Architecture for anomaly detection in a laser heating surface process

Javier Mesonero
Department of Artificial Intelligence
Universidad Politécnica de Madrid
Madrid, Spain
Email:javier.mesonero.perez@alumnos.upm.es

Concha Bielza
Department of Artificial Intelligence
Universidad Politécnica de Madrid
Madrid, Spain
Email: mcbielza@fi.upm.es

Pedro Larrañaga
Department of Artificial Intelligence
Universidad Politécnica de Madrid
Madrid, Spain
Email: pedro.larranaga@fi.upm.es

***Abstract*—Anomaly detection is an increasingly common task in many industrial environments. Cyber-physical systems stand out in this field due to their unique position in industrial areas. This paper introduces a new architecture aimed to detect anomalies in a real laser heating surface process, which is designed for field-programmable gate arrays (FPGAs). The FPGA design offers advantages of highly parallelized and pipelined architectures. The system will classify one process into normal or abnormal taking into account spatial information about where the laser spot is. The proposed design estimates a probability density function from data; then it performs an image convolution transforming the probability density function into a kernel density estimation function. This estimated function should be able to classify in real time.**

I. Introduction

Anomaly detection is the identification of anomalous events that have a behaviour outside the expected pattern. There are different approaches to solve this problem: looking for events that differ from the majority of the instances, labelling a data set as normal or abnormal and generating a classifier, or building a model that represents normal instances.

In this paper, a system of anomaly detection is aimed at finding errors in a laser surface heat treatment process. The goal of the laser surface heat treatment is to reinforce the metal surface by increasing the temperature without affecting the core. In this process it is considered that an anomaly has occurred when the surface is not heated at a desired temperature or the heating pattern does not behave as expected. Usually an event is considered anomalous according to an abnormality score: if the event score is higher than an established threshold the event will be considered abnormal. In general anomalies have an underlying negative meaning in results of the process.

The anomaly detection approach has been used in other laser applications. For example [7] used continuous hidden Markov models to identify anomalies in laser welding processes and [4] faced the problem using D-Markov machines. Moreover, [1] and [10] proposed different approaches to solve a similar problem over cyber-physical systems based on kernel density estimation (KDE) and Bayesian networks respectively.

This paper aims to continue the research proposed by Atienza et al. in [1] building an architecture which learns a KDE model taking into account laser spot spatial information.The proposed architecture can be integrated in field programmable gate array (FPGA) applications. FPGAs are widely used in cyber-physical systems because they can be easily reconfigured, and thus we focus our work on them.

In order to build KDE models it is necessary to estimate probability density functions (PDFs). PDF estimation has already been tackled in some proposed FPGA architectures. For example, in [5], they created a one dimensional non-parametric PDF to extract statistical information in real time to be incorporated into existing FPGA applications. They also presented an architecture for non-parametric PDF estimation using one dimensional histograms and kernel-based methods using histograms [6]. Another interesting approach was presented in [9], designing a multi-core PDF estimation using Gaussian kernels.

This work presents a novel architecture to detect anomalies by estimating PDFs, with two dimensional histograms and kernel-based methods to approximate KDE models. The architecture is focused on achieving a highly parallel and pipelined design.

II. Data

A. Data description

The input data used to the proposed architecture was gathered from a thermal camera (NIT Taychon 1024 μCore@1000 fps, 32×32 pixels) that recorded 32 processes from a real laser surface heat treatment applied to cylindrical steel workpieces. Each pixel value is bounded between 0 and 1023 being equivalent to its temperature. Each video contains 21,500 frames, so that each video shows 21.5 seconds of the process for each workpiece. Nevertheless, the laser spot is only visible on around 20700 of these frames though, with some variation in this number between recordings. The recorded surface of the workpiece is 10×20 mm^2. The laser spot during the surface heating process draws an eight-like fixed pattern with a 100Hz frequency. Furthermore, during approximately 3,800 frames, the patterns are modified to avoid a hole in the cylindrical surface that cannot be heated by the laser.

978-1-5090-6505-9/17/$ 31.00

B. Data preprocessing

Before the laser spot movement is processed, we have to obtain the laser spot position in each frame. To do this, we first obtain the differences between contiguous frames for the entire video. The result is a subtraction video that shows the variation in the surface temperature across time. As the laser spot is moving, it applies energy to different areas of the surface. The regions that are heated by the laser spot exhibit higher pixel values in the subtraction video. Then, we compute the centroid of the regions of the surface with higher pixel values in the subtraction video. This gives us the coordinates of the laser spot position on each frame [1]. Finally, we range the pixel values between the lower and upper bounds of histogram bins in the PDF, that is we discretize the values and truncate them to fit in the PDF range.

III. Methodology

The proposed methodology to find anomalies is based on the spatial characteristics of the laser spot position. The laser spot position data is used to build a histogram based PDF. Other works are focused on estimating one dimensional PDF [6]. This novel architecture estimates a bivariate PDF using the spatial information in a laser process.

The circuit is divided in three mayor blocks as shown in Fig 1. The first one estimates the PDF taking the laser spot position as input and builds a two dimensional histogram. The second block makes a kernel operation on the PDF, building a KDE model. Finally, the third block performs an evaluation, comparing the new computed KDE model and a pre-estimated not-anomalous KDE model, which results in an anomaly score.

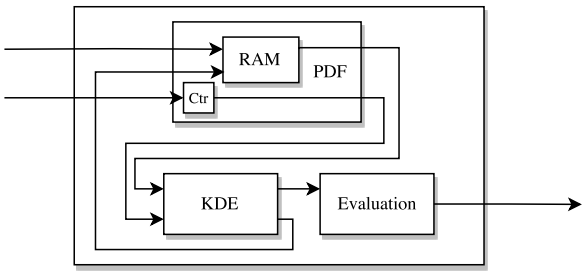

Fig. 1. Architecture design overview.

A. Probability Density Function Estimation

The circuit aimed at holding the PDF is relatively direct. An embedded block RAM with $n \times n$ positions is used, with n equals to the number of bins on one histogram side. The input data is composed of the laser position x and y values indexing a two dimensional point in a one dimensional memory block. This simplification allows to avoid complex direction indexing which can save many cycles.

In each cycle a new input data is stored in its corresponding bin of the histogram during PDF estimation. Also, a counter takes into account the number of samples that are already gathered in memory. The data storing process takes two pipelined cycles: the first cycle loads the current bin value, the second cycle adds one to the load value and store it in its corresponding bin. The nature of the block RAM must be taken into consideration when when two or three consecutive input data values are equal, since it may cause that only one input value is considered rather than two. Another consideration that is worth to mention is in regards to the value n and the word length for each position. The two separated considerations, will affect the FPGA resource consumption. In this context we selected n equal to 32 and a word length equal to 16. These choices take into account the laser heating surface process particular characteristics.

B. Kernel Density Estimation as an Image Convolution

To have a KDE model, a kernel operation is needed in each data point. This means that for each new point in the PDF, adjacent bins should be increased in a kernel defined proportion. There are two ways to do this operation. The first one consists of storing also the new values for adjacent bins for each new data points. An example of this method is found in [6], who used counters to store the PDF and allowed them to increase several bin values in the same clock cycle. In this novel architecture it is not possible due to RAM nature which need one cycle for each load or store operation, so a second way to solve this problem is proposed here: first build a PDF estimation and then proceed to convolve a kernel operation over all the data stored in the histogram PDF.

To do this in a hardware architecture we construct a 5×5 kernel convolver. There are many examples and literature that attempt to solve this problem [11], [2]. An important subject to take into account is related to the convolution operation in the histogram edges and how to operate with the kernel pixels that fall out of the histogram. In order to achieve the best performance and minimum complexity a mixed solution is carried out. In the top and bottom border a zero-padding solution is adopted, meaning that the kernel values which fall out of the histogram shape are equal to zero and do not affect the final pixel value. The left and right borders consider previous rows pixels, so a wrap-around method is implemented so that kernel pixels which fall out of the histogram shape take values of pixels on the other side as shown in Fig. 2.

0	0	0	0	0
P_1	P_2	P_3	P_4	P_5
P_6	P_7	P_8	P_9	P_{10}
P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
P_{16}	P_{17}	P_{18}	P_{19}	P_{20}
P_{21}	P_{22}	P_{23}	P_{24}	P_{25}
0	0	0	0	0

Fig. 2. Proposed mixed method for kernel operations in the borders for a 5×5 image with a 3×3 kernel.

The hardware architecture consists on three mayor modules: a FIFO queue, a convolution module and a control block, as show in Fig 3. The FIFO queue needs at least

$$Mininum\ required\ pixels = n \times (k-1) + k \quad (1)$$

with k equals to the kernel size, i.e. the available directions to store pixels values during the convolution [3]. In the proposed architecture $k = 5$, $n = 32$ and $Mininum\ required\ pixels\ (M) = 133$. The FIFO queue is designed as an embedded block RAM memory. Due to this design it is not possible to use the exact previous value in 1 so that a power of two approximation is needed. We can approximate it with the formula:

$$FIFO\ queue\ length = 2^{\log_2(n)+\lceil \log_2(k) \rceil} \quad (2)$$

This will ensure that the minimum value that fulfils the size requirement is used. In this particular application the FIFO queue size will have 256 addressable directions, with $n = 32$ and $k = 5$.

The convolution module performs a 5×5 kernel operation in two dimensions. The hardware design inputs are the 25 pixel values of the convolution. It takes 5 cycles to perform all the convolution operations. Each cycle multiplies the pixel values by the kernel weight and accumulates the result. Finally, in the fifth cycle the final convolution result is returned. The hardware design includes five multiplexors and five multipliers. This reduces the necessary resources on the FPGA. The convolution module also has the kernel definition. It is possible to operate with kernels smaller than 5×5, but it is not recommended due to the efficiency loss, since it always needs five cycles to complete the convolution.

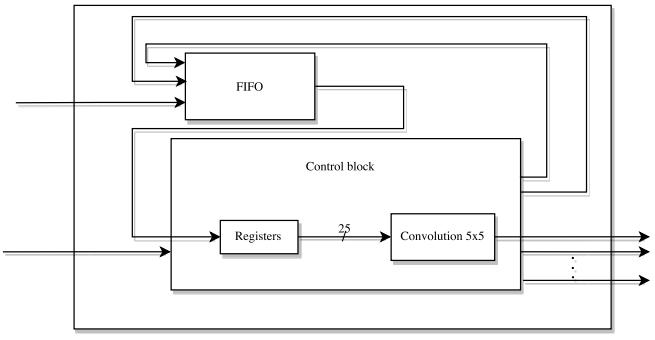

Fig. 3. Architecture design overview for Kernel Convolution module.

The control block is the convolution's "brain", it connects different image convolution parts, takes into account specials situations like the first and last cycles, also ensure the correct data flow between PDF RAM, FIFO queue and convolution module. During the first cycles, the FIFO queue will store $2\times n$ zeros and the first M pixels, one per cycle. Once this charge phase is finished it will start to produce results. It takes seven cycles per pixel or bin to return its convolved value. Thus, the whole process takes a number of cycles equals to:

$$7n^2 + M + 2n \quad (3)$$

The regular convolution phase only needs 7 cycles because in the Control block there are 20 registers that store the used values extracted from the FIFO in previous cycles, so that it only performs 5 loads from the FIFO queue to these internal registers. This explains the regular convolution takes five cycles in the middle phase. During the first cycle a push is performed in the registers causing that the five oldest values are popped out and replaced by new ones, as show in Fig 4.

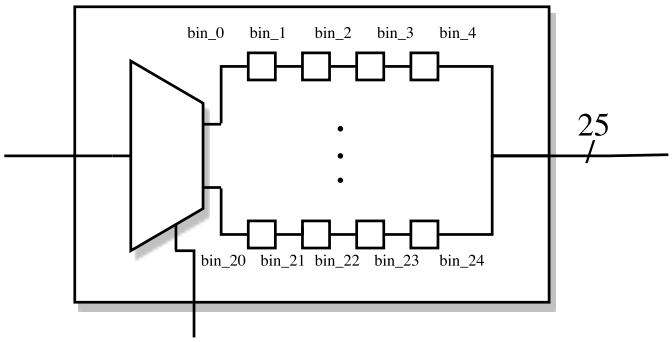

Fig. 4. Internal register structure to store values during convolution.

The underlying hardware architecture works as follows: once the first necessary values are stored in the FIFO queue it starts to work in a seven cycle loop. The first cycle performs a push in the internal registers. During the next five cycles, new values are stored in registers waiting until the next push to enter the convolution registers. In parallel a previous value kernel convolution computation is being performed returning the final result in the sixth cycle. The seventh (last) cycle updates the PDF and FIFO queue direction values and also keeps track of how many bins have been calculated so far.

C. KDE Model Evaluation

Once the KDE model is built the next step is to classify the test instances in normal or abnormal. The test and train KDE models are compared and the returned result is called abnormality score. We propose two comparative methods to obtain the score: one focused on achieving a fast performance, conceptually easy and light in resource consumption; and the other method uses the same method proposed in [1], based on the Kullback-Leibler (KL) divergence [8].

The first method is a simple distance point by point between the test KDE and the train KDE, calculating the difference for each point in absolute value and then summing them all. Then the result is normalized dividing by the total number of histogram points and the sum of the used kernel values, $k_{i,j}$.

$$AbsDiff = \frac{\sum |x_{trained} - x_{test}|}{numberOfPoints \times \sum k_{i,j}} \quad (4)$$

In order to continue and compare this work with the method proposed in [1] we also implemented the KL divergence. Additionally it offers a more robust and mathematically based approach to the estimation of the differences between two KDE models.

For two probability distributions P and Q, the Kullback-Leibler divergence from any distribution Q to a reference distribution P is defined as:

$$D_{KL}(P\|Q) = \sum P(i) \log \frac{P(i)}{Q(i)} \quad (5)$$

The trained model stored in the ROM and the test model represented by the KDE function will be the P and Q distributions respectively.

The proposed architecture algorithm follows a mathematically logical path. First, it gathers KDE values, one from the convolution module and another from a pre-calculated embedded block ROM which contains the trained KDE model. Then data inputs enter a pipelined sequence of mathematical blocks implementing the necessary operations. First the integer inputs are transformed into floating point type. Afterwards, a division, a logarithm, a multiplication and an accumulation take place. Each mathematical block has a seven cycle latency to adjust to the latency set by the convolution module. Hence, when a new input comes its divergence will be added to D_{KL} value, 35 cycles later, and when the last bin in the KDE model is convolved, the KL module needs 35 cycles to return the final D_{KL} value result.

IV. Results

The aim of this paper is to propose a fast approximation to build KDE functions. Thus, the tests measure how fast the proposed architecture is and how much speed-up it achieves versus other proposed works. Because of the laser process reliability there are no anomalous data.

In order to measure the proposed work's performance, a complexity study is carried out. The total time in unit cycles will be the sum of the three modules. The first module will take $s + 2$ cycles, with s equals to the number of data inputs. The second and third modules are measured as one module since they run in parallel. Therefore, the second and third total cycles will be $7n^2 + M + 2n$ cycles as seen in (3), plus 35 cycles of the evaluation model (using KL divergence method). The entire process takes a of $7n^2 + M + 2n + s + 36$ cycles. In the laser process problem with $n = 25$, $k = 5$ and $s = 21500$, it takes $28,901$ cycles to finish one complete evaluation process. At a clock frequency of 100MHz it will take 5.78 milliseconds.

To compare this result we calculated how many cycles would take to complete the same problem when implementing the kernels point by point. For each arriving point it will take k^2 cycles to store the data into the PDF, and it will take $7n^2 + M + 2n + 35$ to evaluate the model. Hence, the total cycles needed are $s \times k^2 + 7n^2 + M + 2n + 35$. With the same values for evaluation it will finish in $544,895$ cycles.

Comparing the two methods a 18,8$\times$ speed-up is achieved. This speed improvement is based on the premise that $s >> n$.

Since we lack anomalous data, a test with simulated examples was carried out. The pattern is transposed and its KL score is calculated. The results showed that these anomalous patterns have scores up to 33 times higher than non anomalous ones. The same simulated test was performed using the point by point distance. The obtained anomalous score is up to 2.5 times higher in the transposed pattern. The KL divergence comparison method performs better as expected.

Comparisons with others works are not possible because the other papers are focused only on estimating PDF, whereas in this work we also used it for anomaly detection.

V. Conclusion

We developed a novel approach to anomaly detection taking into account spatial information. The architecture uses a kernel-based function to output an abnormality score. Complexity magnitude studies showed how the convolution implementation improves the performance against a brute force built kernel-based function.

The proposed architecture can work with any integer value ranged between the lower an upper bound. This means that the input values could also be the position given by the laser mechanical pieces that control the spot during the pattern.

There are many potential future lines of improvement this architecture. New parallel implementations might be investigated which improve how the cost scales with problem size.The FPGAs parallel and easily pipelined nature can reduce significantly the necessary cycles when building the PDF, which takes almost 75% of the amount of cycles. New ways to calculate the abnormality score will also worth investigating.

Acknowledgment

This research has received funding from the Spanish Center for the Development of Industrial Technology (CDTI) as part of project TIC-20150093 and partial funding from the Spanish Ministry of Economy and Competitiveness as part of project TIN2016-79684-P and from Madrid Regional Government as part of project S2013/ICE-2845-CASI-CAM-CM.

References

[1] Atienza, D., Bielza, C. and Larrañaga, P. Anomaly detection with a spatio-temporal tracking of the laser spot. Proceedings of the Eight European Starting AI Researcher Symposium (STAIRS 2016), pp. 137-142, 2016.

[2] Bailey, D. G. *Design for Embedded Image Processing on FPGAs*. John Wiley and Sons, 2011.

[3] Benedetti, A., Prati, A., and Scarabottolo, N. Image convolution on FPGAs: the implementation of a multi-FPGA FIFO structure. Proceedings of the 24th Euromicro Conference, Vol. 1, pp. 123-130, 1998.

[4] Chinmay, R., Asok, R., Soumik, S. and Murat, Y. Review and comparative evaluation of symbolic dynamic filtering for detection of anomaly patterns. Signal, Image and Video Processing, Vol. 3, No. 2, pp. 101-114, 2009.

[5] Fahmy, S. A. Histogram-based probability density function estimation on FPGAs. 2010 International Conference on IEEE,Field-Programmable Technology (FPT), pp. 449-453, 2010.

[6] Fahmy, S. A., and Mohan, A. R. Architecture for real-time nonparametric probability density function estimation. IEEE Transactions on Very Large Scale Integration Systems, Vol. 21, No. 5, pp. 910-920, 2013.

[7] Jager, M., Knoll, C. and Hamprecht, F. A. Weakly supervised learning of a classifier for unusual event detection. IEEE Transactions on Image Processing, Vol. 17, No. 9, pp. 1700-1708, 2008.

[8] Kullback, S. and Leibler, R. A. On information and sufficiency. The Annals of Mathematical Statistics, Vol. 22, No 1, pp. 79-86, 1951.

[9] Nagarajan, K., and Sivakumar, M. Design and analysis of a multi-core PDF estimation algorithm on FPGAs. Proceedings of the 24th Euromicro Conference, Vol. 1, pp. 123-130, 1998.

[10] Ogbechie, A., Díaz-Rozo, J., Larrañaga, P., and Bielza, C. Dynamic Bayesian network-based anomaly detection for in-process visual inspection of laser surface heat treatment. Machine Learning for Cyber Physical Systems, pp. 17-24, Springer, 2016.

[11] Ström, H. *A Parallel FPGA Implementation of Image Convolution.*. MSc Thesis, Linköping University, 2016.