# D-face: Parallel Implementation of CNN Based Face Classifier using Drone Data On K40 & Jetson TK1

Harish, Ayyappa, Santosh and P K Baruah
*Department of Mathematics and Computer Science*
*Sri Sathya Sai Institute of Higher Learning(SSSIHL)*
*PrashantiNilayam, India*
*harishatchutanna@gmail.com and pkbaruah@sssihl.edu.in*

Kaliuday Balleda and Sairam K M Menon
*MulticoreWare Inc*
*California (HQ), 530 Lakeside Drive, Suite #140*
*Sunnyvale, CA 94085*
*kaliuday, sairam@multicorewareinc.com*

***Abstract*—Deep Convolutional Neural Networks(CNNs) are shown to perform very well in the areas such as video surveillance, object classification and face classification. Face classification has become pertinent to numerous applications, especially in this big data era of social platforms and social media. With the usage of unmanned air-borne vehicles like drones, the problem of face classification becomes very challenging because of the large visual variations.**

**In this paper we introduce D-face, a novel application for face classification, and the input to the classification model is taken from the drone video sequences. In recent times, the CNNs have gained a significant importance in the class of classification problems, but the cons being its computational intensity. High performance computing using Graphical Processing Units(GPUs) has become an important tool in solving the compute intensive problems. We have used GPUs to speedup the classification rate for our problem. We bring about a faster parallel implementation that achieves almost 13x on GPUs compared to its serial implementation.**

***Keywords*-Face classification, Convolutional neural networks, GPU**

## I. Introduction

With the increasing popularity of Graphical Processing Units(GPU's) in the recent past, they have been largely exploited for carrying out heavy parallel computations which resulted in massive speed up of execution time. GPUs are widely used in domains such as bio-informatics, data science, computer vision, and machine learning. Face classification is one such area in computer vision domain which has a wide range of applications such as human and computer interaction, surveillance systems, gender classification, facial feature extraction etc.

In general, face detectors work very well for the near frontal faces [1], but the problem arises when there is unconstrained face detection, i.e., factors such as pose changes and different light conditions can lead to visual disparity which in turn hinders the performance of the face classifier. So CNN based solution to this problem seems to be the most effective one [2]. The main focus of this paper is parallelization of a face classification algorithm from drone video sequences using deep convolution neural networks.
Face classification is the fundamental step in complex problems like face recognition process. Generally, faces are classified either from still frames or from a video depending upon the application.

The methods for face classification include - pixel based, edge based, parts based, haar wavelets and haar like features. Haar features have added a great importance to facial classification since faces can be classified even with occlusion [3]. Face classification can be doneine learning algorithms like Support Vector Machines, Bayesian Classifier, Artificial Neural Networks, Fisher linear discriminant, sparse network of Winnows, decision trees etc. have been used for this purpose [4].

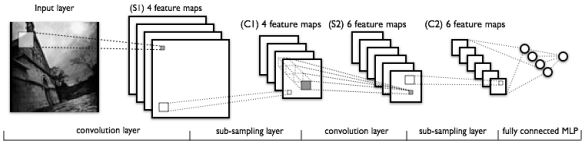


Figure 1. CNN Architecture [5]

CNNs consists of mainly three layers apart from the non-linearity and the loss layers, they are:

**Convolution layer:** Given an image as input, the main job of convolution layer is to perform the dot product between the weights and the locally connected regions, to compute the output for each neuron.

**Pooling layer :** Down sampling operation of the image along the spatial dimensions of the image ( i.e., width, height) is done by Pooling Layer.

**Fully connected layer (FC):** FC layer will compute the class scores. As is the case with Artificial Neural Networks and as the name implies, each neuron in the FC layer is connected to all the previous layer units.

## II. Literature Survey

Jieping Ye et. al.[6] proposed a novel LDA algorithm called as 2D LDA. In novel LDA algorithm the features say, nose, eyes, mouth are easily extracted, but the challenge was the class scatter and singularity. The 2D LDA algorithm overcomes the singularity problem, and the main difference between the LDA and 2D LDA is in the data representation. The hybrid approach of using LDA+2D-LDA achieved more classification and recognition accuracy.

In Imagenet classification [7], the convolutional deep neural network is trained using 1.2 million high resolution images for the classification. The training of the neural network is made faster by an efficient GPU implementation and non-saturating neurons.

AkinobuShimizu et.al [8], proposed a classification approach for finding the frontal and nearly frontal faces in strewed or cluttered images. The decomposition of the gradient direction gives a sophisticated discriminative ability than the image intensity and the gradient map. Furthermore, the detection performance was improved by combining the intensity and gradient direction.

Noel O Connor , Saman cooray [9], discussed in detail about detecting frontal faces; the technique they used was combining the feature extraction and statistical facial classification. The use of eye facial feature points was very helpful in normalizing the search space, which helped in reduction of image pixel location analysis. In turn this lead to a new approach for face classification.

The literature shows that the existing algorithms concentrate mainly on the frontal face classification and detection. The proposed algorithm and its parallel implementation performs very well with the frontal faces and also with visual variations such as, occlusions and partial faces. The remainder of the paper is organized as follows:

In Section 3, a detailed explanation of the proposed method i.e; how CNN's are used for face classification is given, d Section 4 talks about the serial and parallel implementations and the parallel strategies used. In Section 5 we present experimental results of the proposed optimized face classification approach. Finally we conclude with the future work.

## III. Proposed Method For Face Detection

We use CNN for face classification and customized the design accordingly for our problem.

### A. Outline of the algorithm

**step-1** *Intially, the neural network is trained using the back propagation algorithm.*
**step-2** *Once the CNN has been trained, the input images to be classified are read one after the other.*
**step-3** *We then use forward propagation method which consists of applying a series of convolution, max pooling and fully connected layers.*
**step-4** *Finally, we have the classification probabilities for each of the given image.*

## IV. Implementation

### A. Serial implementation

1) Initially, from the trained CNN, we populate array of structures corresponding to the number of convolution layers and maxpooling layers.

2) For each of the image to be classified we invoke the predict frame function for classification.

3) Each of the layer in CNN (i.e convolutionLayer, maxPoolingLayer etc.) is implemented as a separate function in our case. The PredictFrame function invokes the respective layers as when required.

The intermediary information between the layers is passed using structures(For eg: The output of convolutionlayer1 is passed as an input to the maxpoolinglayer1 in form of CONVOUT1 structure). Finally, we have convert function which converts the output of Fullyconnected layer to a float value.

After profiling the code with gprof profiler it was found that convolutionlayer function is the most time consuming step(93%, refer Fig2). Thus, it was decided to parallelize the convolution layer for improving the performance of the classifier.

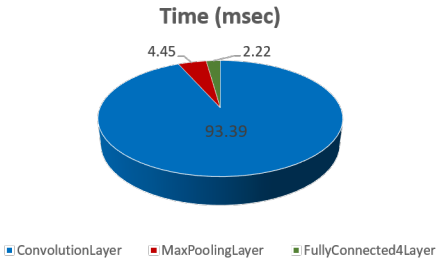


Figure 2. Profiling Results

### B. Parallel Implementation

Firstly, we copied all the CVLayers data, MaxpoolLayers data and FullyConnectedLayers data, biases, weights are copied from CPU host memory to GPU global memory. Convolution layer function is launched as a 2D kernel representing the two dimensions of the input image. The weights and biases are copied into the local memory. Loop unrolling has been done for the loop iterating over the kernels.

### C. GPU Optimizations :

**2D Kernel:** Since we have three convolution layers in our design, we launched the first convolution layer with (6,6)blocks with each block containing (16,16) threads. For the second convolution layer we launched (4,4) grid of blocks with each block containing (6,6) threads and finally for the third convolution layer we launched 1D grid with (12,12) threads.

**Local Memory:** In order to reduce the global memory accesses, we copied all the weights and biases from the global memory to the local memory. The array of biases are of sizes 16, 16, 16 and those of weights are 2352 , 6400 and 2304.

**Loop Unrolling:** Since among the 6 nested loops present in our implementation we could parallelize only the outermost loops in our algorithm, we did loop unrolling for the inner most loop which iterates over the number of channels, which are 16 in total.

## V. Experimental Set-up

### A. Hardware

- CPU : Intel Core i5-4670k
- Discreet GPU : Nvidia Tesla K40
- Embedded Platform : Jetson TK1
  - Tegra K1 SOC
  - Kepler GPU with 192 cores
  - 4-Plus-1 quad-core ARM Cortex-A15 CPU.
- 6 CH Remote Control Quadcopter (Figure-3)

Figure 3. 6CH Remote Control Quadcopter (2.4 GHz Frequency)

### B. Input

- Input proposals are collected from a drone video. The drone used for this task is a 6 CH remote control quad copter shown in Fig.3.
- Around 20hrs of drone video data is collected with varying height which is ranging from 6ft to 10ft. Input video consists both face and non face data.

## VI. Results and Analysis

The serial implementation of the algorithm implemented in C was taking around 64 msec. A series of optimizations were introduced in a progressive manner and at the end, the computational time was drastically brought down.

### A. Discreet GPU

#### 1) Serial Vs Parallel

Figure - 4, presents the timing comparison between CPU implementation and GPU implementation. Overall CPU implementation takes 64milliseconds per proposal for the classification. Whereas naive and optimized versions takes 54milliseconds and 5milliseconds respectively.

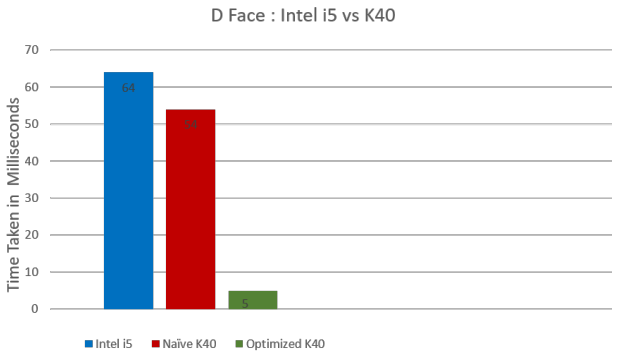


Figure 4. Timing Comparison between Serial and Parallel

#### 2) Speedup factor

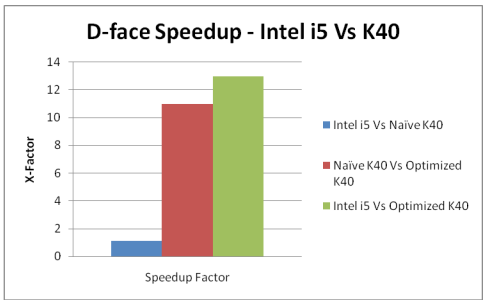


Figure 5. Speedup Factor - K40

Figure - 5, presents the speedup factor. Overall Optimized K40 implementation achieves around 13x speedup in comparison with CPU implementation.

#### 3) Optimizations

Figure - 6, presents the reduction of overall timings after application of various CUDA optimizations.

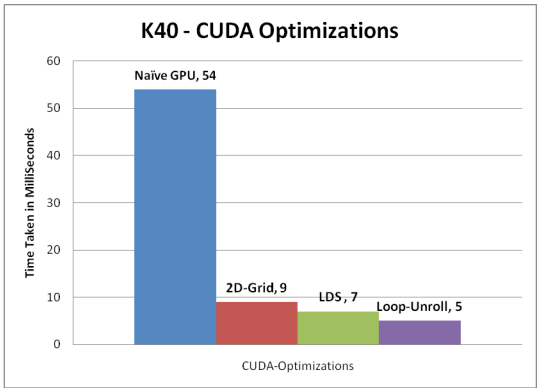


Figure 6. K40 - CUDA Optimizations

### B. Jetson TK1

#### 1) Serial Vs Parallel

Figure - 7, presents timings comparison between ARM CPU implementation and Tegra K1 GPU implementation. There is a significant improvement between CPU and Tegra K1 GPU.

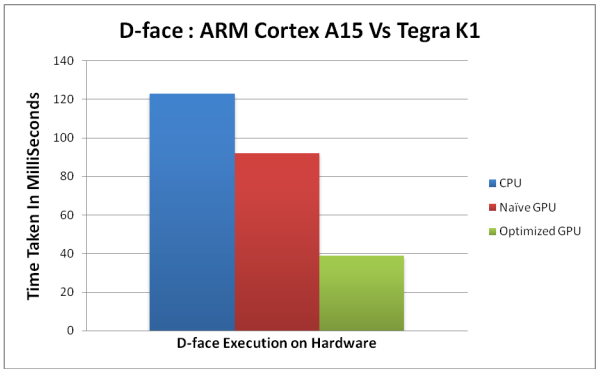


Figure 7. Timing Comparison

#### 2) Speedup factor

Figure - 8, presents the speedup achieved using Tegra K1 GPU. Around 3.2x speedup is achieved overall in comparison with ARM CPU implementation.

#### 3) Optimizations

Figure - 9, presents the impact of various CUDA-Optimizations on top of Naive GPU implementation.

## VII. Conclusions and Future Work

In this work, we present a GPU based implementation for face detection using convolution neural net-

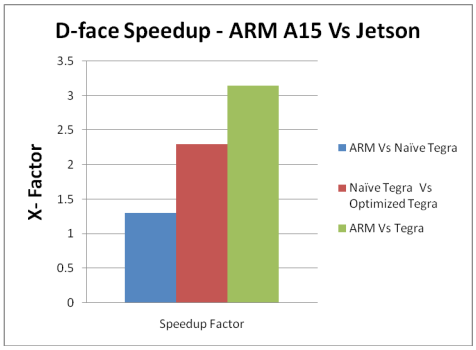


Figure 8. Speedup Factor - Jetson TK1

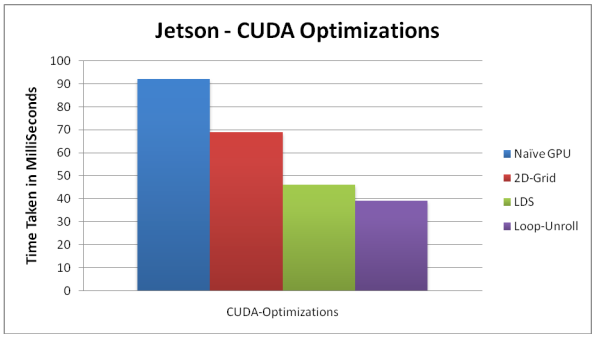


Figure 9. CUDA Optimizations

works(CNNs). We used Caffe and Torch framework for the CNN design and training. The time taken by the serial implementation is 64 msec, then we profiled the code and found that the convolution layer function was taking the maximun amount of execution time. A complete outlook of various optimizations used for high performance implementation were explained and most of the optimizations done in this work are based on the convolution layer function. We got a performance of almost 13x over the serial implementation using Tesla K40 GPU computing module.

In future, we can use vector data types, dynamic parallelism optimizations and various other video processing techniques to further improve the performance, and try to extend this work for real time applications such as drone face detection, video surveillance and various other security related applications.

## ACKNOWLEDGMENT

We would like to dedicate this work to the founder Chancellor of SSSIHL, Bhagawan Sri Sathya Sai Baba. We also would like to express our deep sense of gratitude to MulticoreWare, Inc for giving us all the technical support to make this project a reality and for supporting to take it further. This work was partially supported by a Nvidia grant under Professor partnership program.

## REFERENCES

[1] Viola, Paul, and Michael J. Jones. ”Robust real-time face detection.” International journal of computer vision 57.2 (2004): 137-154.

[2] Vidit Jain and Erik Learned-Miller. FDDB: A Benchmark for Face Detection in Unconstrained Settings

[3] Yang, Ming-Hsuan, David J. Kriegman, and Narendra Ahuja. ”Detecting faces in images: A survey.” Pattern Analysis and Machine Intelligence, IEEE Transactions on 24.1 (2002): 34-58.

[4] Jayech, Khlifia, and Mohamed Ali Mahjoub. ”Clustering and Bayesian network for image of faces classification.” arXiv preprint arXiv:1204.1679 (2012).

[5] http://deeplearning/tutorial/lenet.html

[6] Ye, Jieping, Ravi Janardan, and Qi Li. ”Two-dimensional linear discriminant analysis.” Advances in neural information processing systems. 2004.

[7] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. ”Imagenet classification with deep convolutional neural networks.” Advances in neural information processing systems. 2012.

[8] Lin-Lin Huanga; AkinobuShimizu a, Yoshihoro Hagiharab, Hidefumi Kobatakea Gradient feature extraction for classification-based face Detection Pattern Recognition 36 (2003) 2501 2511 Pattern Recognition Society. Published by Elsevier Ltd.

[9] Saman cooray, Noel O Connor Facial features and appearance-based classification for face detection in color images.