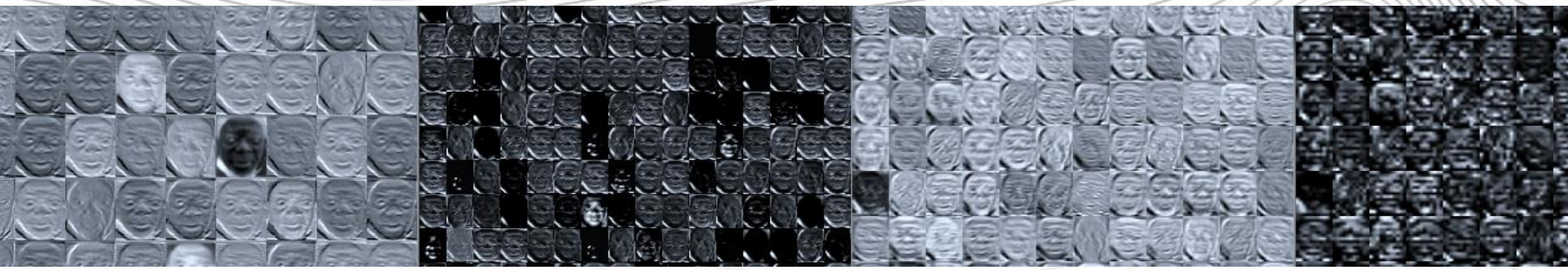
Implementing Deep Learning for Video Analytics on Tegra X1

research@hertasecurity.com



Index

- **Who we are, what we do**
- **Video analytics pipeline**
  - Video decoding
  - Facial detection and preprocessing
  - DNN: learning recipes + deployment
- **Multi-DNN performance on TX1**
  - Latency per architecture / layer

Who we are

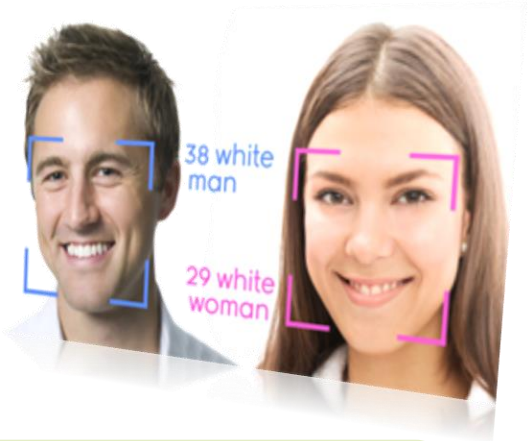
- **Facial recognition** company
- **GPU**-powered solutions for **security** and **marketing**
- Offices in Barcelona, Madrid, London, Los Angeles

What we do

Real-time, unconstrained facial analysis on HD streams.

Video surveillance
Facial recognition in *crowds,*
live and forensic (up to 400 fps).

Facial video analytics
Gender, age, ethnicity,
facial expression, counting...

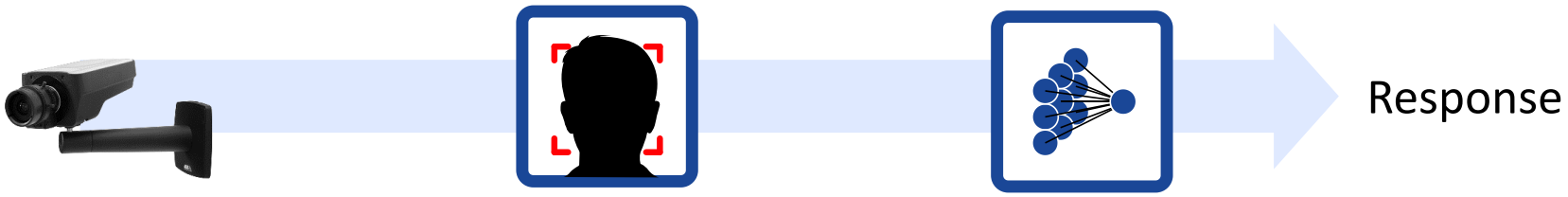


#GTC16

YOU Bar – We know what you're drinking

HERTA booth 727 – Live demos (discrete + embedded)

Video analytics pipeline

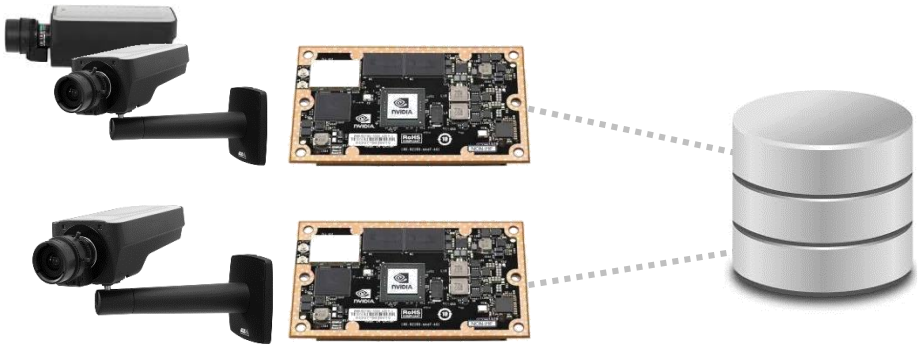


Video decoding

Detection and preprocessing

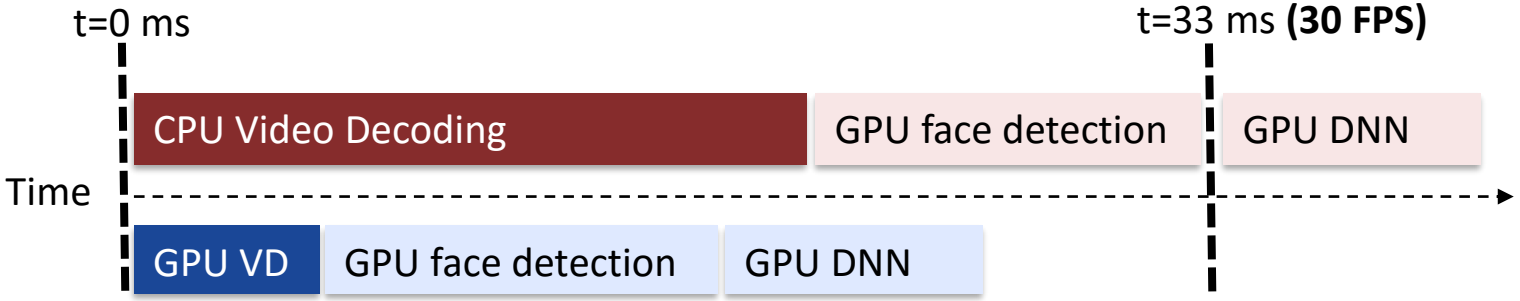
DNN inference

- Fully local processing
- Enables targeted response (e.g., digital signage)
- Send aggregated statistics to remote DB



Video decoding

- **Tegra X1** features on-die hardware video decoder



- Stages of video acquisition:
  - RTSP parsing
  - Video demuxing
  - Video decoding

gstreamer

H.265 HEVC H.264 MPEG-4/AVC

Face detection

- **DNN-based face detection** (sliding-window):
  Still slow, even on discrete GPUs!

Soft cascade of CNN2 (2015) [1]	640x480	**15 FPS**	Tesla K20
HOG + DNN (2015) [2]	640x480	**0.7 FPS**	GTX 760

- Not yet feasible for 4K on power-constrained TX1.
  Instead, dedicated cascade using custom CUDA kernels

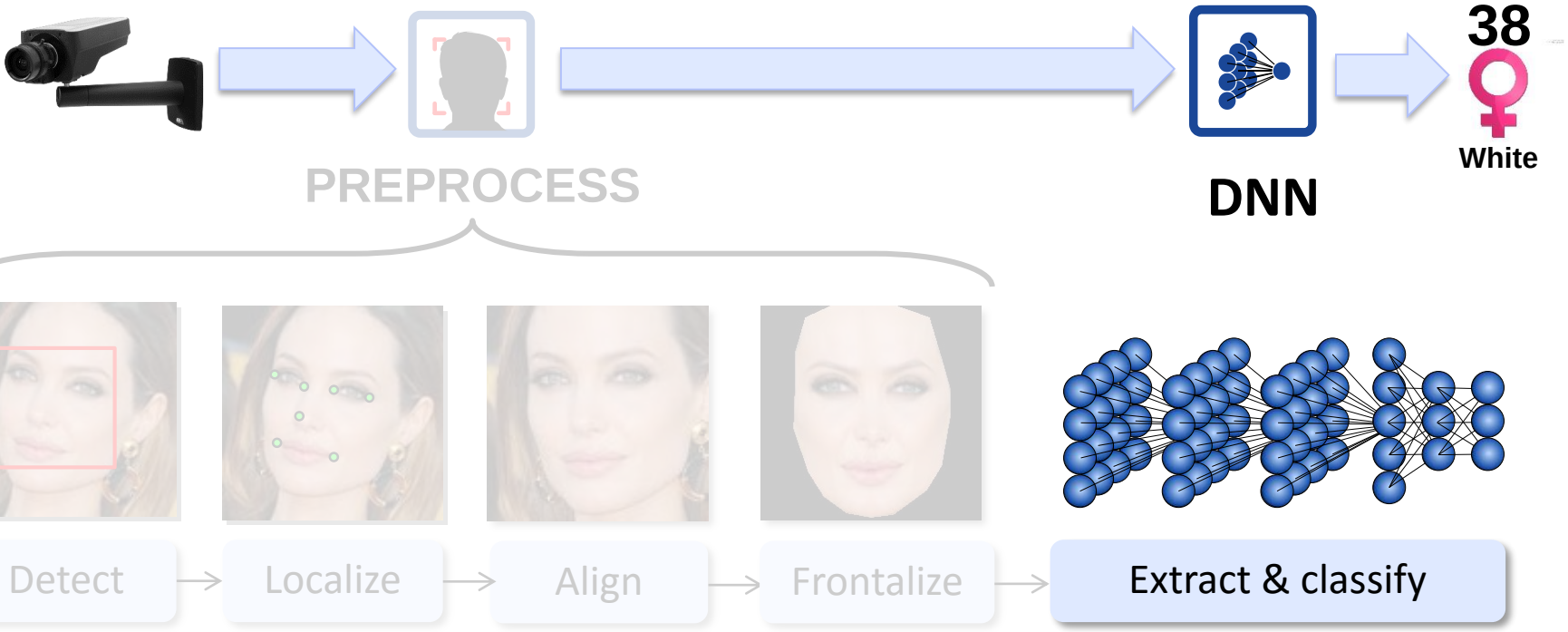
Our boosting approach	3840x2160	up to **30 FPS**	Tegra X1
Our boosting approach	3840x2160	up to **150 FPS**	Quadro K2200

[1] Angelova et al, *Real-time pedestrian detection with deep cascades.* BMVC'15

[2] Luo et al, *Switchable deep network for pedestrian detection*. CVPR'15

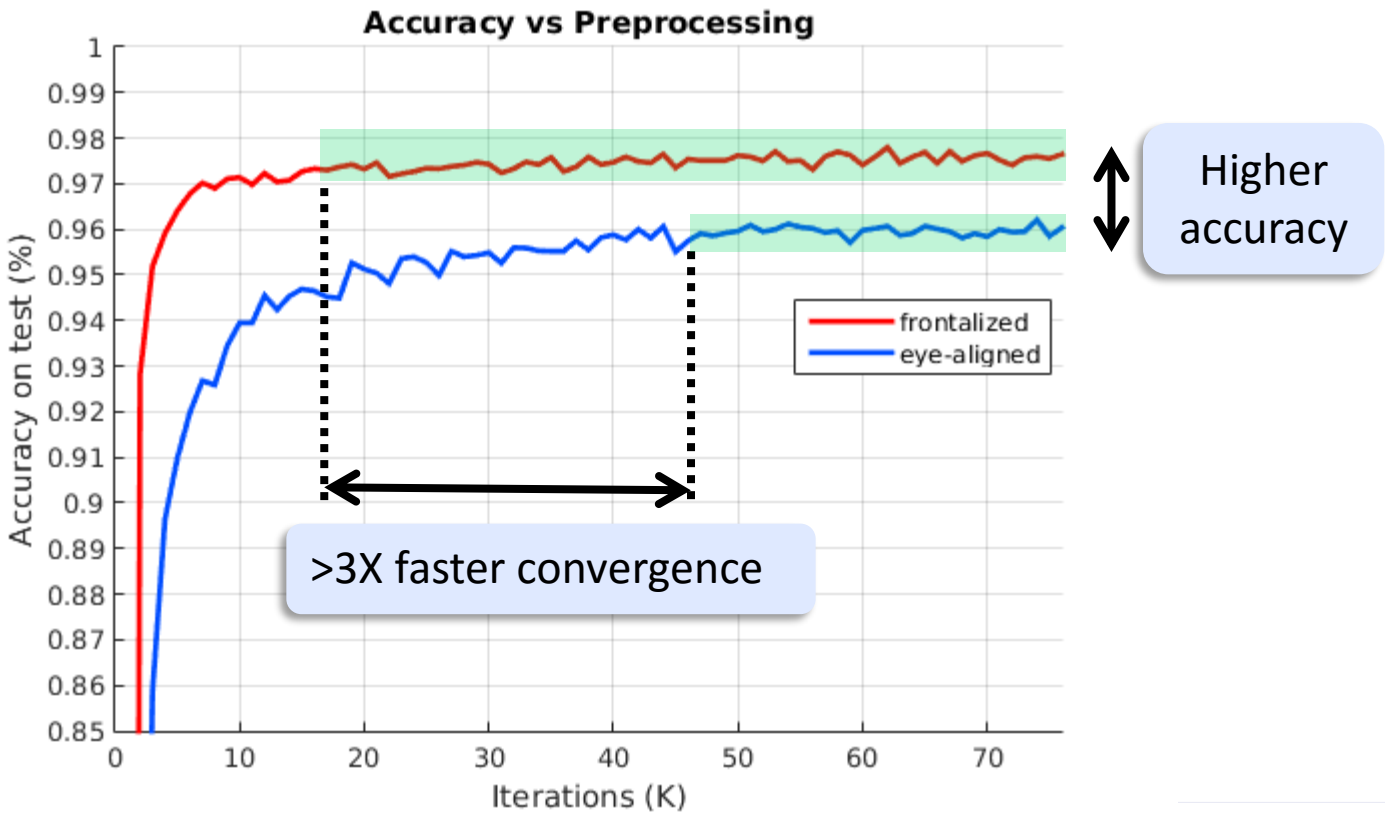
Preprocessing (I)

Preprocessing: Independent tasks for detection, normalization, alignment



Effects of custom preprocessing

- Feeding the network: **eye-aligned** vs. **frontalized** faces



DNN deployment

herta security

- **Deployment**: very fast, models are already trained.

- Basically: sequence of **matrix multiplications**, *(GEMV / GEMM)*
  And a bunch of extremely fast **CUDA kernels**. *(pooling, LRN, ReLU…)*
  Not a big deal for TK1 / TX1, right?

- But how deep is ***deep***?

AlexNet – **8** layers

VGG – **19** layers

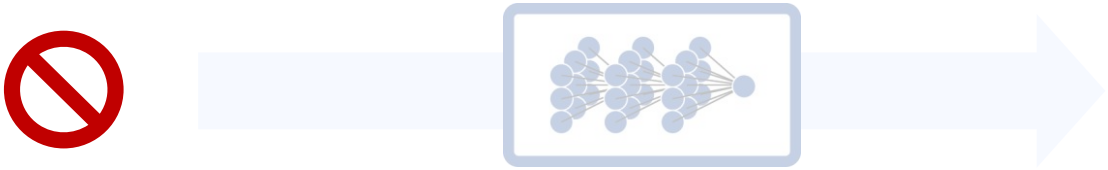
Inception – **22-75** layers (+most with parallel subnetworks)

ResNet – **152** layers

…plus 4K video processing, multi-net inference… All **real-time.**

DNN deployment (II)

- Alternatives: busy DNN, or independent custom tasks
- A good **compromise**: efficient preprocessing, shallower guided DNNs.



Busy DNN:
Localization + classification

Detection
and preprocessing

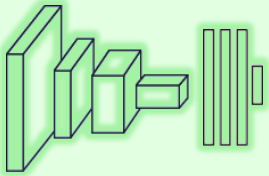
Guided DNN
(~8-15 layers)

DNN recipes (I)

herta security

DATA

ARCHITECTURE



$$obj(\Theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

LOSS

- Softmax
- Euclidean
- Contrastive
- Info gain

REGULARIZERS

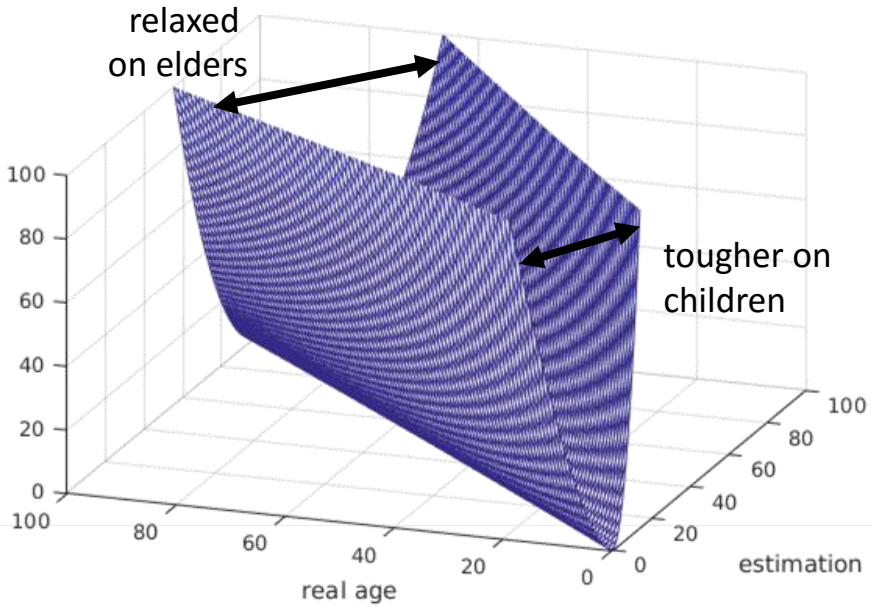
- Weight decay
- Dropout
- Batch normalization
- Stochastic pooling
- MaxOut
- Multi-net averaging

DNN recipes (II)

herta security

LOSSES

- **SoftMax:** great for binary / multiclass classification
- **Information gain:** for unbalanced datasets
- **Contrastive + Information gain:** for Siamese nets
- **“Euclidean”:** for regression, much better after tuning



DNN recipes (III)

herta security

REGULARIZERS

- **Dropout:** *always* great to generalize small datasets.
- **Batch normalize:** for us, better after PReLU + dropout!
- **Stochastic pooling:** mixed results.
- **MaxOut**: basically, partial multi-net averaging.
- **Multi-net averaging:** good to generalize, but expensive.

ARCHITECTURE

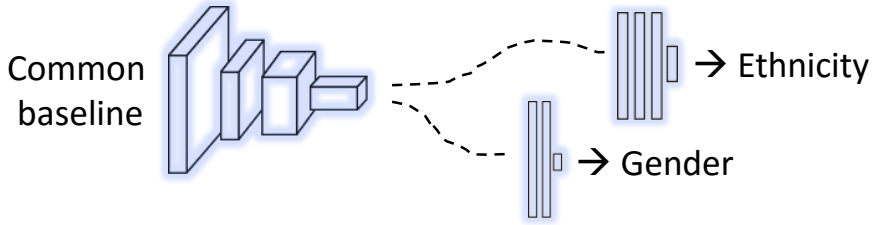
- **Weight initialization:** Xavier / Fan-out vs Average / He
- **Activation functions:** ReLU / PReLU / ELU
  ("*ELU need deeper nets to shine*")

DNN recipes (IV)

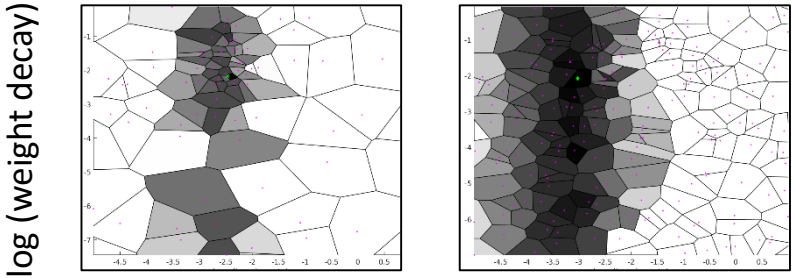
herta security

ARCHITECTURE

- **Convolutional autoencoders:** “just” for initialization.
- **Common arch + task-oriented heads:** dedicated is better.



- **Hyperparam cross-validation:** random walk / Monte-Carlo.



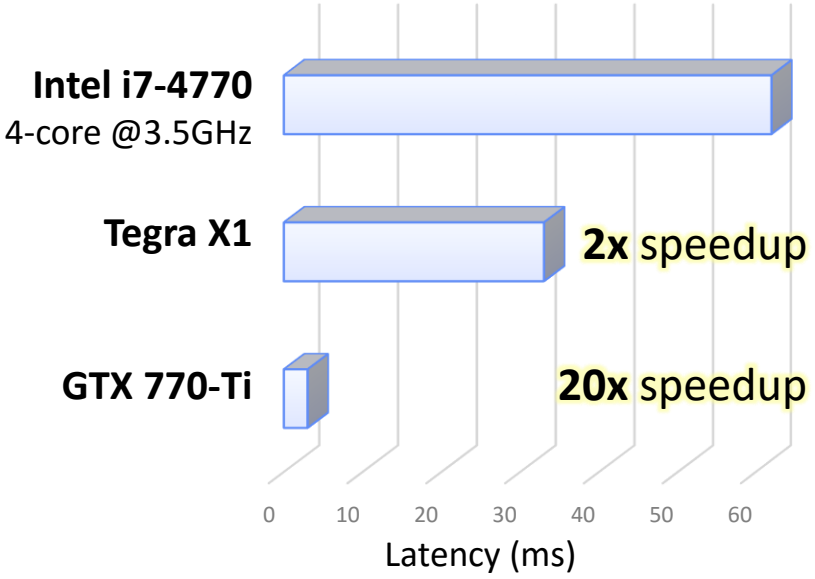
Implementation summary

- Demographic estimation on TX1:
  - H.264/HVEC **video decoding** with NVIDIA's GStreamer plugins
  - Custom CUDA kernels for **detection + preprocessing**
  - cuDNN v4 **DNN layers:** Convolution, MaxPool, SoftMax
  - Custom **DNN features + layers** (not yet in cuDNN): He, PReLU, MaxOut, StdPool…
  - Custom **DNN inference engine** on TX1.
  - OpenGL for display

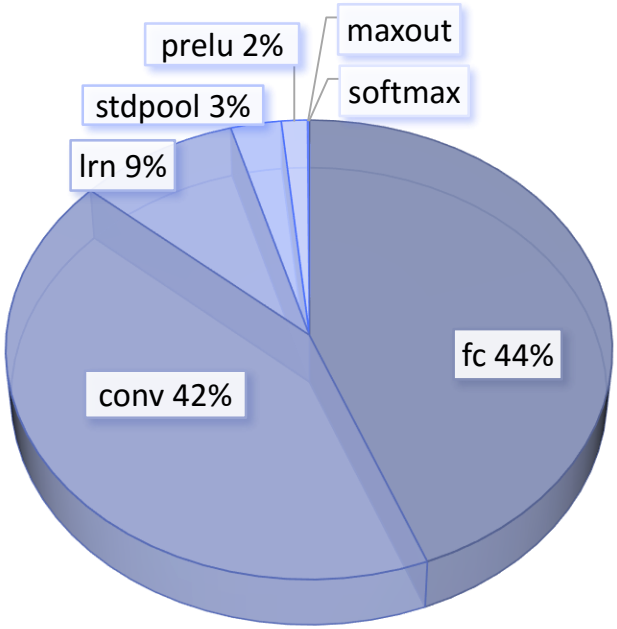
Multi-DNN performance on TX1

Simultaneous estimation of ***Gender + Ethnicity + Age*** in HD streams.

Latency per architecture



Latency per layer



***Next**: bottleneck convs, FP16, better kernels*

We acknowledge NVIDIA for their support,
through the UPC/BSC GPU Center of Excellence.

herta security

Nacho Navarro
(1958 – 2016)

Expert in OS, enjoyed the internals of UNIX, Mach, Chorus, Exokernels
Associate Professor at DAC (UPC)
Leader of the group Accelerators for HPC (BSC)
Leader of the BSC/UPC NVIDIA GPU CoE

Pioneered research in GPU systems
Automatic CPU + GPU memory management (adopted by industry in CUDA 4/6)
Transparent multi-GPU programming
Architectural support for making GPUs first-class OS citizens

Established and led the NVIDIA BSC GPU Center of Excellence (since 2011)
Teaching GPU programming, introducing CUDA to undergraduate courses @ UPC
PUMPs summer school with Wen-mei Hwu and David Kirk (since 2009)
PRACE introduction to CUDA programming (since 2012)
Managing GPU development machines supporting 150+ people