**Name Surname: Emre Karaman**

**Date: 17 November 2015**

**Speaker:** Emre Karaman / IUI Lab Koc University

**Title:** “Clustering appearance and shape by learning jigsaws”, IUI Lab Reading Group.

**Problem Definition:** Discovering both the shape and the appearance of object parts in an image without supervision.

**Main Problems:**

* There is no way to choose the shape and size of a patch in existing patch based models.
* A predefined set of patch sizes and shapes are used in existing patch based models.
* Histogram bases models throws away spatial information, template-based models cope poorly with articulation.

**Related Work & Main Differences:**

* The closest work is the epitome model of Jojic et al. Epitomes are learned using a set of fixed shaped patches over a small range of sizes. In contrast, in the jigsaw model, the inference process chooses appropriately shaped and sized pieces from the training images when learning the jigsaw.
* This work is also related with the seminal work of Freeman et al. that proposed a general machinery for inferring underlying scenes from images. They define Markov Random Field as Jigsaw model however they use a set of fixed size image patches.

**Proposed Solution:**

* They proposed ntopng an open-source web-based monitoring console.
* The main goal of ntopng is the ability to provide a real-time view of network traffic flowing in large networks while providing dynamic analytics able to show key performance indicators and bottleneck root cause analysis.

**Conclusions:**

* ntopng is fully scriptable by means of an embedded Lua JIT interpreter, guaranteeing both flexibility and performance.
* Validations tests have demonstrated that ntopng can effectively monitor 10 Gbit traffic on commodity hardware due to its efficient processing framework.

**My Questions & Deductions:**

* How they serve and export monitoring data?
  + Monitoring data is represented using HTML 5 served by the embedded web server, and it can be exported to external monitoring applications by means of a REST API.
* Why do they achieve to detect and characterize the most popular network protocols?
  + This is achieved by developing a specific framework instead of including this logic within ntopng. This avoids the need of modifying ntopng when new protocols are added to the framework.