int J Adv Eng Sci Technol 9(2):187–192

Christe SA, Vignesh M, Kandaswamy A (2011) An efficient FPGA implementation of MRI image filtering and tumour characterization using Xilinx system generator. Int J VLSI Des Commun Syst 2(4):95–109

Dabarera R, Rodrigo R (2010) Vision based elephant recognition for management and conservation. In: Proceedings of the 5th international conference on information and automation for sustainability (ICIAfs'10), pp 163–166

Douglas-Hamilton I (2008) The current elephant poaching trend. Pachyderm 45:154–157

Douglas-Hamilton I, Krink T, Vollrath F (2005) Movements and corridors of African elephants in relation to protected areas. Naturwissenschaften 92:158–163

Fernando P, Leimgruber P, Prasad T, Pastorini J (2012) Problem-elephant translocation: Translocating the problem and the elephant? PLoS One 7:e5091

Goswami VR, Lauretta MV, Madhusudan MD, Karanth KU (2012) Optimizing individual identification and survey effort for photographic capture-recapture sampling of species with temporally variable morphological traits. Anim Conserv 15(2):174–183

Hannuna SL, Campbell NW, Gibson DP (2005) Identifying quadruped gait in wildlife video. In: IEEE international conference on image processing, pp I-713

Hoare RE, Du Toit JD (1999) Coexistence between people and elephants in African Savannas. Conserv Biol 13(3):633–639 **(International Journal of Reconfigurable Computing 13(3):1–7)**

Hussain HM, Seker H (2012) An adaptive implementation of a dynamically reconfigurable K-Nearest neighbour classifier on FPGA. In: Proceedings of NASA/ESA conference on adaptive hardware and systems.

Hussain HM, Benkrid K, Ebrahim A, Erdogan AT, Seker H (2012) Novel dynamic partial reconfiguration implementation of K-Means clustering on FPGAs: comparative results with GPPs and GPUs. Int J Reconfigurable Comput 2012:135926. https://doi.org/10.1155/2012/135926

Hwang W-J, Hsu C-C, Li H-Y, Weng S-K, Yu T-Y (2010) High speed c-means clustering in reconfigurable hardware. Elsevier, New York

Kamavisdar P, Saluja S, Agrawal S (2013) A survey on image classification approaches and techniques. Int J Adv Res Comput Commun Eng 2(1):1005–1009

Kumar KP, Parameswaran L (2013) A hybrid method for object identification and event detection in video. In: Computer vision, pattern recognition, image processing and graphics (NCVPRIPG), 2013 fourth national conference on IEEE, pp 1–4

Kumara HN, Rathnakumar S, Kumar MA, Singh M (2012) Estimating Asian elephant, *Elephas maximus*, density through distance sampling in the tropical forests of Biligiri Rangaswamy Temple Tiger Reserve, India. J Trop Conserv Sci 5(2):163–172

Ladgham A, Hamdaoui F, Sakly A, Mitbaa A (2012) Real time implementation of detection of bacteria in microscopic images using system generator. J Biosens Bioelectron 3(5):2155–6120

Lahiri M, Tantipathananandh C, Warungu R, Rubenstein DI, Berger-Wolf TY (2011) Biometric animal databases from field photographs: identification of individual zebra in the wild. In: Proceedings of the 1st ACM international conference on multimedia retrieval, pp 6

Lemieux AM, Clarke RV (2009) The international ban on ivory sales and its effects on elephant poaching in Africa. Br J Criminol 49:451–471

Mangai NS, Vinod ST, Chandy DA (2015) Recognition of elephants in infrared images using clustering- based image segmentation. Int J Electron Secur Digital Forensics 7(3):234–244

Muthukrishnan R, Radha M (2011) Edge detection techniques for image segmentation. Int J Comput Sci Inf Technol 3(6):259

Olivier PI, Ferreira SM, Van Aarde RJ (2009) Dung survey bias and elephant population estimates in southern Mozambique. Afr J Ecol 47:202–213

Pinter-Wolman N (2012) Human–Elephant conflict in Africa: the legal and political viability of translocations, wildlife corridors, and transfrontier parks for large mammal conservation. J Int Wildl Law Policy 15:152–166

Prabu M (2016) An efficient surveillance system to detect elephant intrusion into forest borders using seismic sensors. Int J Adv Eng Technol 7(1):166–171

Ramanan D, Forsyth DA, Barnard K (2006) Building models of animals from video. IEEE Trans Pattern Anal Mach Intell 28(8):1319–1334

Ramesh G, Mathi S, Pulari SR, Krishnamoorthy V (2017) An automated vision-based method to detect elephants for mitigation of human–elephant conflicts. In: 2017 International conference on advances in computing, communications and informatics (ICACCI), pp 2284–2288

Saegusa T, Maruyama T (2007) An FPGA implementation of real-time K-means clustering for color images. Proc J Real Time Proc 2:309–318

Santiapillai C, Wijeyamohan S, Bandara G, Athurupana R, Dissanayake N et al (2010) An assessment of the human–elephant conflict in Sri Lanka. Ceylon J Sci Biol Sci 39:21–33

Shaukin A (2015) Segmentation and identification of special object (elephant) in an image. Int J Sci Eng Technol Res 4(9):3180–3188

Suen CY, Cheriet M, Said N (1998) A recursive thresholding technique for image segmentation. IEEE Trans Image Process 7(6):918–921

Sugumar SJ, Jayaparvathy R (2013) An early warning system for elephant intrusion along the forest border areas. Curr Sci 104:1515–1526

Sugumar SJ, Jayaparvathy R (2014) An improved real time image detection system for elephant intrusion along the forest border areas. Sci World J 2014:393958. https://doi.org/10.1155/2014/393958

Van Aarde RJ, Ferreira S, Jackson T, Page B, De Beer Y et al (2008) Elephant population biology and ecology. In: Scholes RJ, Mennell KG (eds) Assessment of South African elephant management. Wits University Press, Johannesburg, pp 84–145

Wang XY, Yu YJ, Yang HY (2011) An effective image retrieval scheme using color, texture and shape features. Comput Stand Interfaces 33(1):59–68

Wrege PH, Rowland ED, Thompson BG, Batruch N (2010) Use of acoustic tools to reveal otherwise cryptic responses of forest elephants to oil exploration. Conserv Biol 24:1578–1585

Zeppelzauer M (2013) Automated detection of elephants in wildlife video. EURASIP J Image Video Process 2013:46. https://doi.org/10.1186/1687-5281-2013-46

Zeppelzauer M, Stoeger AS (2015) Establishing the fundamentals for an elephant early warning and monitoring system. BMC Res Notes 8(409):03–15