Real-time Face Detection in FHD Images Exploiting Both Embedded CPU and GPU

**Introduction**

Face detection has become very popular these days as its applications, along with face recognition, have become prevalent. Among them, many require real-time processing for the incoming video streams, such as intelligent surveillance systems. Recently, the widely used image resolution for surveillance camera has been increased to Full High Definition (FHD, 1920x1080), which obviously increases the computation time for face detection.

On the other hand, CPU-GPU heterogeneous systems have become a mainstream platform in both server and embedded domains with ever increasing demand for powerful accelerator. In this paper, we present parallelization techniques that exploit both data and task parallelism of LBP-based face detection algorithm on an embedded heterogeneous platform.

**LBP-based Face Detection**

- **Viola-Jonse Face Detection Framework**
  - Scanning
  - Pyramid of images

- **Cascade classifier**

- **Feature evaluation process**
  - For each single image, for x = 1, ystep : Image.width for y = 1, ystep : Image.height runClassifierCascadeSum()

- **LBP operation**

- **Pipeline of LBP based face detection**

**Proposed Methods**

- **LBP Parallelization**
  - Highly memory-intensive

- **Pipeline of execution**

- **Reducing global memory loads**
  - Packing smaller type of data into larger type of data
  - Reducing indirect memory loads by merging multiple tables into one big table

- **Multi-phase Scanning Kernel**
  - Skip evaluating the next patch when the current patch fails at the very first stage

- **Data-parallel task partitioning**
  - Alleviates performance bottleneck to satisfy real-time constraint exploiting both task/data-parallelism

**Experimental Results**

- **NVIDIA Jetson TK1**
  - Released in 2014
  - First embedded GPU that supports CUDA
  - Same architecture with server GPU (Kepler)

- **Parameters**

- **Comparison with OpenCV**

- **Multi-phase kernel launching**

**Face Detection Results**

**Real-time achievement from task partitioning**