International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 7
July 2017

Academic Science

GPU based human detection for Video Surveillance Applications

1.Maddimsetti Srinivas
Department of Electronics and Communications Engineering
KL University, Guntur, Andhra Pradesh, India 522502
2.Kallepalli Bapayya
Department of Electronics and Communications Engineering
KL University, Guntur, Andhra Pradesh, India 522502
3.T Sharath Chandra
Department of Electronics and Communications Engineering
CVR College of Engineering Hyderabad, India

Abstract—Present work targets the detection of humans in images and videos. Our focus is on developing robust feature extraction algorithms that encode image regions as high dimensional feature vectors that support high accuracy human/non-human decisions. To test our feature sets we adopt a relatively simple learning framework that uses linear Support Vector Machines to classify each possible image region as a Human or as a non-Human. This work makes three main contributions. Firstly, we introduce grids of locally normalized Histograms of Oriented Gradients (HOG) as descriptors for people detection in static images. The HOG descriptors are computed over dense and overlapping grids of spatial blocks, with image gradient orientation features extracted at fixed resolution and gathered into a high dimensional feature vector. They are designed to be robust to small changes in image contour locations and directions, and significant changes in image illumination and color. Secondly, the Human detection algorithm has to be executed on modern parallel platforms to achieve the detection goal in real time. Thirdly, alert video monitoring station with an indication.

Keywords—HOG, Feature extraction, Human detection, SVM classifier, GPU, RTSP.

I. INTRODUCTION

With the increasing danger of crime, video surveillance attracts much more attention and it has been adopted in different applications for crime controlling and detection of doubtful person. The advanced video surveillance system requires analyzing the behavior of the people in order to detect the incident of the possible dangerous cases, But till now video surveillance paired with security guards which means security guards should watch surveillance TV at 24/7 days it made more pricy.

Computer vision is a research field that enables machines to see, understand, and analyze the real world. One straightforward usage of this technology is in robotics. If the robots have the same views and the same interpretations as human beings, then they can behave more like human beings. As long as the machines can collect data about the world to analyze and find objects in the direction they were programmed The easiest way for machines to get snapshots" of the real world is by collecting images and videos. Therefore, most computer vision research focuses on image and video analysis.

When given an image or video, one way that human beings understand the context is by identifying objects inside the scenes. Similarly, if machines are able to recognize objects in scenes, they might be

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 7
July 2017

able to understand the contexts as well. Therefore, object recognition is a major branch of computer vision research.

Human beings are not born with the ability to recognize objects; by learning the ability to recognize objects is achieved. Similarly, machines need to learn before they can identify objects. The object recognition problem is therefore solved in two stages: a training stage that trains machines to be familiar with objects, and a deployment stage that asks machines to recognize objects from images or videos. In the training stage, for each object, a collection of example images or videos is provided containing instances of the object. Special features are extracted from the ex-ample images or videos to represent the characteristics of the object instances.

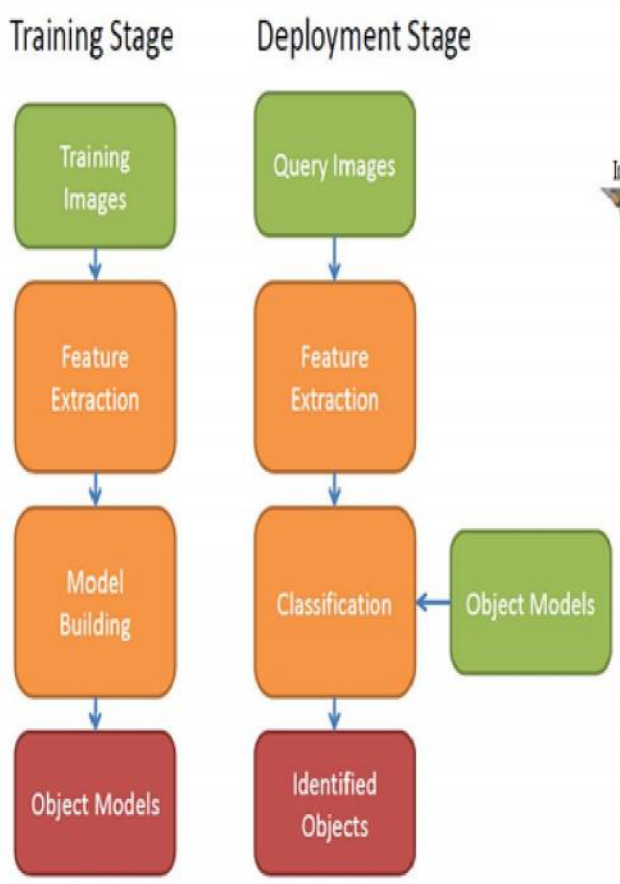

Fig. 1: Object recognition computation flow.

Based on the features of the example object in-stances, a model is built to recognize the object. In the deployment stage, the features are extracted and plugged into that model. The model can then decide whether instances of the object are found in a given image or video.

A new hope for solving the computational challenge of the object recognition problem is to parallelize the algorithms and execute them on modern parallel platforms. This will solve the problem of real time object recognition and enhance the Surveillance cameras purpose. In both military and civilian domains, there is thus a clear benefit to deploying machines which can perform automated situational awareness tasks.

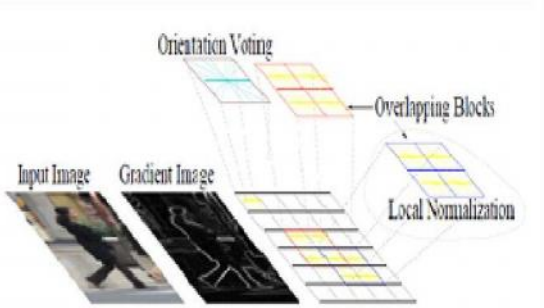

Fig. 2: HOG extraction general view - graphical description of all the extraction steps.

II. HOG FOR OBJECT RECOGNITION

The main goal of this chapter is to explain an outperforming descriptor for object recognition. Both from a theoretically point of view as from a more pragmatic and computationally point of view. As has been said in the previous sections of this thesis the real challenge is finding a descriptor that performs equally well independently of illumination variations and within a relatively wide range of poses variations that can be adopted by pedestrians, this is a descriptor able to cleanly discriminate between pedestrians and not pedestrian even in cluttered background in difficult illumination environments. After examining all this requirements

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 7
July 2017

Academic Science

Dalal and Triggs [4] end up with a feature called Histogram of Oriented Gradients.

A. *Introduction*

The histogram of oriented algorithm for object detection was introduced in [4]. An example of detection results for pedestrians and heads is visible in Figure 1.The HOG detector is a sliding window algorithm. This means that for any given image a window is moved across at all locations and scales and a descriptor is computed. For that window a pertained classifier is used to assign a matching score to the descriptor. The classifier used is a linear SVM classifier and the descriptor is based on histograms of gradient orientations.

B. *Extracting HOG from an image*

Reviewing now a complete definition of the descriptor, the key points of its obtainment and the main configuration used in this thesis.

Brief overview of HOG descriptors:

The main idea behind HOGs is that the local appearance and shape of an object in an image can be characterized as the intensity gradient distribution, this is, by the edges directions. This characterization can be accomplished by dividing the image in little and connected regions, named cells, from within the gradients are computed. Roughly speaking to make the descriptor we should join all the gradients computed through all this regions and count the occurrences of the orientations. Anyway further improvement can be achieved by defining a larger area, called block, where cells are grouped. This blocks are then used to compute a intensity measure of the whole block, used no normalize all the cells within the blocks. Proceeding like this makes the descriptor more invariant to illuminations changes.

Windows, Blocks, and Cells:

The subdivision of the detection task into windows, blocks, and cells suggests that this algorithm will benefit from a parallel processing implementation. Most efficient processing is possible when the overlap between each pair of windows or blocks is an integer multiple of the width of the cells. In this case, cell data can be reused extensively across the various windows and blocks. In our implementation, the detection window was a 64 x 128 pixel window divided into a grid of 8 x 8 pixel cells. Blocks consisted of a 2 x 2 square of cells; each detection window contained 7 x 15 blocks (which overlapped by one cell). The stride between detection windows was set at 8 pixels.

Gradient Computation:

Therefore in this thesis we center in the evaluation of the best choice to compute gradient by applying [-1; 0; 1] for the horizontal direction and [-1; 0; 1]T for the vertical direction.

The equations defining the gradients are

$$G_x(i,j) = \frac{\partial I}{\partial x}(i,j) \qquad (1)$$

Bin number	Orientation degrees range
0	[0, 40)
1	[40, 80)
2	[80, 120)
3	[120, 160)
4	[160, 200)
5	[200, 240)
6	[240, 280)
7	[280, 320)
8	[320, 360)

Fig. 3: Binning Numbering.

$$G_y(i,j) = \frac{\partial I}{\partial y}(i,j) \qquad (2)$$

The gradient magnitude is then computed as the square root of the quadratic sum of each gradient component, this is

$$M_x(i,j) = \sqrt{G_x^2(i,j) + G_y^2(i,j)} \qquad (3)$$

The angle can be calculated as the four-quadrant inverse tangent of Gy and Gx

$$\theta_x(i,j) = \arctan \frac{G_y(i,j)}{G_x(i,j)} \qquad (4)$$

Spatial/Orientation binning: Each pixel calculates a weighted vote for an edge orientation histogram channel based on the orientation of the gradient element centered on it, and the votes are accumulated into orientation bins over local spatial

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 7
July 2017

regions that we call cells. Cells can be either rectangular or radial (log-polar sectors). The orientation bins are evenly spaced over 0 to 180 (unsigned. gradient) or 0 to 360 (signed gradient). To reduce aliasing, votes are interpolated bilinearly between the neighboring bin centers in both orientation and position. The vote is a function of the gradient magnitude at the pixel, either the magnitude itself, its square, its square root, or a clipped form of the magnitude representing soft presence/ absence of an edge at the pixel. In practice, using the magnitude itself gives the best results. Taking the square root reduces performance slightly, while using binary edge presence voting decreases it significantly.

Normalization and descriptor Blocks: Gradient strengths vary over a wide range owing to local variations in illumination and foreground-background contrast, so effective local contrast normalization turns out to be essential for good performance. We evaluated a number of different normalization schemes. Most of them are based on grouping cells into larger spatial blocks and contrast normalizing each block separately. The final descriptor is then the vector of all components of the normalized cell responses from all of the blocks in the detection window.

Detector window: All the blocks inside a detection window form a HOG descriptor. This is used as input for a pretrained linear SVM classifier. For people detection the best results have been achieved using 16x16 pixel

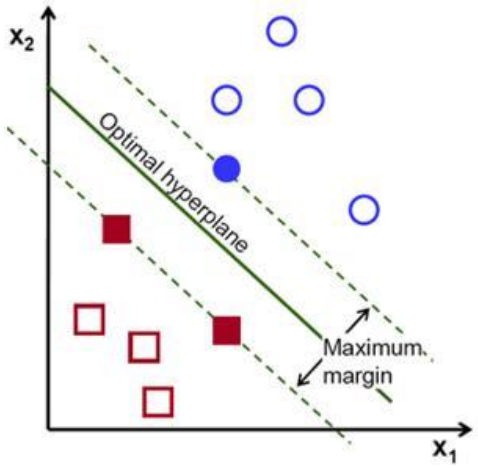

Fig. 4: support vector machine.

blocks containing 2x2 cells of 8x8 pixels. The block stride is 8 pixels (one cell) so the histogram of a cell is normalized over 4 blocks. Each histogram has 9 bins. The detection window is 64 x128. The scale ratio is 1.05.

SVM Classifier: A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm out-puts an optimal hyperplane which categorizes new examples For a linearly separable set of 2D-points which belong to one of two classes, find a separating straight line. Then, the operation of the SVM algorithm is based on finding the hyperplane that gives the largest minimum distance to the training examples. Twice, this distance receives the important name of margin within SVMs theory. Therefore, the optimal separating hyperplane maximizes the margin of the training data.

III. HISTOGRAMS OF ORIENTED GRADIENTS ON THE GPU

Introduction to GPU Computing: The first GPUs were designed as graphics accelerators, supporting only specific fixed-function pipelines. Starting in the late 1990s, the hardware became increasingly programmable, culminating in NVIDIA's first GPU in 1999. Less than a year after NVIDIA coined the term GPU, artists and game developers weren't the only ones doing ground-breaking work with the technology: Researchers were tapping its excellent floating point performance. The General Purpose GPU (GPGPU) movement had dawned.

CUDA Programming Model:

CUDA programmers develop code for the GPU by creating C functions called kernels. Only one kernel can be run on the device at a time, and all configured threads execute the kernel in parallel. The threads are grouped

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 7
July 2017

into thread blocks which are then organized in a grid as seen in Figure 2.9 below. Threads in a block can cooperate (they can share memory and can be synchronized) while blocks in a grid are independent.

Most CUDA applications follow a set program flow. The host first loads data from a source such as a text file and stores it into a data structure in host memory. The host then allocates device memory for the data and copies the data to the allocated space. Kernels are then launched to process the data and produce results. These results are then copied back to the host for display or further processing.

Parallel Algorithms:

Our algorithm can be split into 2 parts: host and GPU processing (as shown in Figure 5). The image is acquired by the host, copied into GPU memory and then processed by the GPU. After all scales and windows are processed, the SVM scores are transferred back to the CPU, where they are formatted. A formatted result contains information about the position of the window (x, y, scale) and its score. The non-maximal suppression algorithm is executed (using the host CPU) on the formatted data, in order to obtain the final fused results [9].

IV. IMPLEMENTATION DETAILS

LIVE555 Streaming Media has a media server with RTSP protocol which is having a collection of .264 files will act as camera (we are requesting frames in a video file with is requested by the OpenCV client application).

LIVE555 Streaming Media community have Source-code libraries for standards based RTP/RTCP/RTSP/SIP multimedia streaming, suitable for embedded and/or low cost streaming applications. ”openRTSP” is a command-line program that can be used to open, stream, receive

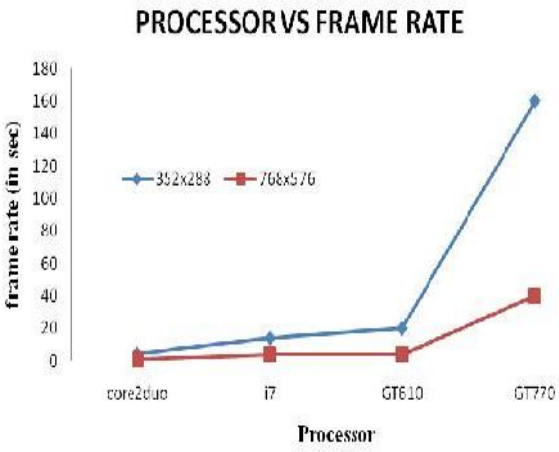

Fig. 7: Processor Vs Frame rate

and (optionally) record media streams that are specified by a RTSP URL.

V. RESULTS:

Speed comparisons Due to the hog detection is a sliding window based Algorithm detection depends on the frame resolution and the clock rate of the system. By using parallel algorithm with GPU, HOG algorithm suits for real time requirements.

Performance Comparisons: On GT 770 the frame rate and detection window performances with respect to scale.

VI. CONCLUSION AND FUTURE DIRECTIONS:

The development process of Human detection has been successfully developed and tested. The project developed on OpenCV 2.4.9 with the help of CUDA 6.0

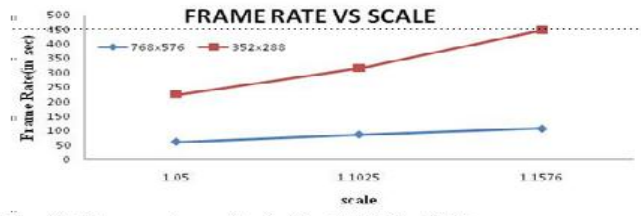

Fig. 8. Framerate vs Scale for GT 770 GPU

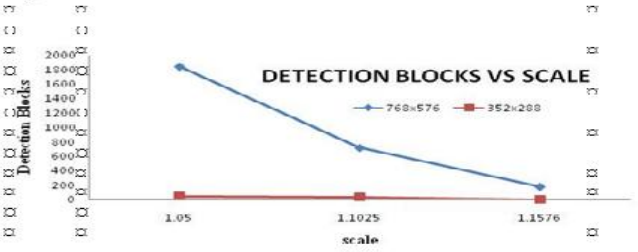

Fig. 9. Detection blocks Vs Scale for GT 770 GPU

in LINUX operating system. In this project The system streams the video file from a media server through RTSP protocol with the help of video capture class in OpenCV package and detects humans at all conditions with the help of detect multi scale operation on image.

Surveillance system helps to monitor a given area of interest. Multiple cameras are used to cover a large area. In order to track objects successfully in multiple cameras, one needs to handshake among objects captured in multiple cameras.

REFERENCES

[1] J. S. Warm, R. Parasuraman, and G. Matthews, Vigilance Requires Hard Mental Work and Is Stressful,Hum. Factors J. Hum. Factors Ergon. Soc., vol. 50, no. 3, pp. 433441, Jun. 2008.

[2] N. H. Mackworth, The breakdown of vigilance durning pro-longed visual search,Q. J. Exp. Psychol.,vol. 1, no. 1, pp. 621, Apr. 1948.

[3] F.M. Donald, The classification of vigilance tasks in the real world.Ergonomics, vol. 51, no. 11, pp.164355, Nov. 2008.

[4] Navneet Dalal and Bill TriggsHistograms of oriented gradients for human detection.Computer Vision and Pattern Recogni-tion, IEEE Computer Society Conference on, 1:886893, 2005.

[5] David G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60:91110, 2004

[6] Christian Wojek, Gyuri Dorko, Andre Schulz, and Bernt Schiele. Sliding-windows for rapid object class localization: A parallel technique. In Proceedings of the 30th DAGM symposium on Pattern Recognition, pages 7181, Berlin, Heidelberg, 2008. Springer-Verlag.

[7] NVIDIA CUDA (Compute Unified Device Architecture), http://www.nvidia.com/object/cuda home.html

[8] gpu.GPU accelerated → Computer → VISION, http://docs.opencv.org/modules/gpu/doc/gpu.html

[9] Victor Adrian Prisacariu, Ian Reid fastHOG - a real-time GPU implementation of HOG University of Oxford, Department of Engineering Science

[10] Dilgic, B. Dept. of Electr. Eng. and Comput. Sci., MIT, Cam-bridge, MA, USA Horn, B.K.P.; Masaki,Fast human detection with cascaded ensembles on the GPU, Intelligent Vehicles Symposium (IV), 2010 IEEE Conference on 21 24 June: 325 332,2010

[11] C. I larris and M. Stephens. A combined corner and edge detector .In Alvey Vision Conference pages 147151, 1988.

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 60(2):91.110, 2004

[13] SourceForge.net: Open source Computer Vision Library, http://sourceforge.net/projects/opencvlibrary

[14] M.J swain and D.H ballard, "color indexing", intl J comput vis 7(1),11-32(1991).

[15] B.M.Mehtre, M.S.kankanhalli, A.D Narasimhulu and G.C.Man, "color matching for image retrieval", pattern recognition let 16,325-331(march 1995).

[16] The image database is available on http://www.cs.cmu.edu/~cil/v-images.html

srinivas Mr. M.Srinivas completed his Master of Technology at IIT kharagpur in Visiual Information and Embedded Sys-tems Engineering in 2011.He worked as Visiting faculty at LNMIIIT, Jaipur till 2012. He is currently working as an Assistant professor in ECE Department of C.V.R COLLEGE OF ENGINEER-ING,HYDERABAD. His research interest includes Computer vision, Video coding Standards, Medical imaging.

Bapayya Mr. K.Bapayya was born in 1983 .He received his B.tech in ECE from Jawaharlal Nehru Technological Univer-sity, Hyderabad in 2005. He has com-pleted his Master of Technology at Gudlavalleru Engineering College, Gud-lavalleru in 2007.He is currently work-ing as an Assistant professor in ECE Department of C.V.R COLLEGE OF ENGINEERING,HYDERABAD. His re-search interest includes Embedded Sys-

tems, Communications, FPGA's and Image processing.

sharath T.sharath chandra was born in jagtial, karimnagar in 1989.he received the B.tech degree in electronics and commu-nication engineering from the scient insti-tute of technology, jntu university in 2010, and the M.tech degree in embedded sys-tems from the CVR college of engineering in 2014. his research interests includes computer vision, parallel programing and embeddable electronics.