# Road Traffic Analysis on the CNN Universal Machine

Emel Arslan, Zeynep Orman

***Abstract*— CNN Universal Machines that contain two different processors working interactively with each other, have an important impact for image processing applications with their advanced computing features. These processors are named as the ACE16k processor which is the hardware implementation of cellular neural networks and the Digital Signal Processor (DSP). Bio-inspired (Bi-i) Cellular Vision System is also a CNN Universal Machine and its standalone architecture is built on CNN-type (ACE16k) and DSP-type microprocessors. In this study, certain objects in moving images are detected and their features are extracted. By using these features, an algorithm that finds out the path of moving vehicles and determines the traffic density is implemented on the Bi-i Cellular Vision System. Finally, the output images and the data related with traffic density which are obtained as results of this implementation are evaluated.**

***Index Terms*— Bi-i Cellular Vision System, ACE16k, Digital Signal Processor, Target Tracking, Traffic Analysis**

## I. Introduction

IMAGE processing is one of the most important research topics in recent years. It is widely used in areas such as military, security, health, biology, astronomy, archeology and industry [1, 2]. For an image to be processed, it should first be presented in a format that a computer can understand, this means it should be converted into its related digital form. In the digital form, each of its pixel is expressed by means of the corresponding element of a matrix.

Algorithms that are developed for digital image processing require fast systems due to their processing load. Although conventional computer systems are in an increasing trend in terms of their speed, they are insufficient when image dimensions are getting larger, because they can process transactions in a serial manner. These computers cannot satisfy the need for speed, when we especially consider the implementation of real-time moving image processing algorithms that require at least 15-25 frames to be processed in a second.

Cellular Neural Network (CNN) theory that was proposed by Chua and Yang in 1988, is an analog, nonlinear and a real-time processing neural network model [3]. CNNs also have advanced features for image processing applications. In 1993, Roska and Chua have presented the CNN Universal Machine [4]. This analogical array computer has cellular processors (ACE4k, ACE16k, etc. [5, 6]) which are the hardware implementation of CNNs and it is very suitable for image processing applications with its advanced computing capabilities. Bi-i Cellular Vision System is a CNN Universal Machine that can process high-speed and real time transactions and can be defined as a compact, independent and intelligent camera. This system has high-resolution sensors and two different processors named as CNN (ACE16k) and Digital Signal Processor (DSP) that can communicate with each other [7, 8].

In this study, firstly, an algorithm that finds out the path of vehicles and determines the density on the road traffic is proposed. This algorithm initially recognizes vehicles in moving images and extracts their features. Then, it detects the path of these vehicles which they are tracking all along the road and calculates the traffic density by using these extracted features. Secondly, an implementation of this algorithm is presented on the Bi-i Cellular Vision System and finally, the output images and the traffic density data which are obtained as the result of this study are evaluated.

Moving target tracking is an important research field that is frequently used in robotics, video surveillance, traffic control, etc. Generally, there are two kinds of approaches adopted for target tracking [2].

- Recognition-based target tracking
- Motion-based target tracking

Recognition-based target tracking methods extract features and match them between image frames to fit the global motion model of moving image sequence. On the other hand, motion-based methods track image features over a sequence of images by using the difference between each frame. Although the latter one has an approved theory foundation, the former one is more popular.

In recent years, there are numerous applications in the literature that use different methods for tracking moving objects. Kalman filtering is a widely used method for target tracking systems [2]. Some different approaches were also presented in [9] and [10] for traffic analysis and target

This work was supported by Scientific Research Projects Coordination Unit of Istanbul University. Project number: 16595.

Emel Arslan is with the Research and Application Center for Computer Sciences, Istanbul University, Istanbul, Beyazıt, 34452, Turkey ( phone: +90-212-4400093; fax: +90-212-4400094; e-mail: earslan@istanbul.edu.tr).

Zeynep Orman is with the Computer Engineering Department, Istanbul University, Istanbul, Avcilar, 34320, (e-mail: ormanz@istanbul.edu.tr).

ISBN: 978-988-18210-9-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

tracking. [11] and [12] proposed different systems that could detect moving objects on a mobile robot. In [13], Yilmaz et al. presented a detailed survey on object tracking and discussed and classified various related algorithms.

The remainder of this letter is organized as follows. Section 2 introduces fundamental concepts about CNN architecture, CNN Universal Machine, ACE16k processor, Bi-i Vision System and Bi-i programming, respectively. In Section 3, an algorithm that detects the path of vehicles and determines the traffic density is proposed and in Section 4, this algorithm is implemented on the Bi-i Vision System. Section 5 evaluates the obtained results and finally the paper is summarized in Section 6.

## II. CNN Architecture and Bi-i Cellular Vision System

This section provides fundamental concepts about CNN architecture and Bi-i Cellular Vision System.

### A. *Architecture of the Cellular Neural Networks*

Cellular Neural Networks (CNNs) derived from Hopfield Neural network is introduced in [3]. The two most fundamental components of the CNN paradigm are the use of analog processing cells with continuous signal values, and local interaction within a finite radius.

The structural design of CNN is formed with basic circuits called cells. Each cell containing a linear capacitor, a non-linear voltage controlled current source, and a few resistive linear circuit elements is connected to its neighboring cells; therefore direct interactions take place only among adjacent cells. Mathematical expression of the standard CNN model is described by the set of linear differential equations given by (1) which are associated with the cells in the circuit. The activation function of the CNN cell can be expressed by the set of nonlinear equation (2).

$$C\frac{d}{dt}x_{ij}(t) = -\frac{1}{R}x_{ij}(t) + \sum_{C(k,l)\in S_r(i,j)} A(i,j;k,l)y_{kl}(t) + \sum_{C(k,l)\in S_r(i,j)} B(i,j;k,l)u_{kl}(t) + z_{ij} \qquad (1)$$

$$y_{ij} = f(x_{ij}) = \frac{1}{2}\left(\left|x_{ij}+1\right| - \left|x_{ij}-1\right|\right) \qquad (2)$$

where,

$x_{ij} \in R$ ; State variable of cell *C(i,j)*,

$y_{kl} \in R$ ; Outputs of cells,

$u_{kl} \in R$ ; Inputs of cells,

$z_{ij} \in R$ ; Threshold,

*A(i, j ; k, l)* ; Feedback operator,

*B(i, j ; k, l)* ; Control operator.

$y_{ij}$ ; Output equation.

Without loss of generality, linear resistor (R) and linear capacitor (C) values can be set to 1. The block diagram of a cell C(i, j ) is shown in the Figure 1.

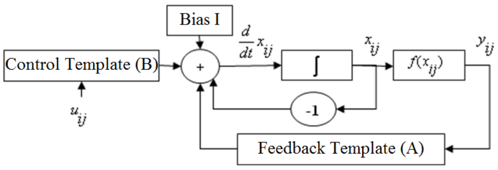


Fig. 1 The block diagram of a cell C(i, j)

In this way CNNs have provided an ideal framework for programmable analog array computing. It means that the CNN can be used as a programmable device where the instructions are represented by the templates which define the connections between a cell and its neighboring cells. In general, the CNN templates consists of the feedback template (B), control template (A) and bias value. Basically, three different images can describe a CNN layer, that is, the input U, the state X and the output Y.

Let us assume a CNN with MxN cells are arranged in M rows and N columns and the cell in row i and column j is denoted as C(i,j)[3]. r-neighborhood of a C(i,j) cell is defined with the following definition (3) provided that r is a positive value.

$$N_r(i,j) = \left\{ C(k,l) \;\middle|\; \max_{1\le k\le M, 1\le l\le N} \{|k-i|,|l-j|\} \le r \right\} \qquad (3)$$

### B. *CNN Universal Machine*

The hardware implementation of CNN is easier compared to the Artificial Neural Networks as there is only connection between the neighbor cells and the cell structure. Analogical Cellular Engines (ACE4k, ACE16k etc. [6]) are based on CNN Universal Machine architecture. CNN Universal Machine(CNN-UM) architecture has been called by Roska and Chua as analogical computation since it can both perform the analog array operations and the logical operations together [4].

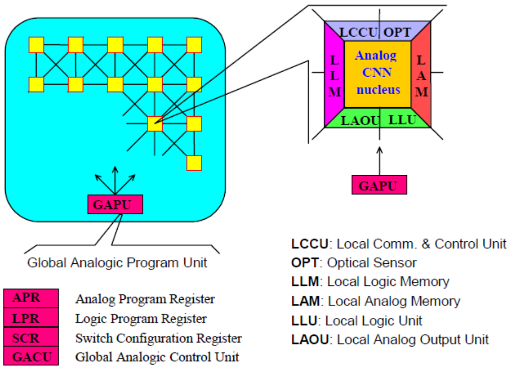


Fig. 2 The architecture of the CNN Universal Machine

Fig. 2 denotes the CNN-UM architecture which is based on the dynamic computing of a simple CNN. This figure shows the elements in the complex CNN Nucleus and the functional blocks of the Global Analogic Programming Unit[8].

### C. *ACE16k Processor*

ACE16k, is a CNN based processor of CNN Universal Machine which can perform analog operations. ACE16k which is used to perform various image processing operations contain low resolution (128 x 128) CMOS gray level image sensor and analog processor arrays. This processor array is much faster (30000 frames per second) than the conventional processors in image processing applications since it can processes the whole image in parallel.

### D. *Bi-i Cellular Vision System*

The Bi-i Cellular Vision System which contains two different processors, a CNN based ACE16k and a DSP that can be defined as a compact, standalone and intelligent camera capable of real time operations at very high speed [7, 8]. The images are stored in local memories with the help of two different sensors as a (1280x1024) color CMOS sensor, and a (128x128) ACE16K sensor.

The block diagram of the Bi-i V2 Cellular Vision System is given in Figure 3. As seen from the figure, this system has a color CMOS sensor array (IBIS 5-C) and two high-end digital signal processors (TX C6415 and TX C6701). This system runs an embedded Linux on the communication processor and has complex external interfaces like USB, FireWire and a general digital I/O in addition to the Ethernet and RS232[8].

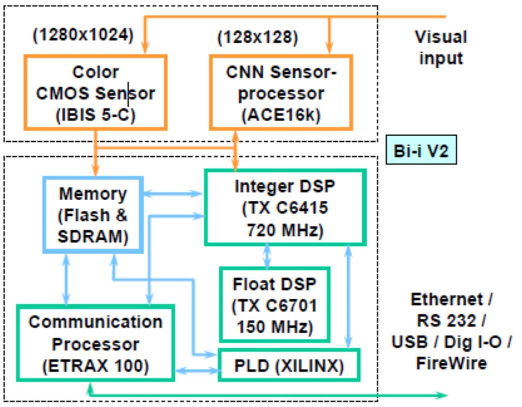

(1280x1024)
(128x128)
Visual input
Color CMOS Sensor (IBIS 5-C)
CNN Sensor-processor (ACE16k)
Bi-i V2
Memory (Flash & SDRAM)
Integer DSP (TX C6415 720 MHz)
Float DSP (TX C6701 150 MHz)
Communication Processor (ETRAX 100)
PLD (XILINX)
Ethernet / RS 232 / USB / Dig I-O / FireWire

Fig. 3 The block diagram of the Bi-i V2 Cellular Vision System

### E. *Bi-i Programming*

CNN Universal Machine has two different programming methods. One of them is AMC (Analogical Macro Code) language which is a conventional Bi-i programming method. The codes written in AMC language are converted to binary basis and run on Bi-i. Another method is the Bi-i (Software Development Kit - SDK) which is used to develop more complex applications. Bi-i SDK, consists of the C++ programming library which is a group used to develop applications. These libraries can also used for the Digital Signal Processor (DSP) with the development unit Code Composer Studio and they contain many functions to control the whole ACE16k circuit [7].

## III. THE PROPOSED ALGORITHM

In this section, we will present an algorithm that detects the path of vehicles and determines the density of a traffic flow on a highway. This algorithm can be successfully used for traffic analysis. Traffic analysis that we consider in this study includes detection of the path of vehicles and calculation of the traffic density, respectively. The colored moving image we consider, is first preprocessed so that it is transformed in a format that the functions can process to detect moving objects and the path of each object in the image. We will discuss these preprocessing operations in the Preprocessing Section in detail.

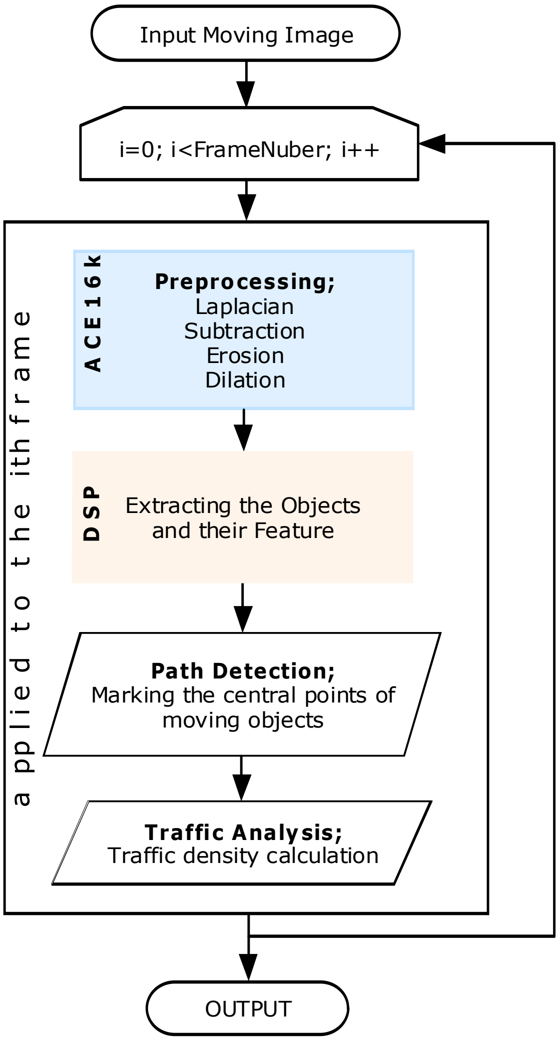


Fig. 4 Flow diagram of the proposed algorithm

After preprocessing phase, the moving image can be an input to a function that finds out the objects and extracts their features from the image. This function produces an output of extracted features of the detected objects. These features will then be an input to another function that detects moving objects in the image.

Then, using these extracted features, locations of each moving object are marked in each frame. When this process is complete for all frames, the path that all vehicles are tracking all along the image will be plotted. These extracted features are also used for determining the traffic density. In this application, the camera's field of view is considered to be a 15 meters distance. The sample images belong to a 3 lanes road and depending on the average vehicle length, each lane is assumed to have a maximum of 3 vehicles at a time. According to this data, the traffic density based on the number of vehicles in each frame is evaluated as gridlock traffic, heavy traffic and moving well traffic.

Figure 4 denotes the steps of the algorithm that is used for this implementation. As clearly seen from the algorithm, preprocessing phase is run on the ACE16k processor and other operations run on the DSP processor. Therefore, the algorithm is implemented on both processors interactively for different phases which make the processing of images faster.

## IV. IMPLEMENTATION OF THE ALGORITHM ON THE BI-I VISION SYSTEM

In this section, the algorithm that detects the path of vehicles in moving images and determines the traffic density will be implemented on the Bi-i Cellular Vision System. The phases of this implementation are given in detail as follows:

### *A. Preprocessing Phase*

The moving image we consider for our implementation, first undergoes a series of enhancement operations to get better results in detection of objects.

Because the moving image is colored, we first transform it into a gray level image by using RGB2ByteMatrix function and then apply the preprocessing operations. To increase the success of detecting objects in a moving image, we use a Laplacian based image enhancement technique so that the edges in the image have been made more apparent [2].

Image enhancement technique based on Laplacian can be implemented by using a Laplace template or a Laplace function that is defined in Bi-i SDK [7]. Laplacian is a second order differential operator and it enhances gray level discontinuous points of an image, this means the edges of the image become much clearer by using this technique. It also reduces the importance of slowly changing gray level areas. The resulting image after we obtain by applying the Laplace function, is subtracted from the base image pixel by pixel. In this manner, the image is enhanced to get better edges in the next step.

The Laplace function and the Subtract function that is used to subtract the resulting image from the original input are both ran on the ACE16k processor.

In Figure 5, original images and gray level images that are obtained during the preprocessing phase of the implementation for different frames are given. It also shows the Laplaced and the substracted images of the original images which are obtained during the implementation of the Laplacian-based image enhancement procedures.

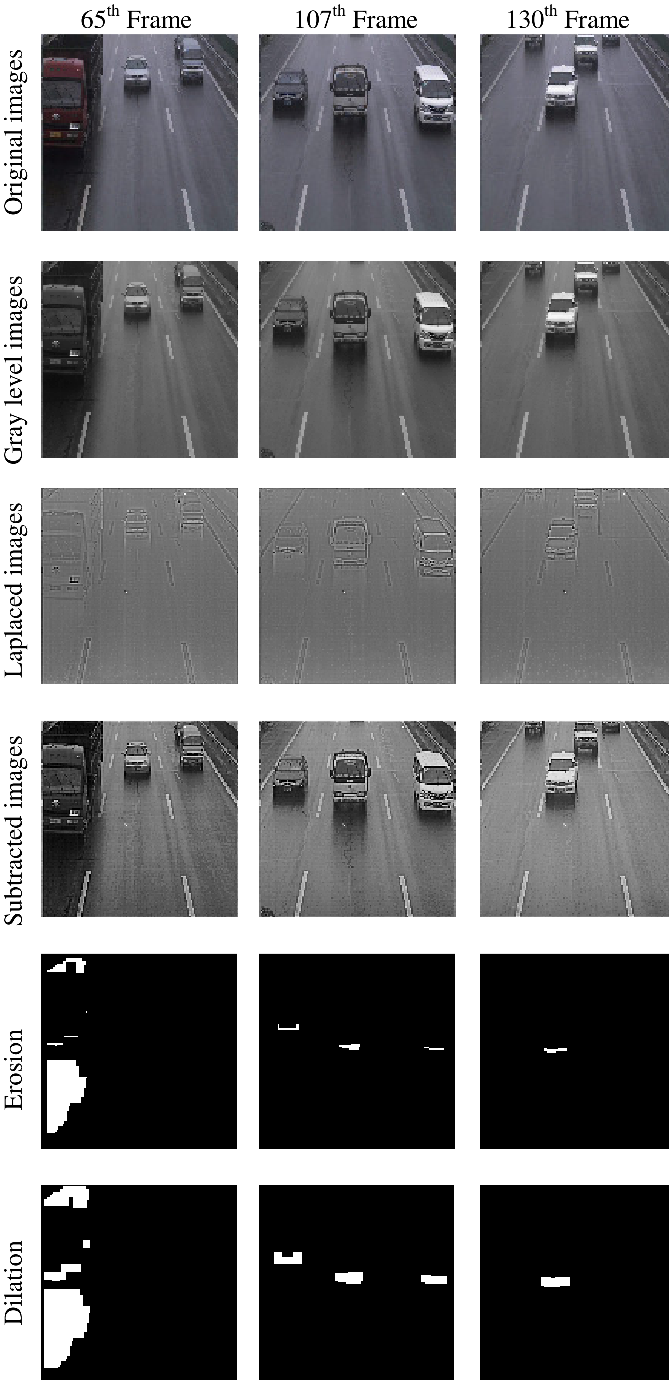


Fig. 5 Resulting images that are obtained after the preprocessing phase for different frames

After the enhanced moving image is converted into a binary image with the ByteMatrix2BitMatrix, it undergoes a 3 iteration erosion and a 2 iteration dilation, respectively. With these operations, the objects become more apparent. Figure 5 also shows the resulting images that are obtained after the erosion and dilation phases.

After the preprocessing phase is completed, the moving image is now ready for the CalcFeatures function to be an input image which will detect the objects and extract the features of these objects.

### *B. Extracting the Objects and their Features*

The CalcFeatures function in the Instant Vision library is used to detect the objects in each frame and extract the features of these objects from the preprocessed binary image. This function can output the features like area, bounding box, extremes, orientation, centeral point, eccentricity, diameter, extent etc. that belong to each object in the image.

In our implementation, the CalcFeatures function is used with the following command line:

```
CalcFeatures (ObjNum, Features, PicBin, FEAT_CENTROID, InstantVision::CONN4, 256, CALCACC_FLOAT, CALCMODE_SQUARED);
```

FEAT_CENTROID parameter of the CalcFeatures function indicates that this function should only calculate the central point of the objects. It is very important in our implementation to know the central point of the objects in each frame to find out the path that these objects are tracking. We've only used FEAT_CENTROID Feature Selection parameter because there is no need to calculate the other features.

The objects which their central points are calculated are now ready for the operations that will detect the path of these objects. The functions that can perform these operations are defined in MTT (Multi Target Tracking) which is located under the Instant Vision library.

Meas vector variable which contains the central point information of an object is given as an input parameter of a function that will track the moving objects in the image. By using this vector, this function also checks all the objects that are in the path tracer of the Tracker function. If a specific object is detected as a moving object, the function that is written in C++ programming language, will mark the central point of this object. After this operation is performed for every object in each frame, the path of each vehicle will be plotted in the moving image. Another function that is also written in C++ is used for obtaining data about traffic density based on number of vehicles in each frame.

## V. Experimental Results

For this implementation, the moving image we consider is a traffic flow taken on a highway. Therefore, each moving object in the image corresponds to a vehicle such as cars, trucks, etc. When the central point of the moving objects in each frame is marked, it will give us the path that they are tracking all along the highway.

When the algorithm which is detailed in Section 3, is implemented on the Bi-i Vision System, some sample output images can be obtained as given in Figure 6 for moving object one and object two, respectively. Figure 7 shows the gathered data for traffic density in different frames.

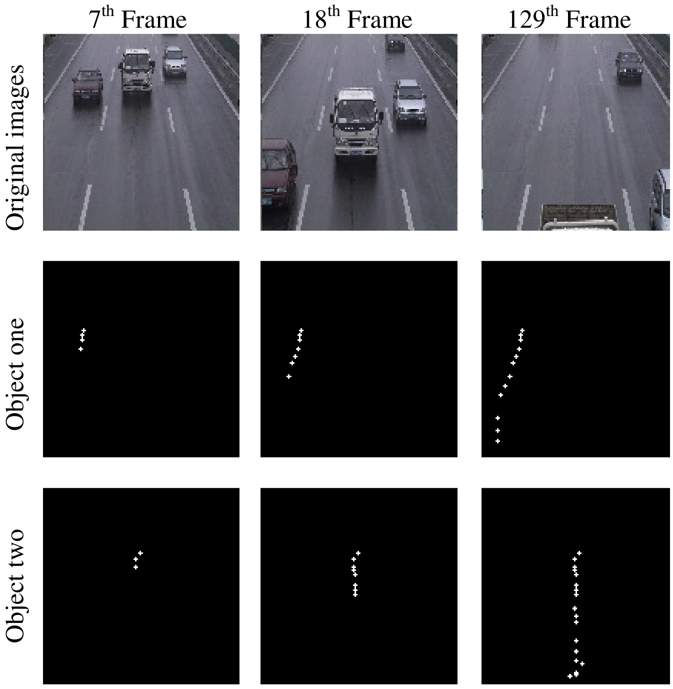
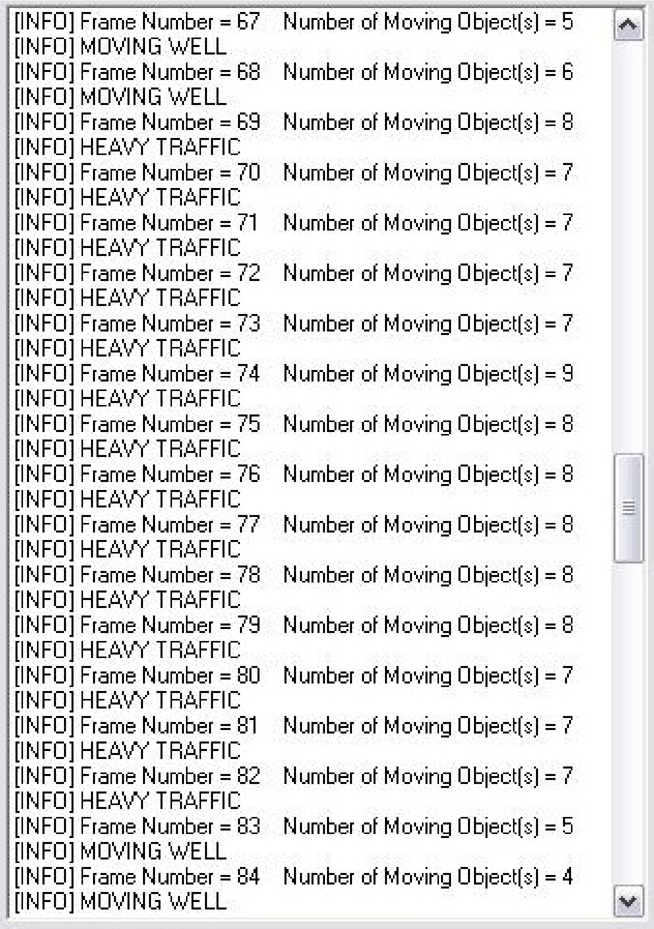


Fig. 6 Sample output images showing the path of moving objects



```
[INFO] Frame Number = 67    Number of Moving Object(s) = 5
[INFO] MOVING WELL
[INFO] Frame Number = 68    Number of Moving Object(s) = 6
[INFO] MOVING WELL
[INFO] Frame Number = 69    Number of Moving Object(s) = 8
[INFO] HEAVY TRAFFIC
[INFO] Frame Number = 70    Number of Moving Object(s) = 7
[INFO] HEAVY TRAFFIC
[INFO] Frame Number = 71    Number of Moving Object(s) = 7
[INFO] HEAVY TRAFFIC
[INFO] Frame Number = 72    Number of Moving Object(s) = 7
[INFO] HEAVY TRAFFIC
[INFO] Frame Number = 73    Number of Moving Object(s) = 7
[INFO] HEAVY TRAFFIC
[INFO] Frame Number = 74    Number of Moving Object(s) = 9
[INFO] HEAVY TRAFFIC
[INFO] Frame Number = 75    Number of Moving Object(s) = 8
[INFO] HEAVY TRAFFIC
[INFO] Frame Number = 76    Number of Moving Object(s) = 8
[INFO] HEAVY TRAFFIC
[INFO] Frame Number = 77    Number of Moving Object(s) = 8
[INFO] HEAVY TRAFFIC
[INFO] Frame Number = 78    Number of Moving Object(s) = 8
[INFO] HEAVY TRAFFIC
[INFO] Frame Number = 79    Number of Moving Object(s) = 8
[INFO] HEAVY TRAFFIC
[INFO] Frame Number = 80    Number of Moving Object(s) = 7
[INFO] HEAVY TRAFFIC
[INFO] Frame Number = 81    Number of Moving Object(s) = 7
[INFO] HEAVY TRAFFIC
[INFO] Frame Number = 82    Number of Moving Object(s) = 7
[INFO] HEAVY TRAFFIC
[INFO] Frame Number = 83    Number of Moving Object(s) = 5
[INFO] MOVING WELL
[INFO] Frame Number = 84    Number of Moving Object(s) = 4
[INFO] MOVING WELL
```

Fig.7 Traffic density data

## VI. Conclusion

In this paper, we have presented an algorithm that detects objects and extracts their features in moving images and by using these features, this algorithm finds out the path of the vehicles on a highway and determines the traffic density. We have also implemented this algorithm by using the Bi-i Cellular Vision System which offers an ultrahigh speed processing in image processing applications. The preprocessing operations of the algorithm which are Laplacian image enhancement, subtraction, dilation and erosion are implemented on the ACE16k processor, respectively. The other operations that are converting the image into a gray level image, extracting features of the objects, detecting moving objects by using the extracted features and detecting the path of these objects in the image are all implemented on the Digital Signal Processor (DSP). When tested in traffic scenes, the algorithm successfully detects the path of each vehicle in the moving image by generating different image files and obtains the traffic density information.

This study shows that the Bi-i Vision System which is successfully used in image processing applications, can also give very fruitful results for moving and multi-target tracking applications.

For a future work, the algorithm that we proposed for path detection and traffic density can be improved to determine the traffic faults like lane violation of the vehicles in a traffic flow or to analyze traffic on a highway for different applications like determining the speed of vehicles and/or the distance between two vehicles.

## References

[1] C. Gonzales and R.E. Woods, "Digital Image Processing", Prentice Hall, New Jersey, 2002

[2] T. Acharya and A.K. Ray, "Image Processing: Principles and Applications", Wiley and Sons, 2005

[3] L. O. Chua and L. Yang, " Cellular neural networks: Theory and applications", IEEE Trans. on CAS, vol. 35 no. 10, pp.1257–1290, 1988

[4] T. Roska and L. O. Chua, "The CNN universal machine: an analogic array computer", IEEE Trans. on CAS-I, vol.. 40 no.3, pp. 163–173, 1993

[5] T. Roska and A. Rodriguez-Vazquez, "Towards visual microprocessors", Proceedings of the IEEE, vol. 90 no.7, pp. 1244–1257, 2002

[6] A.R. Vazquez, G. L. Cembrano, L. Carranza, E.R.Moreno, R.C. Galan, F.J. Garrido , R.D. Castro and S. E. Meana, "ACE16k: the third generation of mixed-signal SIMDCNN ACE chips toward VSoCs", IEEE Trans. CAS-I, vol. 51, no.5, pp. 851– 863, 2004

[7] AnaLogic Computers Ltd http://www.analogic-computers.com /Support

[8] A. Zarandy and C. Rekeczky, "Bi-i: a standalone ultra high speed cellular vision system.", IEEE Circuit and Systems Magazine, vol. 5, no.2, pp. 36–45, 2005

[9] CUCCHIARA, R., GRANA, C., NERI, G., PICCARDI, M. , and PRATI, A., "The Sakbot System for Moving Object Detection and Tracking", Video-Based Surveillance Systems- Computer Vision and Distributed Processing, 145-157, 2001

[10] E. Atkociunas, R. Blake, A. Juozapavicius, M. Kazimianec, "Image Processing in Road Traffic Analysis", Nonlinear Analysis: Modelling and Control, Vol. 10, No. 4, pp.315–332, 2005.

[11] JUNG, B. and SUKHATME, G.S., "Detecting Moving Objects using a Single Camera on a Mobile Robot in an Outdoor Environment", In the 8th Conference on Intelligent Autonomous Systems, 980-987, Amsterdam, The Netherlands, March 10-13, 2004.

[12] BEHRAD, A., SHAHROKNI, A. and MOTAMEDI, S.A., "A robust vision-based moving target detection and tracking system", In the Proceeding of Image and Vision Computing Conference, University of Otago, Dunedin, New Zealand, November, 2001.

[13] YILMAZ, A., JAVED, O., and SHAH, M., "Object tracking: A survey", ACM Comput. Surv. 38(4), Article 13, 45 pages, 2006