# Performance estimation of deep learning methods for change detection on satellite images with a low-power GPU

**A. Di Pilato**[1,2], *N. Taggio*[3], *M. Iacobellis*[3]

[1]University of Bari
[2]National Institute for Nuclear Physics (INFN)
[3]Planetek Italia

22-26 March 2021

# Research overview in remote sensing

- **Scenario**: unprecedented growth of Earth Observation (EO) missions
  - Advanced sensor technologies
  - Large number of satellites launched to orbit

- **Two** main drawback effects:
  1. Huge amounts of heterogeneous data acquired daily from satellites
     →***Big Data problem***
  2. Limited transmission bandwidth
     →considerable **delay** between acquisition onboard and download on ground

# Research overview in remote sensing

- **Solutions**: exploit new hardware/software technologies and strategies
  - **Heterogeneous hardware** architectures (CPU + GPU)
  - **Deep Learning** algorithms

- **Objectives**
  1. ***Selection*** of data that carry particularly interesting information in an *automated way* (onboard pre-processing phase)

     **Deep Learning approach**
  2. ***Accelerate*** data management

     **Heterogeneous computing approach**

# EO Change Detection

*"**Change detection** is the process of identifying differences in the state of an object or phenomenon by observing it at different times."*[1]

We can distinguish several classes of changes:

- ❑ **Urban changes**: new houses and infrastructures, roads, bridges, etc. (human intervention)
- ❑ **Natural changes**: vegetation growth, glacier melting, etc. (climate or seasonal)
- ❑ **Anomalies**: floods, wildfires, earthquake disasters, etc. (occasional events)

Usual "fake" changes:

- ✗ **Earth's surface brightness conditions**
- ✗ **Clouds**

[1] Ashbindu Singh. "Review Article Digital change detection techniques using remotely-sensed data". In: *International Journal of Remote Sensing* 10.6 (1989) doi:10.1080/01431168908903939

# EO Change Detection

**Strategy: compare two satellite images of the same scene but acquired at different times**

❖ **Case 1**: a **very small** number of changes are detected

➢ **Action**: discard the new image or put it in a "low-priority" queue

✓ **Motivation**: the new image does not provide additional information with respect to the previously-acquired one

❖ **Case 2**: a **considerable amount** of changes are detected

➢ **Action**: transmit the new image to the ground station or put it in a "high-priority" queue

✓ **Motivation**: the new image might contain important information that should be submitted to a deeper analysis on ground

# Objective

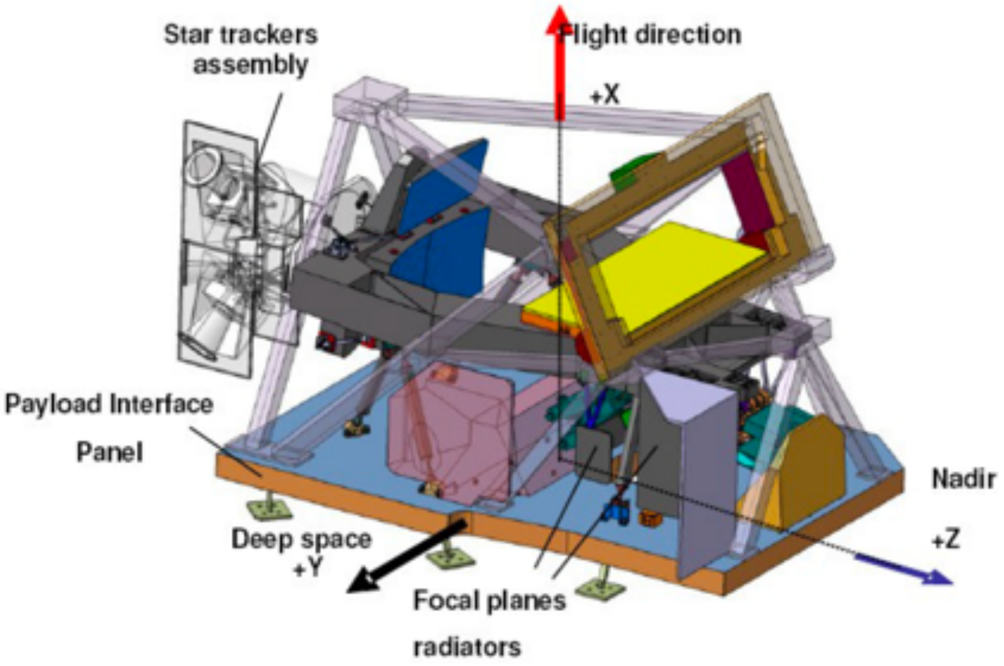
1. **Test** the performance of two DL methods
   - ***Segmentation*** **(on ground)**: classifying a scene pixel by pixel and produce a binary change map
   - ***Classification*** **(onboard)**: classifying an entire scene as including changes or not

2. **Evaluate** the performance of a low-power consumption GPU
   - Benchmark test
   - Comparison with typical data center GPUs

Data selected: ***Sentinel-2*** (L1C products – no atmospheric corrections)

# The Sentinel-2 mission

- The **Copernicus*** **Sentinel-2** mission (developed by **ESA**) consists of a constellation of two polar-orbiting satellites flying in the same orbit but phased at 180° to each other at a mean altitude of 786 km
  - monitors variability in land surface conditions
  - features high revisit time (5 days at equator)
  - each satellite carries a ***Multi-Spectral Instrument*** (MSI) that measures Earth's reflected radiance in **13 bands** (four at 10 m, six at 20 m and three at 60 m of spatial resolution)




*Information about the Copernicus programme available here

# Onera Satellite Change Detection dataset

- **24 Sentinel-2 image pairs** covering regions of approximately 600 × 600 pixels at 10m resolution
  - Only 14 have ground truth and can be used for training
- **13-bands** and **RGB** images available
  - Bands at 20 m and 60 m resolution upsampled to equivalent 10 m resolution
- No (or very few) clouds present
- Pixel-wise ground truth generated (***binary change map***)
- Focused on **urban** areas
  - urban changes labeled
  - natural changes ignored

Sample city: **Pisa** (Italy)

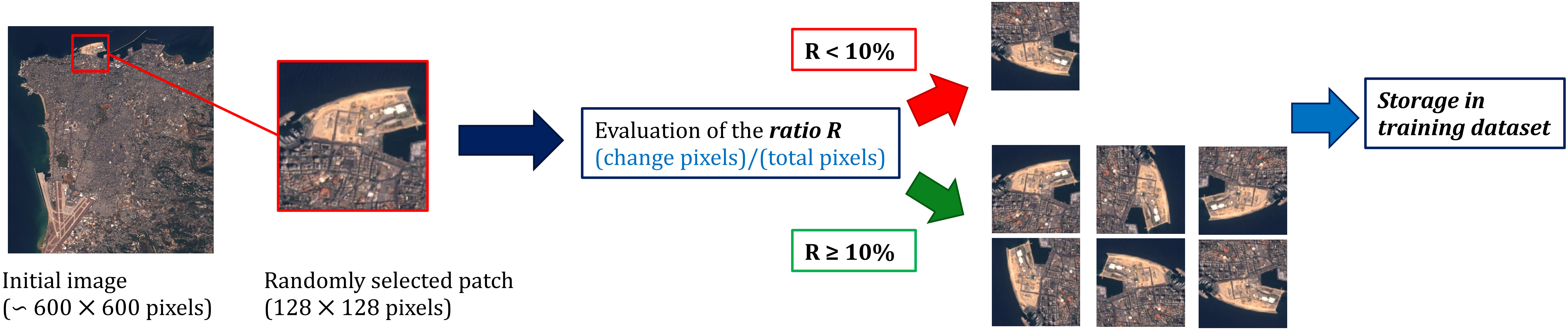
Pre-changes: **04/07/2015**

Post-changes: **11/02/2018**

R.C. Daudt et al. "Urban Change Detection for Multispectral Earth Observation Using Convolutional Neural Networks". In: *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2018.

# Training strategy details - 1

- **Small** dataset but **large** images → pre-processing phase: ***creation of patches of size 128 × 128 pixels***
  - ***Randomly*** selected across the initial images (not out of scene borders)
  - One or more ***random*** transformations applied (rotations by 90°, vertical or horizontal flip) based on the amount of «change» pixels captured
  - Each selection and transformation is applied to images pre and post changes that must be compared, and to the corresponding ground truth
- Introducing randomness in the dataset creation process reduces the ***pixel correlation between patches of the dataset extracted from the same initial image***



# Training strategy details - 2

- **Loss function (to be minimized by the Neural Network):** ***weighted binary cross-entropy***

Number of **change** pixels ***only covering a small percentage*** **(~1%)** of the total number of pixels in each image pair

$$\mathcal{L}_{bce} = -\sum w_i t_i \log s_i$$

where $i$ = 0,1 (binary classification), $w_i$ is the weight assigned to class $i$, $t_i$ = ground truth with respect to class $i$, and $s_i$ = output with respect to class $i$, ($t_i$ and $s_i$ are one-hot vectors)

- **K-fold cross validation method → *useful with small/poor datasets***

Dataset split into ***k=5 parts***. Reserving one part for validation, the model is trained on a partition of the dataset. Training of 5 different models (***k=5 possible partitions***) for each architecture to ***evaluate overall scores*** *of the model architecture, independently of the chosen partition*.

# Scores of interest

| Actual | | Predicted | |
|---|---|---|---|
| | | **Positive** | **Negative** |
| | **Positive** | True Positive $T_P$ | False Negative $F_N$ |
| | **Negative** | False Positive $F_P$ | True Negative $T_N$ |

**Scores of interest (obtained at the validation step)**

$$Precision = \frac{T_P}{T_P + F_P}$$

$$\boxed{Recall = \frac{T_P}{T_P + F_N}}$$

$$F1 = 2\times \frac{Precision \times Recall}{Precision + Recall}$$

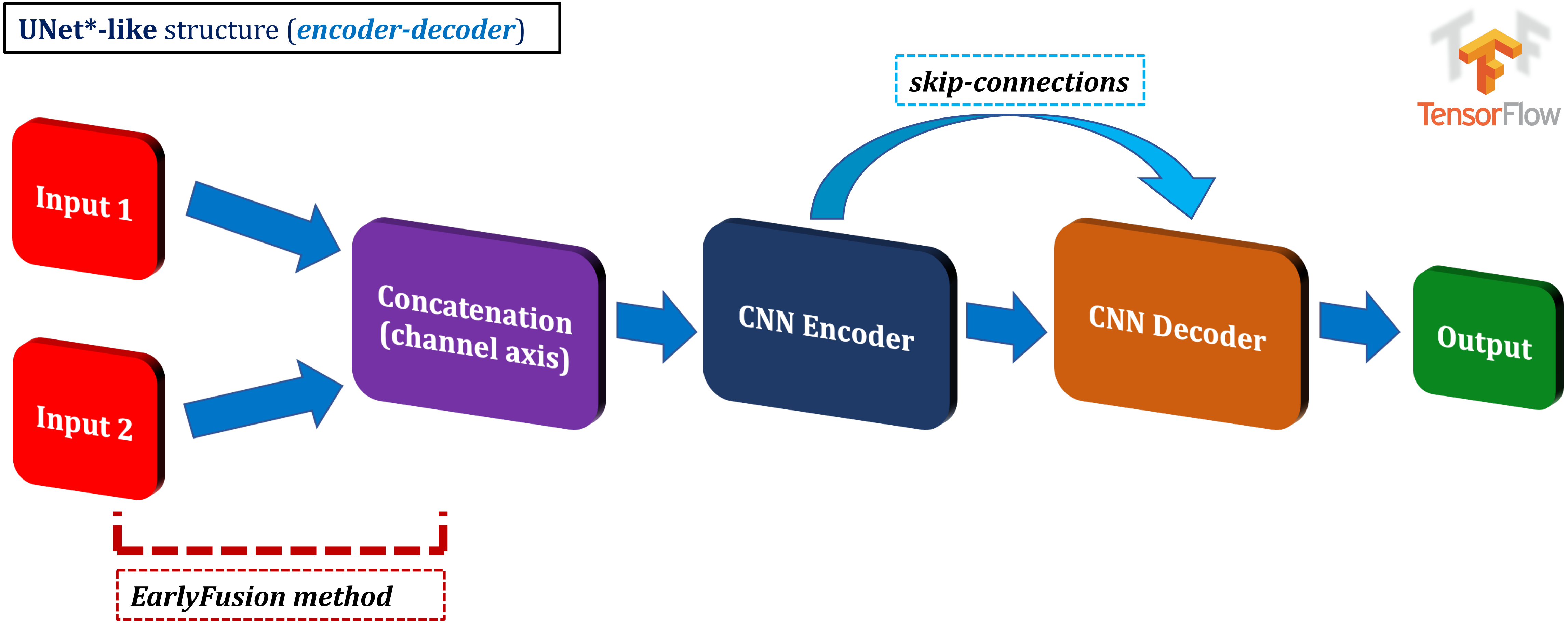
$$Balanced\ Accuracy = \frac{1}{2}\times\left(\frac{T_P}{T_P + F_N} + \frac{T_N}{T_N + F_P}\right)$$

***Recall*** (*sensitivity*) was chosen to be maximized to better describe the model performance for change detection

The general idea is that it's better to have more $F_P$ than losing real changes. ***F1-score*** combines both ***precision*** and ***recall*** to provide an extremely useful value that defines the overall model performance.

# Model architecture - segmentation

**UNet*-like** structure (***encoder-decoder***)

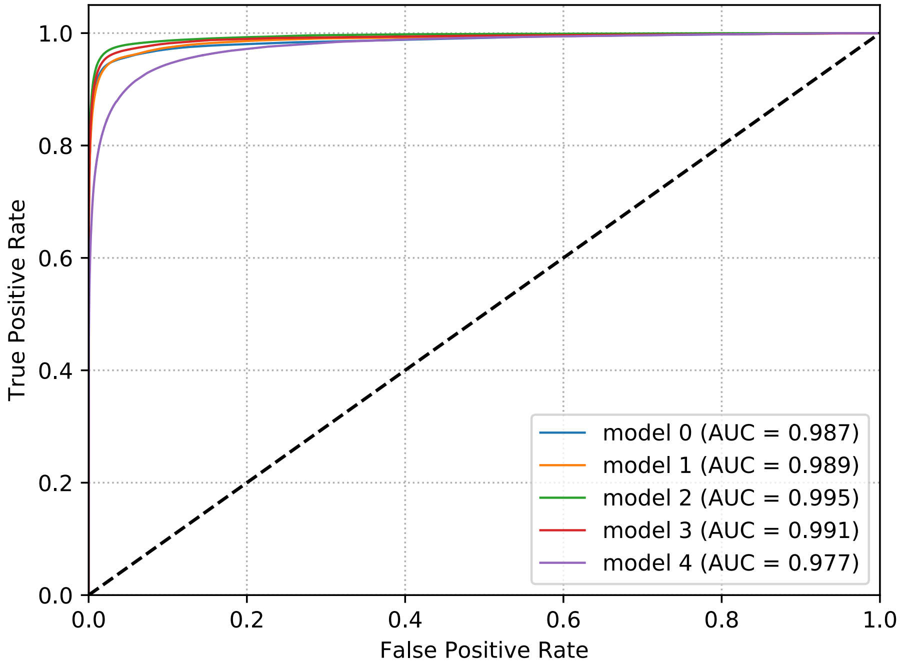



*O. Ronneberger et al. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR abs/1505.04597*, 2015.

# Threshold selection - segmentation

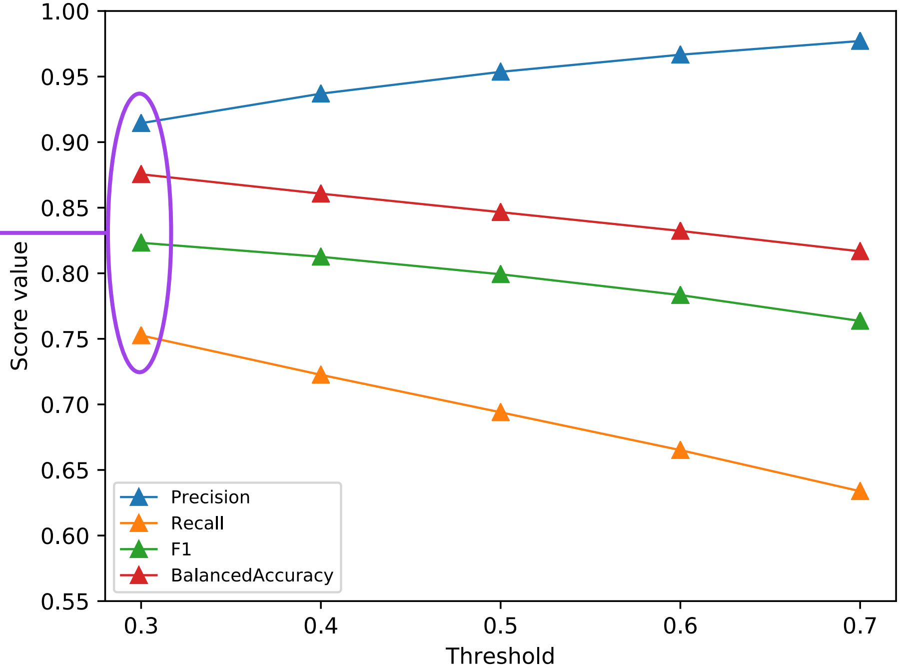
**Objective:** define the **threshold** above which an **output pixel** in the change map is defined as **"change"**

***ROC curve***




Maximizes ***recall*** and ***F1***, while keeping a high value of ***precision***

***Scores vs threshold***

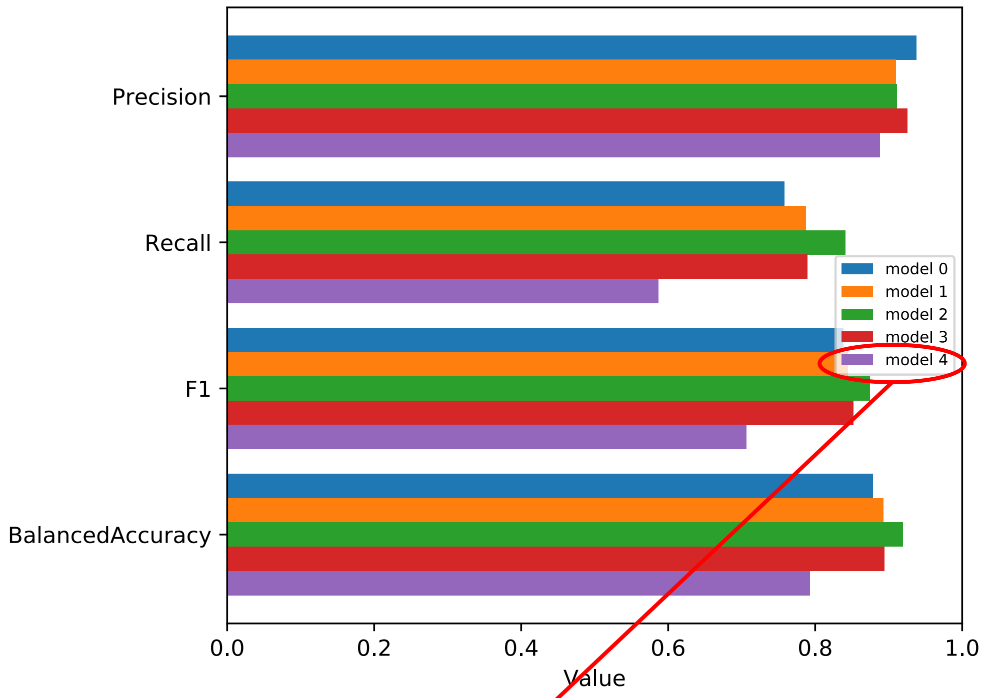



**AUC = Area Under Curve**

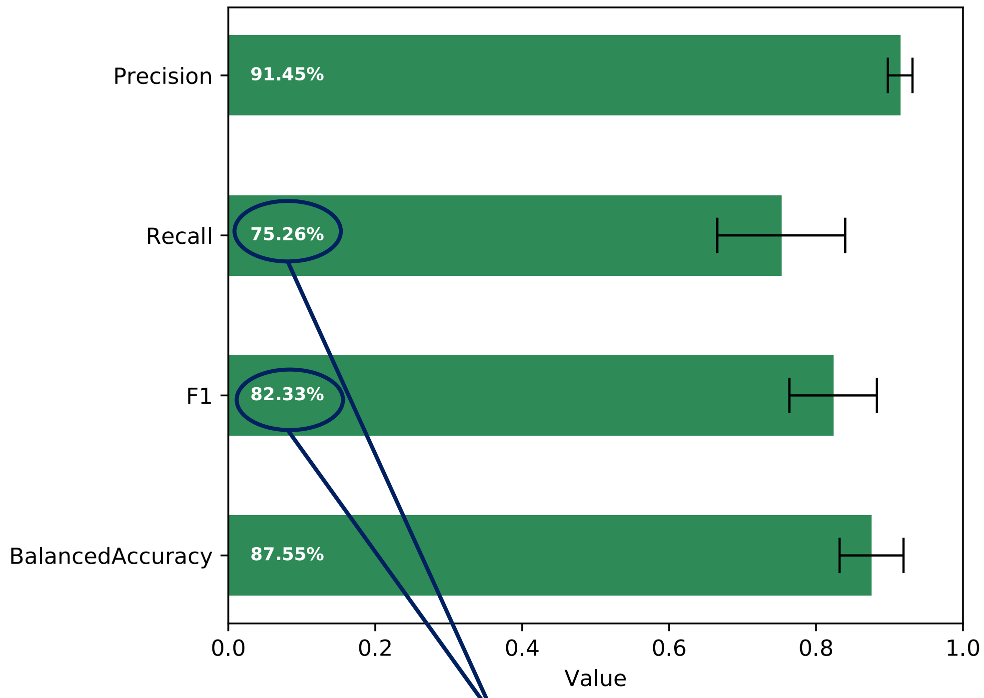
A good binary classifier should have **AUC > 0.5**. Higher AUC values correspond to better performance.

# Validation scores - segmentation

***K-fold validation scores***




***Average validation scores***

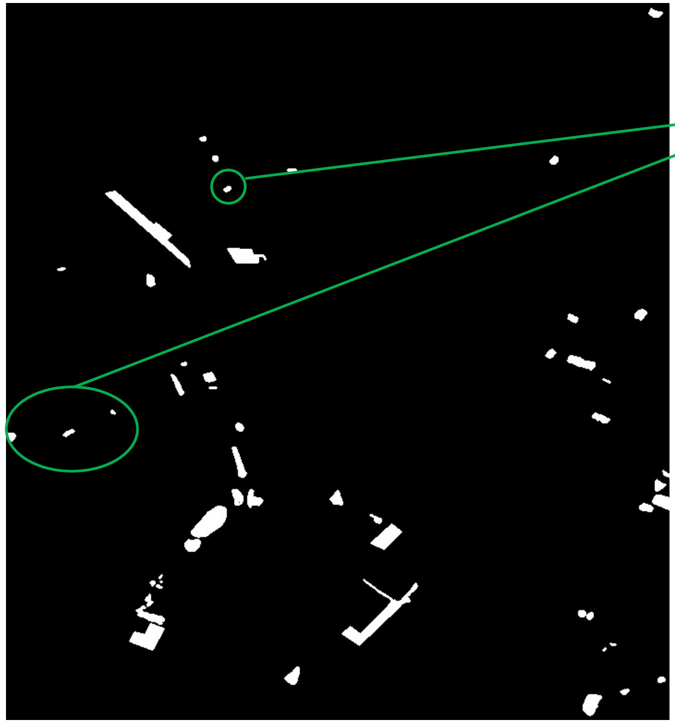



Anomalous performance due to ***possible training issue*** (i.e. loss function stuck in a local minimum)

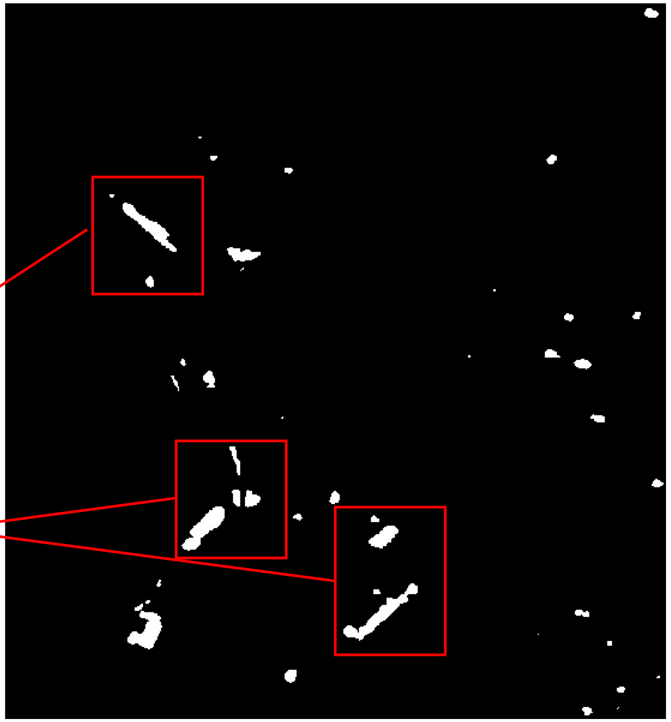
Values ***comparable*** with the state of the art.
Error bars are calculated as the ***standard deviation*** across the five models trained with the K-fold validation process.

# Sample change map - Pisa

***Ground truth***



***Predicted output***



Very small changes not detected

Large structures detected and outlined at correct locations

Most of the changes **are detected** at correct locations. Structures are not perfectly outlined (it would be a "smoking gun" of consistent ***overtraining***) but can be ***refined with post-processing algorithms***.

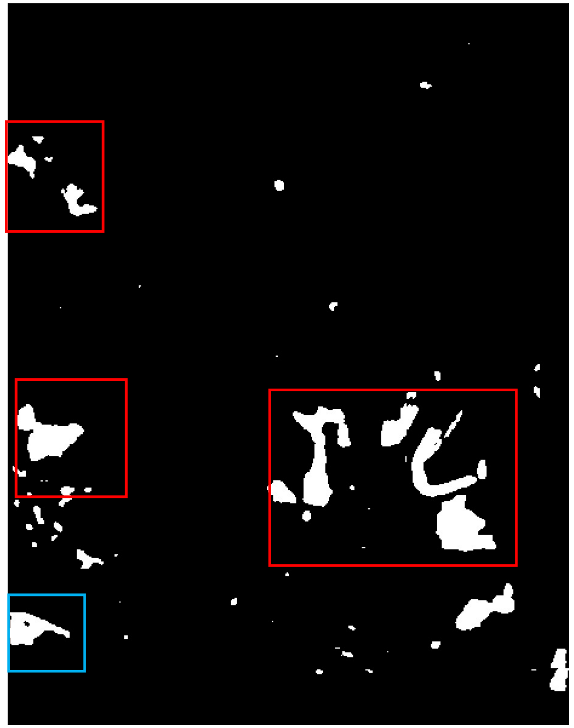
# Sample change map – Hong Kong

***27/09/2016***

***23/03/2018***

***Ground truth***

***Predicted output***



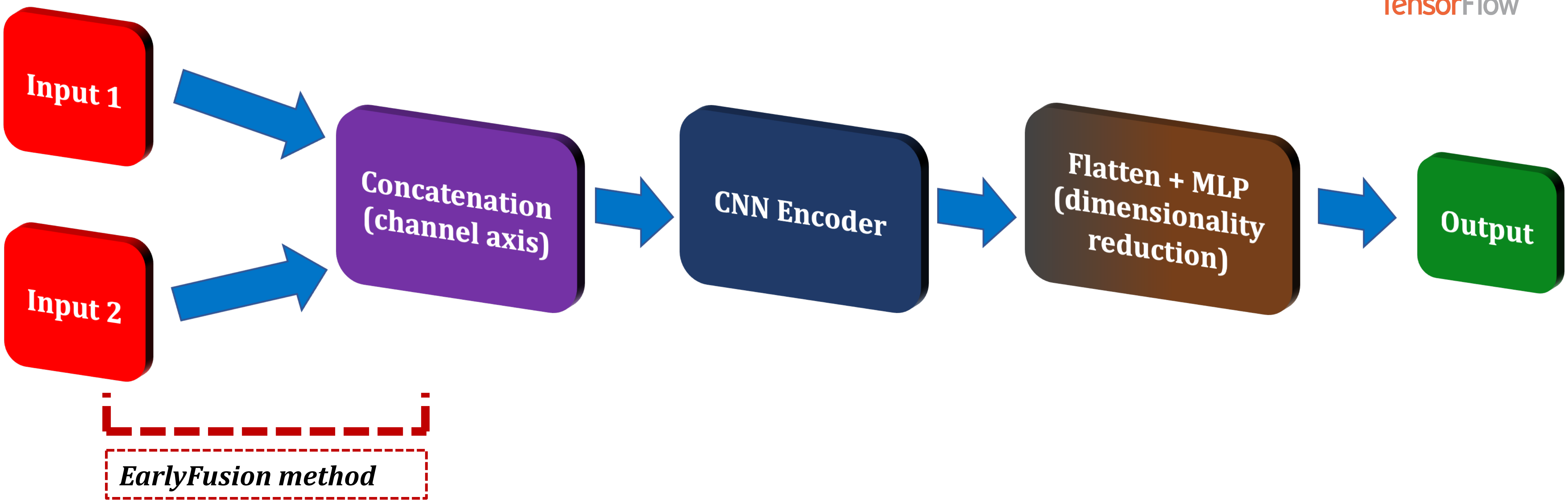
*False negatives*

*True positives*

*False positives* (drawback of the 0.3 threshold)

# Model architecture - classification

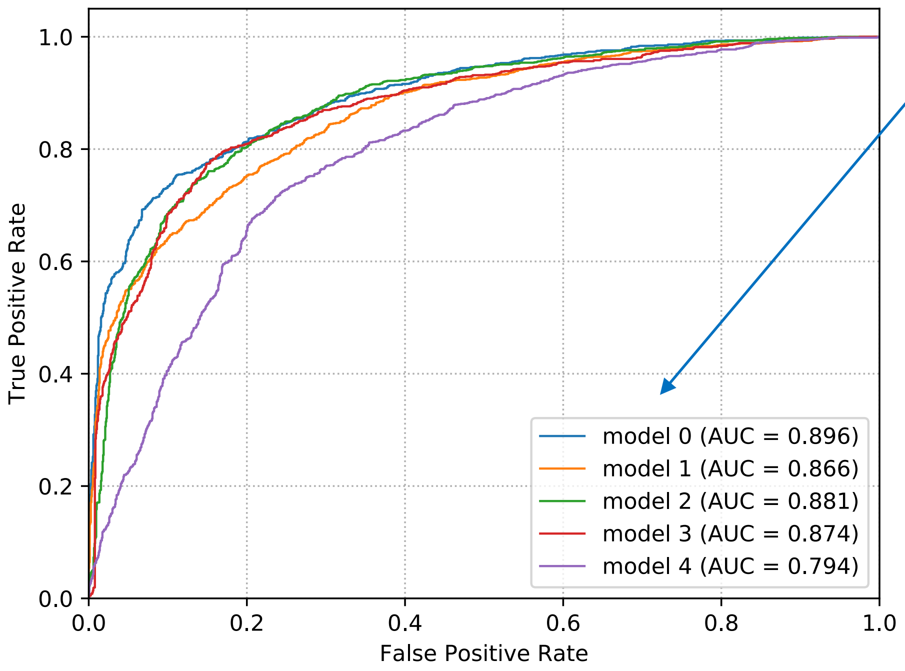
Traditional CNN structure (***encoder+MLP***)



# Threshold selection - classification

**Objective:** define the **threshold** above which the **output of the CNN** is considered as **"change"**

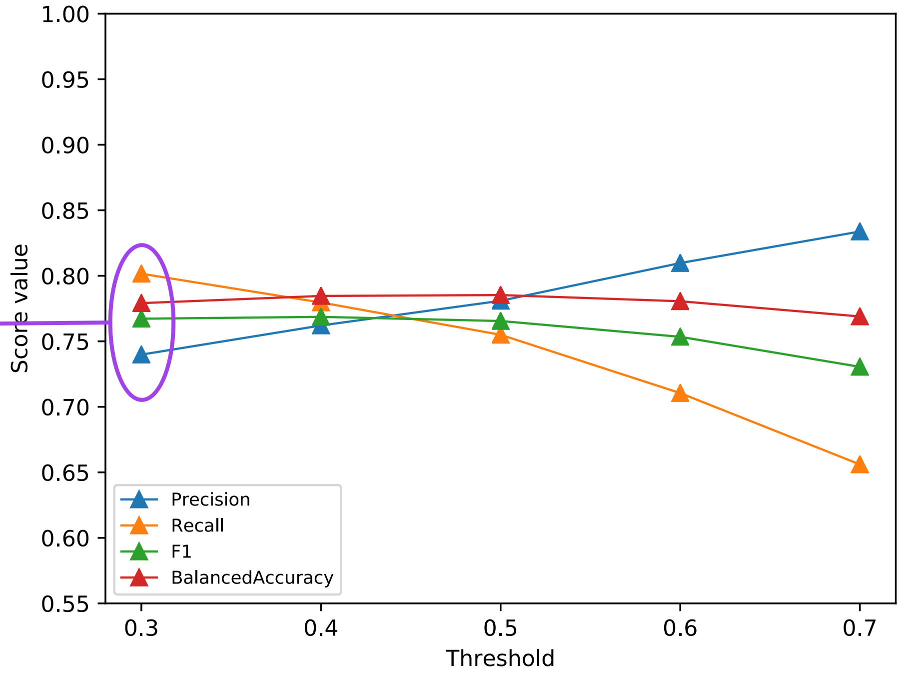
***ROC curve***




Lower AUC scores due to the *particular ground truth definition*

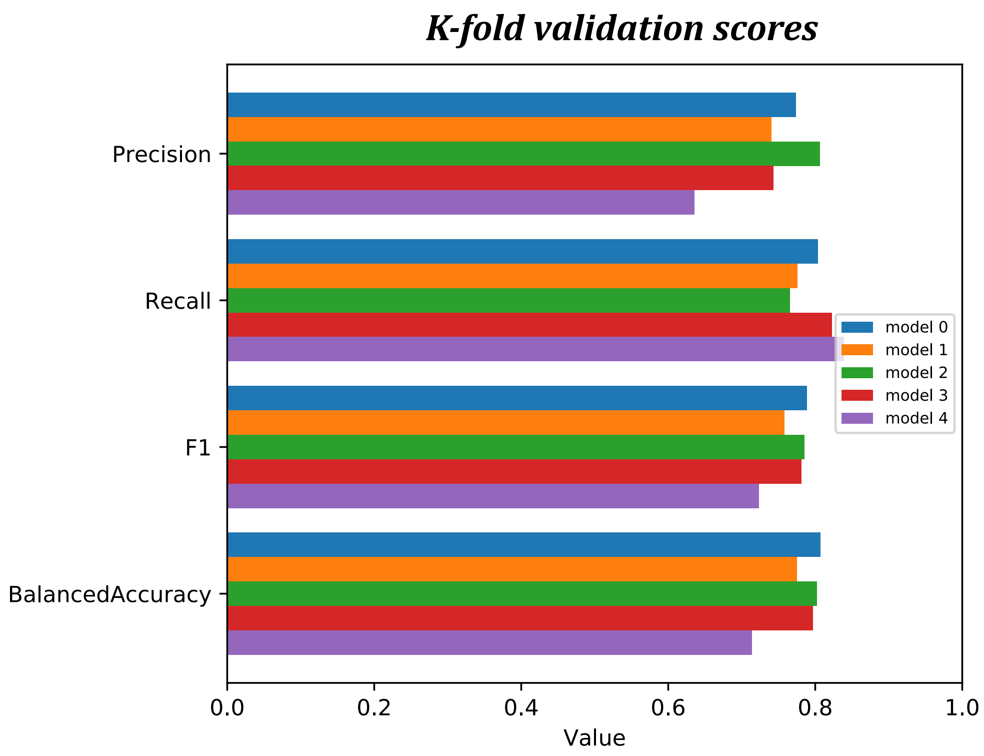
Best choice to have the highest ***recall*** (***F1*** is almost the same of the next threshold)

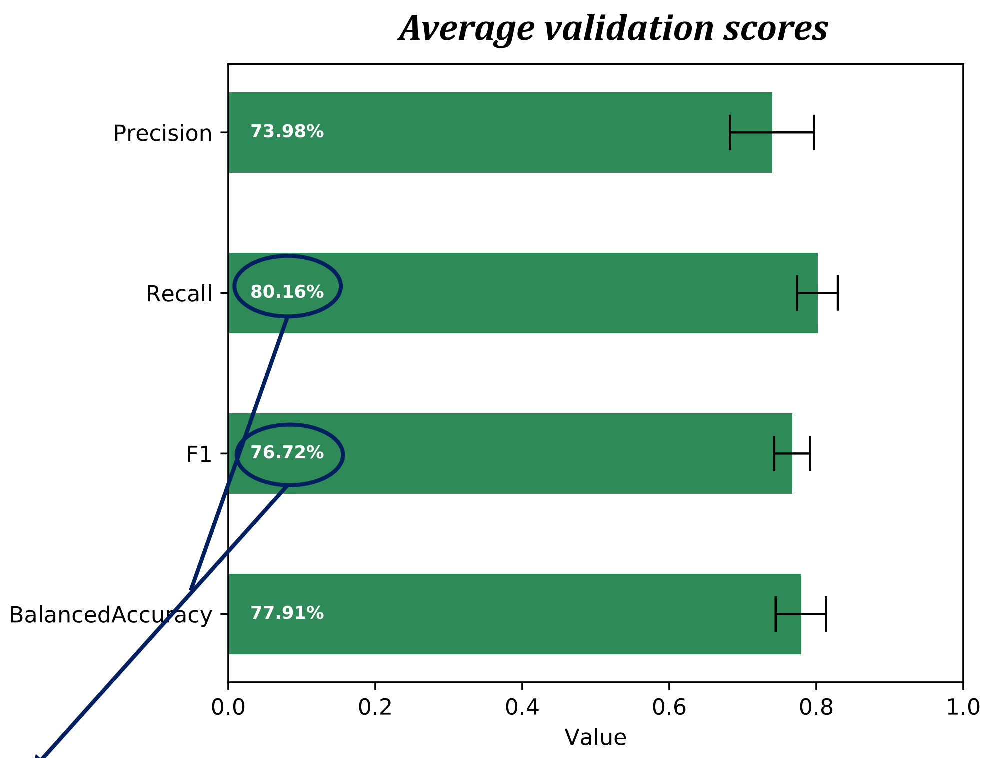
***Scores vs threshold***




**Truth definition: evaluation of the *ratio R*** (change pixels)/(total pixels)
if ***R*** **> 0.15%** **(25 pixels out of 16384) → change**

# Validation scores - classification




High ***recall*** and ***F1*** values. The scores **strongly depends** on the truth definition and specific requirements (in this study, the objective was to have the ***highest possible recall*** while keeping ***precision above ~70%***)

# NVIDIA Jetson AGX Xavier

- **The classification model is faster** and requires less memory occupancy, which make it **ideal for the execution onboard** satellites to allow a fast decision making process.

- Tested both the segmentation and the classification models on a ***low-power consumption GPU:* NVIDIA Jetson AGX Xavier***
  - **64 Tensor Cores** (accelerate dense linear algebra computations)
  - **2 Deep Learning Accelerator cores** (accelerate Neural Network execution efficiently)
  - ***Suitable to be installed onboard future satellite missions***


*available at [Planetek Italia]

# NVIDIA Jetson AGX Xavier

**DLA = Deep Learning Accelerator**
(hardware module)

- ❖ **Two** power modes selected: the **default 15 W** and the **high-performance 30 W** modes, mainly different for:
  - Number of active CPU
  - GPU maximal frequency
  - activation status of DLA cores

**NVPModel clock configuration for Jetson AGX Xavier 16GB and 32GB**

| Property | Mode | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | MAXN | 10W | 15W | 30W | 30W | 30W | 30W | 15W * |
| Power budget | n/a | 10W | 15W | 30W | 30W | 30W | 30W | 15W |
| Mode ID | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Online CPU | 8 | 2 | 4 | 8 | 6 | 4 | 2 | 4 |
| CPU maximal frequency (MHz) | 2265.6 | 1200 | 1200 | 1200 | 1450 | 1780 | 2100 | 2188 |
| GPU TPC | 4 | 2 | 4 | 4 | 4 | 4 | 4 | 4 |
| GPU maximal frequency (MHz) | 1377 | 520 | 670 | 900 | 900 | 900 | 900 | 670 |
| DLA cores | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 |
| DLA maximal frequency (MHz) | 1395.2 | 550 | 750 | 1050 | 1050 | 1050 | 1050 | 0 |
| PVA cores | 2 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| PVA maximal frequency (MHz) | 1088 | 0 | 550 | 760 | 760 | 760 | 760 | 0 |
| Memory maximal frequency (MHz) | 2133 | 1066 | 1333 | 1600 | 1600 | 1600 | 1600 | 1333 |
| SOC clocks maximal frequency (MHz)<br>*All modes* | adsp 300<br>ape 150<br>axi_cbb 408<br>bpmp 896<br>bpmp_apb 408<br>display 800<br>display_hub 400 | | | csi 400<br>host1x 408<br>isp 1190.4<br>nvdec 1190.4<br>nvenc 1075.2<br>nvjpg 716.8<br>pex 500 | | | rce 819.2<br>sce 729.6<br>se 1036.8<br>tsec 1036.8<br>vi 998.4<br>vic 1036.8 | |

# Test setup

- Warmup runs = **50** (GPU set in the same initial state)
- Valid runs = **200** (each consisting of performing inference on the entire batch)
- Batch size = **42** (entire Pisa area)
- Inference time measured and average values calculated **every 20 runs**
- $\boxed{Throughput = \frac{nRuns \times nImages}{elapsedTime}}$
- Benchmark comparison:
  - **TensorFlow** with ***NVIDIA Jetson AGX Xavier***
  - **NVIDIA TensorRT** with ***Jetson AGX Xavier*** (FP16 mode)
  - **TensorFlow** with ***NVIDIA Tesla K40****

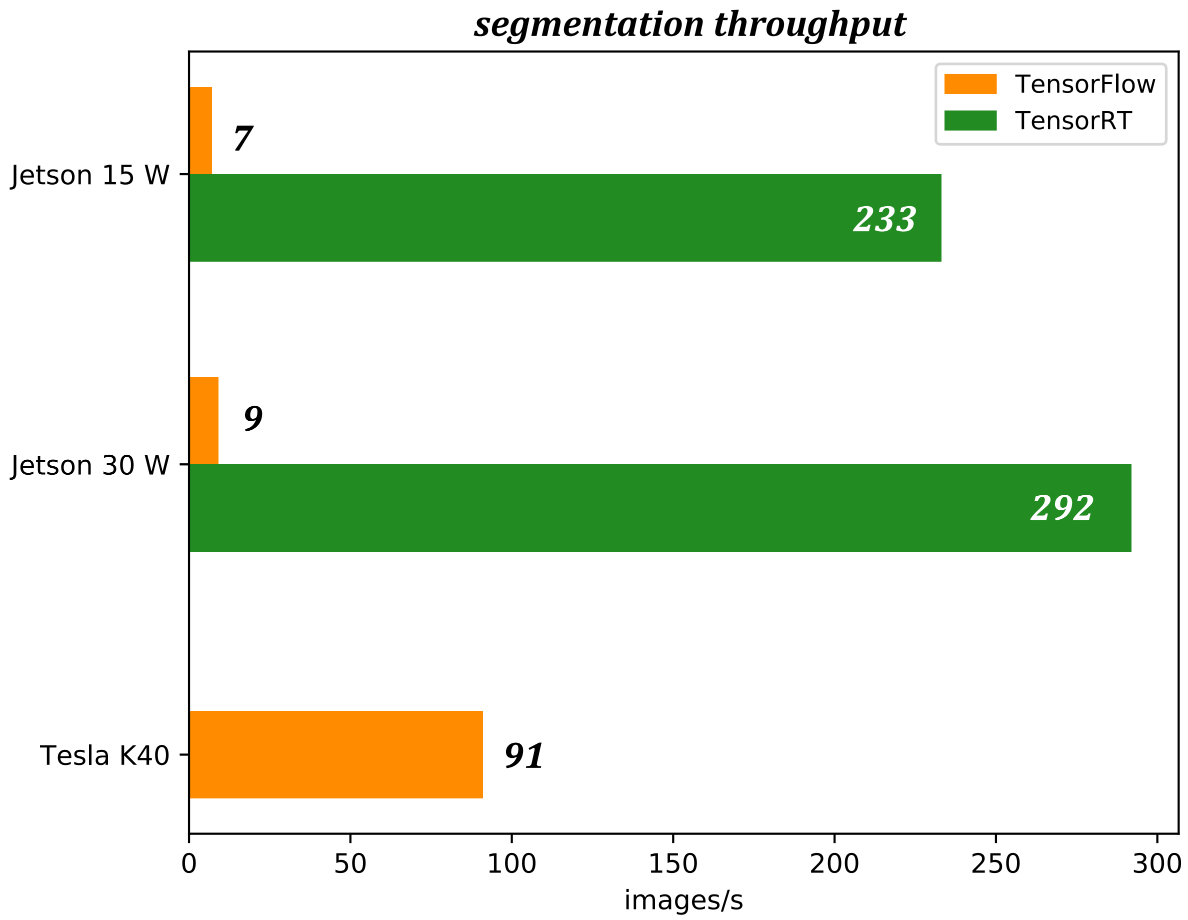
***NVIDIA TensorRT*** *is a software development kit for high-performance deep learning inference on NVIDIA GPUs that delivers low latency and high throughput.*

*available at ReCaS Data Center

# Throughput results - segmentation

- As expected, **TensorRT** shows the highest throughput providing a **32x speedup factor** with respect to **TensorFlow** on Jetson.

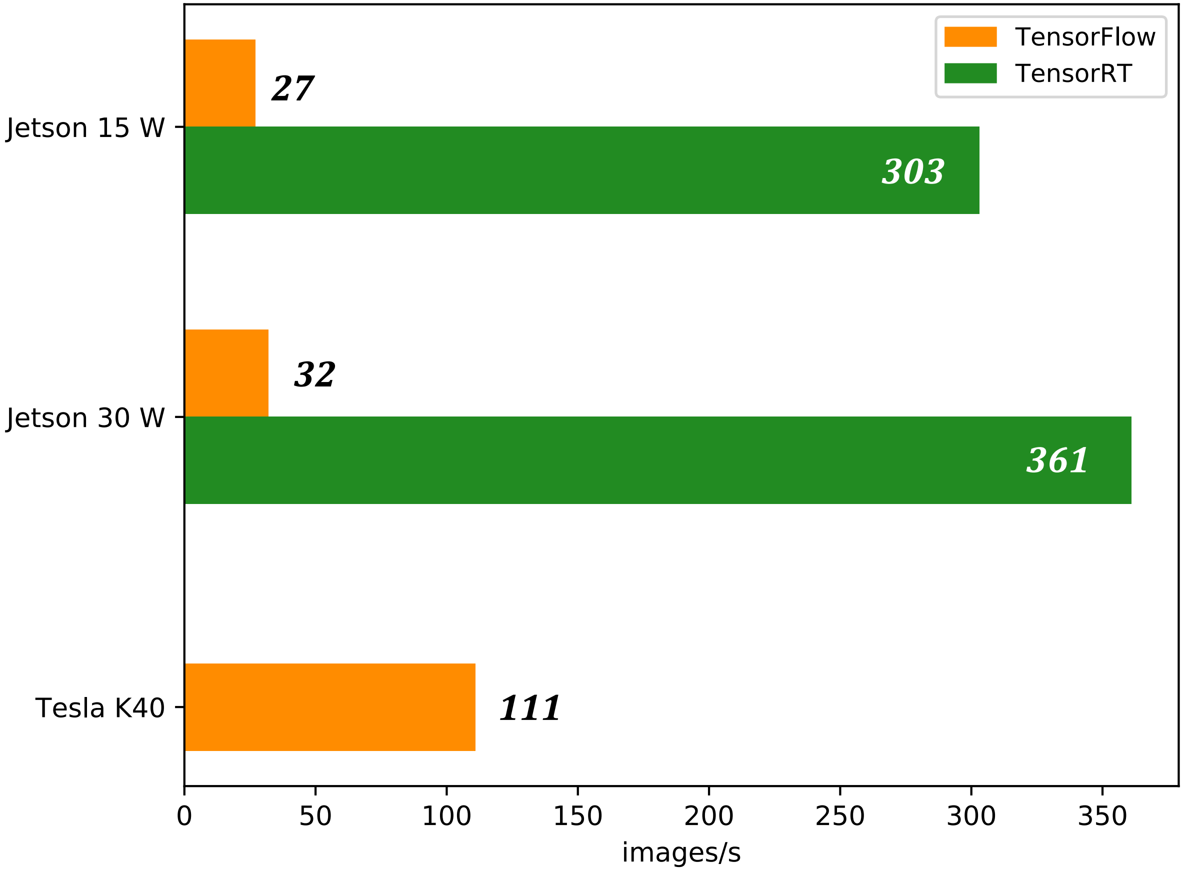
- The throughput provided by **TensorFlow** on a ***high-performance*** **Tesla K40** ***is lower*** than the throughput obtained with **TensorRT** on a ***low-power consumption*** **Jetson** (no matter the power mode)



# Throughput results – classification

- The same result is observed if using a smaller model (CNN for classification)

- **TensorRT** provides a **11x speedup factor** with respect to **TensorFlow** on Jetson. The lower factor with respect to the segmentation model is a key feature of TensorRT, designed to *optimize complex models better than smaller ones* (which usually requires less optimizations).



# Conclusions

- Artificial Intelligence is a **powerful** tool in the Remote Sensing field
  - Allows the satellites to ***autonomously*** perform operations and take decisions

- The change detection task has many **useful applications**:
  - Climate studies
  - Urban growth studies
  - Emergency management
  - Anomaly detection

- On the other hand, heterogeneous platforms onboard can **accelerate** the decision making process
  - Low-power consumption devices are necessary
  - Memory and energy management onboard the payloads must satisfy **hard restrictions** in the space scenario

*“AI is a tool. The choice about how it gets deployed is ours.”*

O. Etzioni

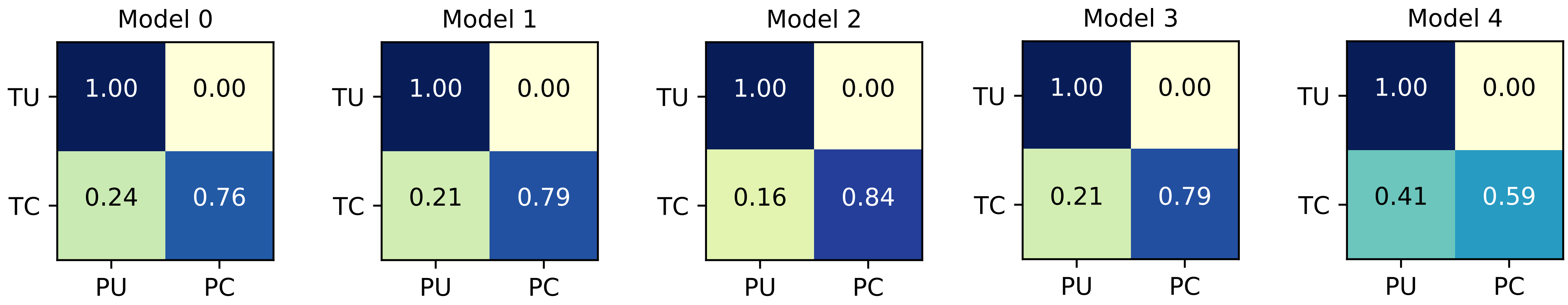
# Thank you for your attention!

# Backup slides

# Backup: Sentinel-2 spectral bands

| | S2A | | S2B | | |
|---|---|---|---|---|---|
| Band Number | Central wavelength (nm) | Bandwidth (nm) | Central wavelength (nm) | Bandwidth (nm) | Spatial resolution (m) |
| 1 | 442.7 | 21 | 442.3 | 21 | 60 |
| 2 | 492.4 | 66 | 492.1 | 66 | 10 |
| 3 | 559.8 | 36 | 559.0 | 36 | 10 |
| 4 | 664.6 | 31 | 665.0 | 31 | 10 |
| 5 | 704.1 | 15 | 703.8 | 16 | 20 |
| 6 | 740.5 | 15 | 739.1 | 15 | 20 |
| 7 | 782.8 | 20 | 779.7 | 20 | 20 |
| 8 | 832.8 | 106 | 833.0 | 106 | 10 |
| 8a | 864.7 | 21 | 864.0 | 22 | 20 |
| 9 | 945.1 | 20 | 943.2 | 21 | 60 |
| 10 | 1373.5 | 31 | 1376.9 | 30 | 60 |
| 11 | 1613.7 | 91 | 1610.4 | 94 | 20 |
| 12 | 2202.4 | 175 | 2185.7 | 185 | 20 |

# Backup: segmentation confusion matrices



# Backup: classification confusion matrices