RESEARCH ARTICLE OPEN ACCESS

Developing Video Surveillance System with Processing-Aware Network-On-Chip Using Parallel Coordinate Descent PCDN

Eng. Rony Kassam [1], Talal Alaalateky [2], Radwan Dandah [3]
Departmetn of Computer Systems and Networks
Tishreen University/Information Engineering Faculty
Syria

ABSTRACT

The Traditional video surveillance takes a huge amount of space storage. Recording everything captured by a surveillance camera consumes the storage devices used by the system. Extracting useful and meaningful information from surveillance videos is a time consuming process due to the long time of the recorded videos. These drawbacks limit the effectiveness of traditional video surveillance systems. In this paper, we propose a design of a Network on chip NoC that can handle the process of data stream analysis according to important features of the monitoring application, and can learn to parallel execute this process using processing elements within the NoC. The research results in linking the PCDN with the NoC and in training the classifier on a specialized dataset of traffic accidents to achieve effective parallelism. The speedup value reached 1.3, which is very important in the distribution of work on several processing elements that analyze and handle data at the same time.

Keywords :— Network-on-Chip, Coordinate Descent Algorithm, CarCrash Dataset, Parallel Training and Classifying.

I. INTRODUCTION

Visual surveillance is an important application because it deals with real-time videos. Where the amount of monitoring data collected may increase significantly over time. According to the International Data Corporation (IDC) report [1] about half of the large data is obtained from visual surveillance and it is estimated that this trend will increase with the following decades. [2] shows that the frequency in the large size of the surveillance video data comes from the similarity of the images to the same model as well as the relationship between the appearance of the component in different camera recordings and its stability for a period of time within the same recording. By encoding the form only the general frequency can be deleted in the scope of the monitored component and thus able to reduce the data size. This size can also be mitigated by encoding the moving components, where the component component's fragmentation algorithm is based on edge information alone, and the motion compensation is only for the component range, so it is suitable for real-time video encoding. The video surveillance system detects and tracks several components and monitors their movement in a specific environment (internal, external) by relying on grayscale video images or on infrared video images from surveillance cameras. For example, when people are tracked and analysed, their shape and parts (head, hands, feet, and trunk) are determined, and then patterns of appearance are created so that they can be traced through the activities they perform. That is, it is possible to determine whether the control area contains people and that the area can be divided into several sub-regions and that tracking will be carried out.

For efficient partitioning, optimization algorithms are used to determine the most suitable areas. One of these algorithms is Coordinate Descent [3] which is based on the idea that the minimum multivariate F(x) can be achieved by reducing the coordinate value along one direction At one time, any single-variable optimization problems can be simplified into a single iterative loop that is applied along this trend. This gradient is one of the optimization algorithms that deal with a single procedure for a structured linear classification. Because of its sequential nature, it is not possible to apply a multi-core effective subsystem easily.

In order to achieve an effective gradient, there is a need for data formations that reflect the videos taken from surveillance cameras. A variety of data has been created by the research [2], thus general data that cannot help the works to reach the effective division of control areas. For the architecture of this system, methods were introduced to exploit the embedded network routers on the chip to divide the work through logical components that link the source and the destination more quickly [4], but the decision-making remains limited to the NoC system and without any coordination with the application and its purpose.

The importance of the research is to activate the embedded networks on the chip within the visual surveillance systems and to extract important information from the videos quickly and smartly and the spread of this work on the appropriate number of contract NoC to increase the feasibility of treatment and at the same time reduce energy consumption.

A built-in network on the NoC chip was developed to deal with the visual surveillance system, which varies according to the objects studied, the dimensions of the area in which they are moved, the values of parameters that assist in object identification, and the PCDN advanced learning algorithm, which can be subdivided with CarCrash. So that the NoC network can modify and distribute its processing capacity (64 elements of Altera FPGA) according to the system generated from the machine learning.

II. PROPOSED VISUAL ANALYTICAL SURVEILLANCE MODEL

A. Hypotheses

In order to solve the problems mentioned in the consumption of processing and storage during the operation of visual surveillance systems, a method was adopted to deal only with frames that contain important information, for example, frames that contain changes in the scene. This can be achieved with a low cost web camera and algorithms to detect change and derive features. Thus there is a need for a system to check the similarities and differences of frames by considering that any difference is an adjustment of one or several parts on the frame. Therefore, the first frame is the original frame and its reference components should be determined. The second frame is the rate also need to study the parameter PSNR, which can be defined by the Mean Square Error MSE caused by the difference between the two images [2]. Where we have two images: I1 and I2 in dimensions m and n, consisting of the number of c channels, given the MSE in relation:

$$MSE = \frac{1}{c\times m\times n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(I_1(i,j) - I_2(i,j))^2 \quad (1)$$

PSNR is then expressed as follows:

$$PSNR = 10\log_{10}\frac{Max^2}{MSE} \quad (2)$$

Where Max is the maximum valid value for pixels. If a simple image contains one byte, each pixel has a value of 255. When taking two series and similar images, MSE will give zero, resulting in a zero division in the PSNR relationship. In this case, PSNR is not defined and because of the need to handle this situation separately, then you can go to a logarithmic scale because pixel values have a wide range.

Also, in order to predict the quality of digital images and video clips, structural similarity (SSIM) can be used in this research to rely on the SSIM indicator applied to the video clips of the data set and different locations within the snapshot. This indicator is calculated when measuring between the xx and y size of the NxN by relationship [2] [5]:

$$SSIM_{(x,y)} = \frac{(2\mu_x\mu_y+C_1)\times(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)\times(\sigma_x^2+\sigma_y^2+C_2)} \quad (3)$$

Where $\mu_x$mean x, μ_y mean y, σ_x^2 dispersion x, σ_y^2 dispersion y, σ_{xy} standard deviation of x and y.

$$C1 = (K_1 \times L), \qquad C2 = (K_2 \times L)$$

Both C1 and C2 are stability variables. Where L is the dynamic range of pixel values (usually the number of bits per 1 pixel), and the following values are constants used in the search Kl = 0.01 and K2 = 0.03. The algorithm scans the connecting frames to make sure they are similar using the PSNR relationship. If the inbound frame does not retain any changes from the last recorded frame, then the frame is ignored and the recorded frame meter increases. Only frames that hold changes are retained, creating a very small video for this step compared to the original video. Each of these summarized videos is divided into sub-frames N x M. Each sub frame is compared to its transmitter in the last recorded frame. If the sub-window does not contain any changes, this sub frame will be deleted and the counter incremented to indicate that the sender window will repeat in the encoding process.

B. Proposed Dataset

To evaluate the proposed technology, a data set derived from the CCTV system was used and adjusted to show the greatest possible number of incidents [6]. The goal of this combination is to predict accidents through advanced learning algorithms, thus increasing the time required for training and testing. This helps to give importance to speeding up this process through the NoC embedded in the FPGA and supported by learning algorithms.

The variety of data is characterized by the large number of small-sized control objects, which is a challenge to the ability of the workbook to deal with these objects. The average length of videos is 366 frames. In order to increase the ability to train the works, this collection provides a number of positive videos of 1416 including only traffic accidents, and there are also clips containing more than one accident. For negative sections (which do not involve accidents) help train the workbook also to deal with the changes. This combination marks the time period of the first incident within each video, with a median duration of 3.69 seconds. This time significantly affects the speed of incident prediction. The videos in this collection are real and were taken in different conditions throughout the year in terms of the number of cameras, variety weather conditions and others. Fig. 1 shows a sample of these data. In the third sample, the size of the studied body (the person crossing the street) is little comparing to bus traffic.

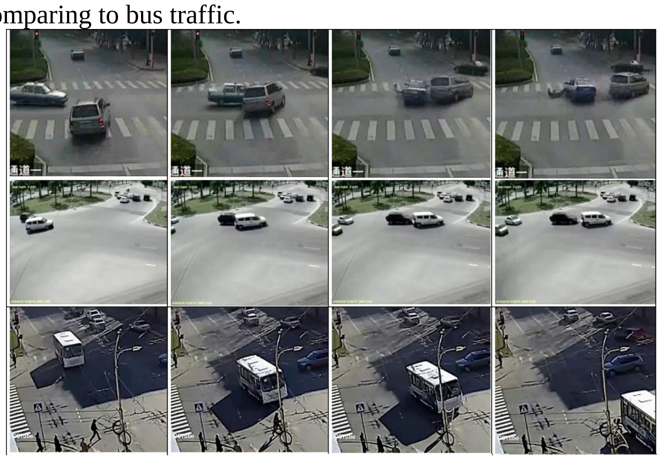


Fig. 1 Sample of proposed dataset.

C. Parallelization of NoC in Coordinate Descent

The parallel work of coordinate descent can be routed to include coordinates on gradients, this strategy is effective because many datasets have been distributed based on their features to non-zero distributions. In order to develop this work, the dataset can be divided into a few other sets that contain less dispersed distributions. When applying any multi-tasking task on a distributed dataset, the time required to execute the task will be affected by the number of operations to be performed, the time of each operation and the number of applications used [7].

$$Running\ Time = \frac{\#operations \times time\ per\ operation}{\#threads} + overhead \quad (4)$$

Thus, when the number of operations is small, the load will lead to an increase in implementation time compared to the case of one thread implementation. On the other hand, when performing these operations on data from this sets contains little information then there will be no possibility of division of work since the information and processes are few, especially in the early stages, and this is evident in the hierarchy of coordinates. Therefore, the trend begins with the formation of a number of instances and a number of features, which can be divided the proposed dataset to ease the distribution of data within each section. According to [7] we find that when focusing on the instances there will be a weakness in the achievement of the subsystem where there will be a need for the presence of thread or sub-treatment of each instance, which is considered as the lines within the database, so many lines will not be processed only in the same way and thus treatment of one thread is sufficient for each instance Therefore, parallel is weak here. Therefore, this research focuses on features.

Let “f” be a feature, then the data generated from the configurations of these features will have the form x1, .., xf arranged descending by the number of positive states (that is, the specified entry can be classified by the specified feature). Let “a” be so that the partial group {x1, .., xa} contains 50% of the positive cases. Let “b” be so that the partial group {x1, .., xb} contains 80% of the positive cases. “Na” is the number of entries that can be categorized by the feature of order “a”, and “Nb” is the number of entries that can be classified by the feature of order “b”, and “N1a” is the number of entries that can be classified by the features of order 1 to a, and in the same way “N1b” is the number of entries that can be classified by the features of order 1 to b.

The research relied on a set of features PSNR, SSIM, 4x4_Frame, 3x3_Frame, 5x5_Frame, and 10x10_Frame. It is useful to know where the processing pressure is in some controlled parts within the studied area and thus the acceleration that can be achieved in the subsystem is effective through the most possible identification of parts and features In which the calculation of each part and each property needs to be dealt with separately through thread or an independent parallel process.

At each step of the coordinate gradation, for a feature j, the goal is to find the smallest solution for the following single variable sub-issue:

$$MIN_z\ f(w + ze_j) - f(w) \quad (5)$$

Where w represents the current solution, and e_j is a vector expressing the studied feature. This problem is solved by using a squared hinge loss to teach the classifier, and proceed the linear search for solutions of the equations. When solutions stop, the value of w is updated with new ones. Thus, this method is known as CD Newton (CDN) [3]. Many improvements have been made to the CDN, especially with the increase in the capacity of the computational computing, where it is possible to do the interpolation of features and thus to deal with their data independently and without the need to pre-processing these data, this algorithm is known as Parallel CDN (PCDN) [8]. This algorithm uses multidimensional approach steps, where multidimensional space is divided by the extrusion of elements outside the diameter determined by the Hessian matrix [9]. This algorithm can therefore process Newton gradients in a subset. The steps of the algorithm can be summarized as follows:

1. In frequency k, the set of properties is randomly divided into partial groups.
2. Each subset is processed by a Newton gradient approach.
3. In implicit repetition t, the P-dimension approach is processed (where P is the size of the studied subgroup). The selected items are used with Hessian diameter. This is useful in making each approach independent of others in order to be able to implement the parallelism.
4. In the same frequency t, the step size is examined in the linear search to update the value of the studied function.

As described above, the parallelism can be applied to implicit repetition (loop t). In this research, this replication was designed to process the operation through the NoC. Also, as shown, PCDN randomly disaggregates the set of features, while in the search, the studied features are set into a different sections that express the more studied application in order to have a quick response. Where there is a possibility to pre-examine and fill up the values of these features in an initial pilot phase and then work.

To simplify the idea of linking this algorithm with the NoC, Fig. 2 shows a model to make an embedded grid on a chip deals with application states that reflect the diversity of data exchange, data format and resource need [10], so that whenever these cases are precisely defined, the work within the NoC is effective.

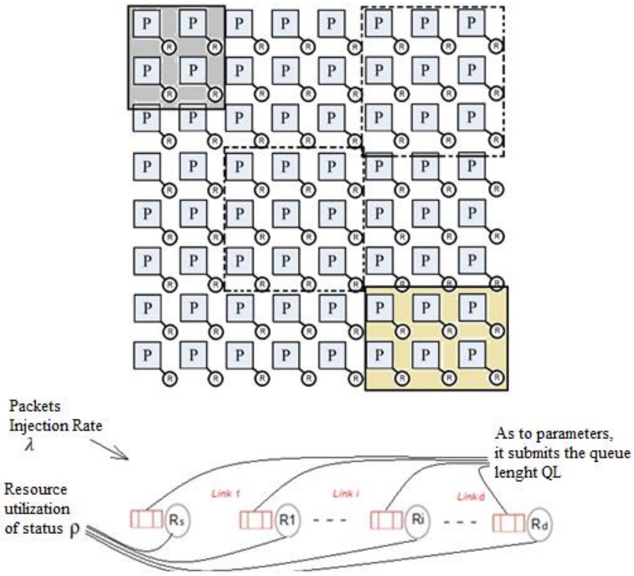


Fig. 2 Simple description of proposed model idea

In the research, an accurate identification of the application status was made by linking the need for processing of the subgroups formed by the PCDN division of features with the resource utilization rate. For the current application data, the rate is associated with the rate of arrival of the packets. In detail, step 5 is added to the PCDN algorithm to the implicit repetition t, a counter with the implementation period of each frequency (related to the frequency of the processing element in the NoC, Altera NIOS II 20MHz to 120MHz). The counter is with the implementation period and is based on the resource utilization rate. The number of subsets of characteristics will be reflected in the number of NoC nodes. It should be noted that this number will change with the training of the forecasting model used and the resource utilization rates will be standardized.

III. RESULTS AND DISCUSSION

SSIM and PSNR were applied to the proposed data set. These values were considered as features that characterize the analysis of the data structure. Therefore, mathematical relations were introduced into the Python code for analysis of dataset. The generated input was expanded from the dataset and values were added for a preliminary prediction of the difference SSIM_Ftur with the x and y locations for each event within the window in which the window was changed including WinPos_Ftur as well as the PSNR_Ftur factor that shows the difference. Once the edit is made, the data set on the Google Drive link can be linked to analysis software. Pydrive API is used with OAuth to obtain a special Client ID, after which Json is created and the project folder is linked to a local web server, Data through URIs and extracting information and placing them within the classifier used by PCDN.

According to the previously defined algorithm, the dataset was subdivided into random subgroups and examined the extent to which these features were included in the dataset. Table 1 presents a summary of the studied data in a way that illustrates the features selected in the research to guide the training and the results.

TABLE 1
SAMPLE OF IMPORTANT FEATURES WITH NUMERICAL DETAILS IN DATASET

Data set	SSIM_Ftur/n	WinPos_Ftur/n	PSNR_Ftur/n	Person/n	Car/n	Bus/n
CarCrash	0.007	0.323	0.156	0.112	0.092	0.051
Instances	Nssim_ftur	Nwinpos_ftur	Npnsr_ftur	Nperson	Ncar	Nbus
1,416	23	940	475	345	285	158
Features	N1ssim_ftur	N1winpos_ftur	N1pnsr_ftur	N1person	N1car	N1bus
336	3292	2910	3045	3069	3084	3120

The table 1 shows the composition of the proposed dataset, the number of entries performed in experiment 1,416, and the number of total features defined in the composition. The focus was on the studied features that were added to the classifier software, as well as the features that reflect the parts studied in the dataset. The WinPos_Ftur feature gives the largest representation of the dataset, especially since it is a compilation of all windows from the frames in which an event occurred. Fig. 3 shows the movement of features within the dataset. Therefore, retraining was carried out by dividing this feature according to specific windows, taking into account the studied parts. Some windows may be more suitable for some parts compared to other parts. The smaller window is acceptable to include the segment. The change locations of events are used and compared them with the change location of studied parts as to the same time period or gradually increased. From here, we can obtain complex properties that are very useful in reducing the processing and increasing its effectiveness.

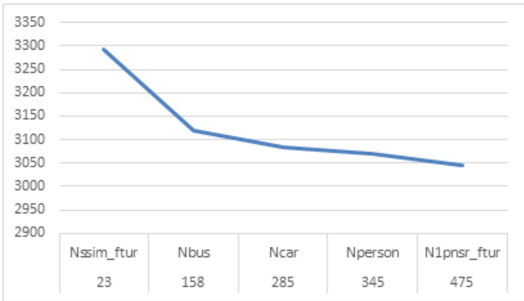


Fig. 3 Feature representations in dataset

The data segmentation is carried out on a method to easily train and achieve the target of linking between the features, it is Hierarchical Data Format (HDF5) [11]. This segmentation based on the size of the image taken from the frame. Fig. 4 shows different sizes of space. 3x3, 4x4, 5x5 and 10x10. Each area contains the average pixel values taken within this area. Thus, considering the change within each area and facility with its location within the main frame, the results of the training will be better and also the ability to parallel the work better.

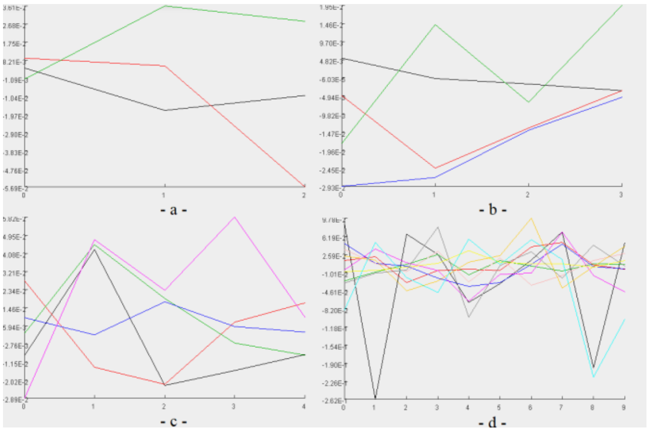


Fig. 4 Pixel values in variety sizes of studied areas: a- 3x3 , b- 4x4, c- 5x5, d- 10x10.

In order to study the change within the studied areas, additional variables were calculated through the relationship between PSNR and MSE. These variables help to know the threshold at which the image is changed, taking into account the image quality factor, and also focusing on the edges of the studied parts. The relationship between these variables was studied using the sample taken from the data set in Fig. 4. It was found that with the increase in the studied area, the PSNR decreases and the MSE increases, while the time required to process the image decreases. Therefore, the trend is to divide the data set and divide the properties into subgroups where the possibility of change is rapid.

In order to perform subdivision sets, must pay attention to the limited width of the packet for memory, so the data transfer rate must be reduced so that one processing item can be accessed for a single feature. This in turn sets the synchronization within linear search only within the subset. It should be noted that when dealing with features without subgroups, this makes access to the step in the linear search quick, i.e, with a few repetitions and at the specified accuracy $e=10^{-3}$, but that does not help to parallel the linear search process. Fig. 5 shows the relationship between the step function and the number of iterations and the size of the subgroup.

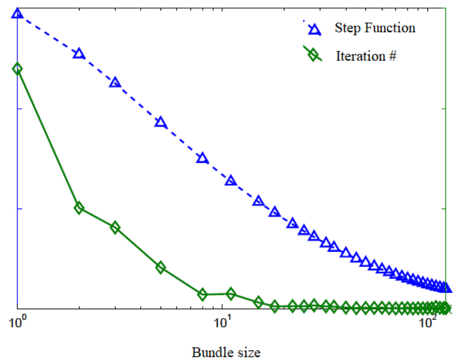


Fig. 5 Relation between Step function, iterations, and subgroup size.

This process is broken down into a subsets that include linked features, this makes the sub-step function minimum, and the PCDN algorithm will be stable with an acceptable number of iterations and the largest acceptable subgroup size to give the desired accuracy. The solution to this process is the essence of reaching an acceptable number of subgroups and acceptable sizes to make the parallel on the one hand and with the required accuracy.

Initially, a PCDN application was applied to the CarCrash data set without any modification in the classifier as input of new features or compound features as in Table 1. The HP DL380 Gen9 server was used with one processor. In some experiments, the number of subsets reached 1 group and a maximum size of 320. Therefore, the aim of introducing and conducting a new features or compound ones (7 application features were added) and overlapping to contribute to increase effectiveness after training.

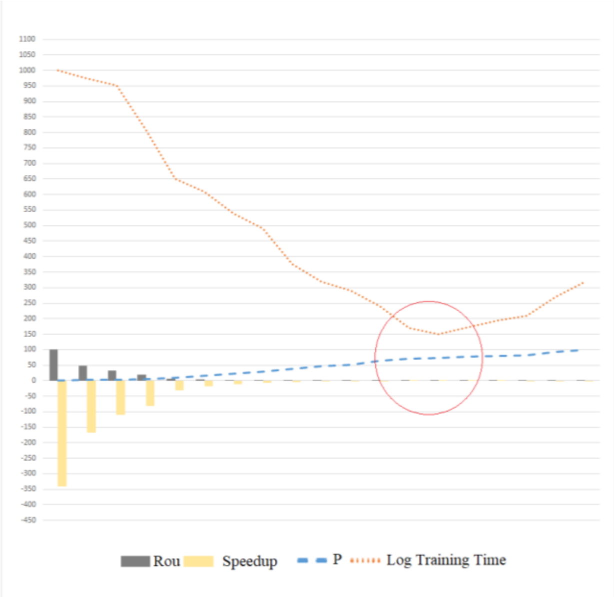
TABLE 2

SAMPLE OF FEATURES AFTER MODIFICATION AND TRAINING EVENTS

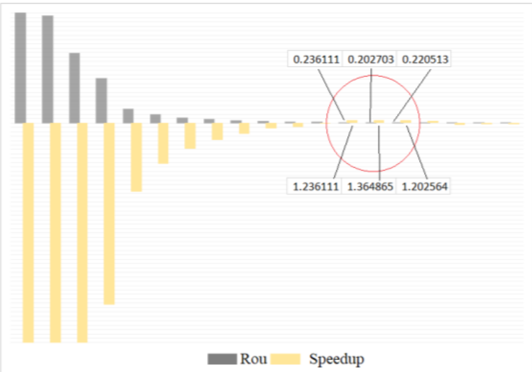
<table>
<tr><td>Data set</td><td>SSIM_Ftur/n</td><td rowspan="6">WonPos_Ftur As to 4 dimensions and as to timestamp of event</td><td>PSNR_Ftur/n</td><td>Person/n</td><td>Car/n</td><td>Bus/n</td></tr>
<tr><td>CarCrash</td><td>0.007</td><td>0.156</td><td>0.112</td><td>0.092</td><td>0.051</td></tr>
<tr><td>Instances</td><td>Nssim_ftur</td><td>Npnsr_ftur</td><td>Nperson</td><td>Ncar</td><td>Nbus</td></tr>
<tr><td>1,416</td><td>23</td><td>475</td><td>345</td><td>285</td><td>158</td></tr>
<tr><td>Features</td><td>N1ssim_ftur</td><td>N1pnsr_ftur</td><td>N1person</td><td>N1car</td><td>N1bus</td></tr>
<tr><td>336</td><td>3292</td><td>3045</td><td>3069</td><td>3084</td><td>3120</td></tr>
<tr><td>Added Furs</td><td colspan="3">Training Data (entries with positive events)</td><td colspan="3">Average positive events per feature</td></tr>
<tr><td>7</td><td colspan="3">703,696</td><td colspan="3">2,094</td></tr>
</table>

The PCDN was re-applied in the same manner with the modifications in the classifier when dealing with the dataset and the need for processing in the subsets and linking them to the NoC design shown in Fig. 2 where the NoC FPGA was used with 64 processing elements and a maximum number of subgroups not more than 64 Because when you increase this number it is difficult to run the experiment and get a result).

Fig. 6 shows that at the beginning of the work, dealing with 1 subset increases the resource utilization rate (ρ = Rou) with a reverse effect on the speedup, where the FPGA operates with one processing element but over time and training the number of subgroups starts to increase Up to 4 groups with an average size of up to 74, and at this moment the resource utilization rate is at its best with a significant speedup of approximately 1.3. These values are very important as subgroups start here with four elements of processing and thus start to exploit the Parallelism dynamically.



- a -



- b -

Fig. 5 Results: a- relation between resource utilization, parallel speedup, training time, subgroup size. b- zoomed speedup

IV. CONCLUSIONS

A built-in network on the NoC chip was developed to deal with the visual surveillance system, which varies according to the objects studied, the dimensions of the area in which they are moved, the values of parameters that assist in object identification, and the PCDN advanced learning algorithm, which can be subdivided with CarCrash. So that the NoC network can modify and distribute its processing capacity (64 elements of Altera FPGA) according to the system generated from the machine learning. Accelerated work was achieved up to 1.3, an important value for the distribution of work within the NoC.

ACKNOWLEDGMENT

Thanks for Tishreen University for helping in the remote research using the Altera board.

REFERENCES

[1] HOCHMUTH, *Worldwide Enterprise Mobility Management Software for Ruggedized/IoT Device Deployments*. IDC MarketScape, 2018, 29 Dec. 2018. https://www.ibm.com/downloads/cas/DJOADXBE

[2] I. TAJ-EDDIN; M. AFIFI; M. KORASHY; D. HAMDY; M. NASSER, S. DERBAZ, *A new compression technique for surveillance videos: Evaluation using new dataset*. Sixth International Conference on Digital Information and Communication Technology and its Applications DICTAP, Konya, 2016.

[3] D. R. PENG, *Advanced Statistical Computing*. Johns Hopkins Bloomberg School of Public Health, USA, 2018, 109.

[4] M. A. ASLAM; S. KUMAR; R. HOLSMARK, *An Efficient Router Architecture and Its FPGA Prototyping to Support Junction Based Routing in NoC Platforms*. Euromicro Conference on Digital System Design, 2013, 297-300.

[5] H. CECOTTI; B. GARDINER, *Classification of images using semi-supervised learning and structural similarity measure*. Irish Machine Vision and Image Processing Conference, 2016, 67-69.

[6] A. P. SHAH; J. LAMARE; T. NGUYEN-ANH; A. HAUPTMANN, *CADP: A Novel Dataset for CCTV Traffic Camera based Accident Analysis*. 15th IEEE International Conference on Advanced Video and Signal Based Surveillance AVSS, 2018, 1-9.

[7] Y. ZHUANG; Y. JUAN; G. YUAN; C. LIN, *Naive Parallelization of Coordinate Descent Methods and an Application on Multi-core L1-regularized Classification.* the 27th ACM International Conference on Information and Knowledge Management, 2018, 1103-1112.

[8] Y. A. BIAN; X. LI; Y. LIU; M. YANG, *Parallel Coordinate Descent Newton Method for Efficient L1-Regularized Loss Minimization.* IEEE Transactions on Neural Networks and Learning Systems, 2017.

[9] A. ALAEDDINI; D. J. KLEIN, *PSPO: Parallel Simultaneous Perturbation Optimization. Asia-Pacific Journal of Operational Research*, 2017.

[10] R. KASSAM; T. ALAATEKY; R. DANDAH, *Building analytical model for latency in Network-on-Chip using queuing theory.* Tishreen University Journal -Engineering Sciences Series, vol. 41, no. 12, 2019.

[11] T. KURTH; A. POCHINSKY; A. SARJE; S. SYRITSYN; A.WALKER-LOUD, *High-Performance I/O: HDF5 for Lattice QCD.* The 32nd International Symposium on Lattice Field Theory, Columbia University, New York, 2014.