BIG DATA ARCHITECTURES
Computer vision and MLOPS

Matteo Marulli PhD. student
https://www.disit.org, https://www.snap4city.org

Corso di: Big Data Architectures

Prof. Paolo Nesi

Dep DINFO, University of Florence
Via S. Marta 3, 50139, Firenze, Italy
DISIT Lab, Sistemi Distribuiti e Tecnologie Internet
http://www.disit.dinfo.unifi.it/ , Https://www.disit.org
paolo.nesi@unifi.it http://www.disit.dinfo.unifi.it/nesi

BY NC ND
SOME RIGHTS RESERVED

Who I am

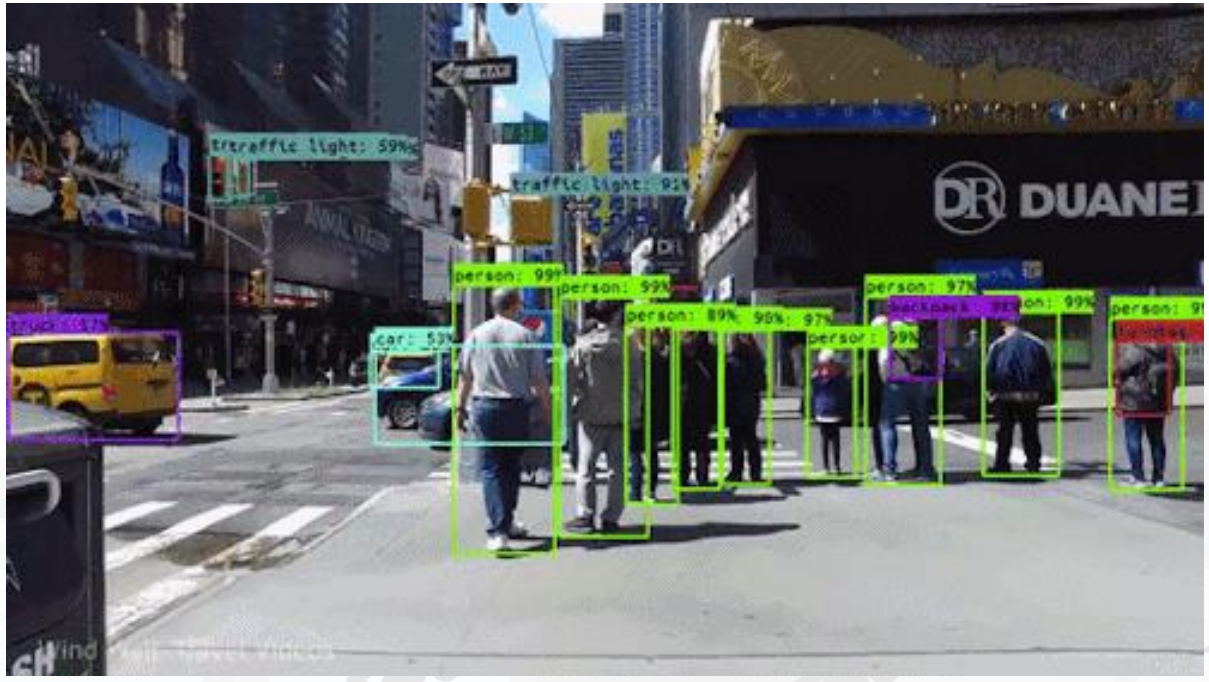
- My name is Matteo Marulli
- I have a master degree in computer science (110/110)
- Now, I'm actually a PhD. Student in Information engineering and working at DISIT-LAB of DINFO
- My research topic is «Artificial Intelligence methods and techniques for text understanding and risk estimation, predictive models for health claims management"
- But I have strong passion about computer vision, computer graphics, machine learning and data mining

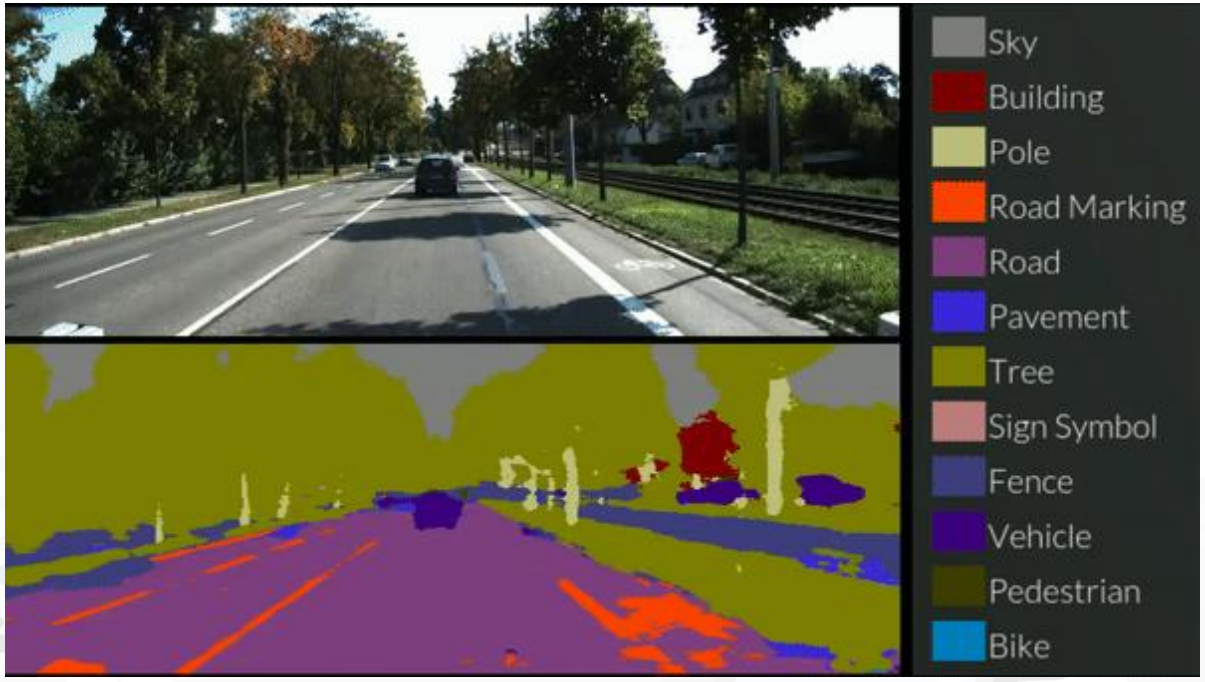
Before we start… who are you?

- Let’s breaking the ice guys
- Join on menti and answer some question please

What is computer vision

Computer vision is a field of artificial intelligence that focuses on the algorithms and techniques enabling computers to comprehend the contents of an image.



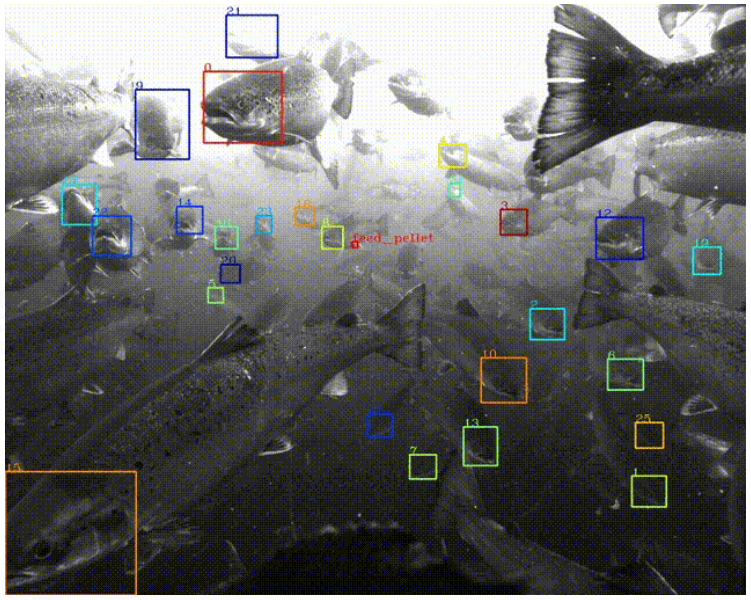


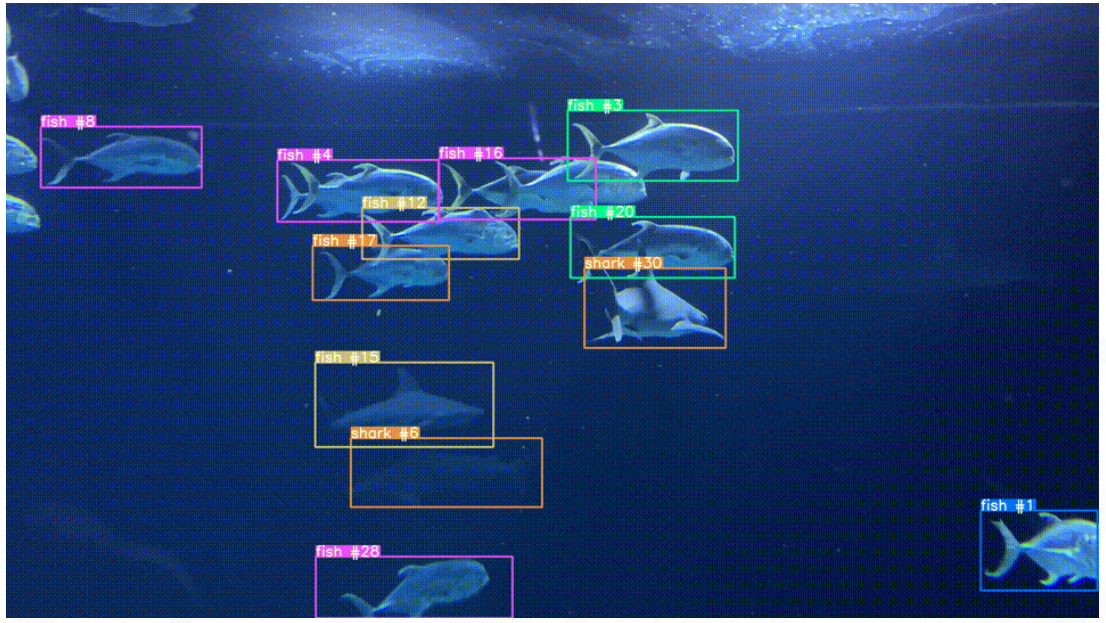
Why is import study computer vision

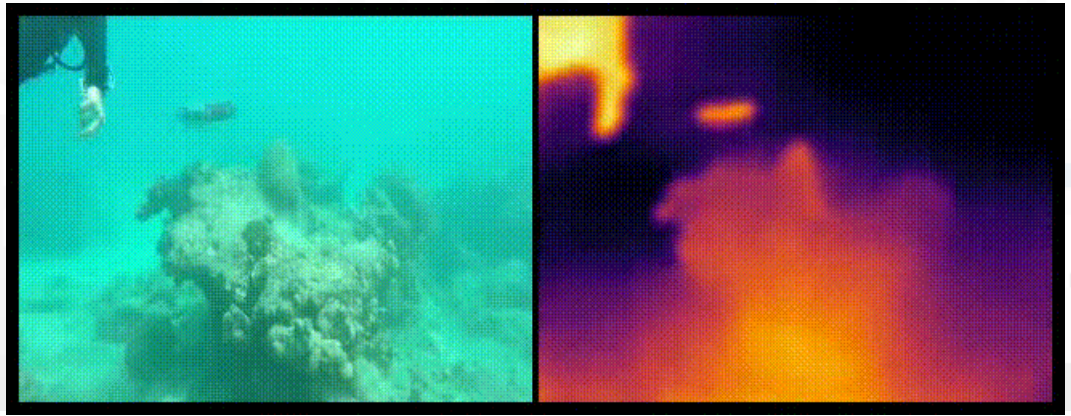
- Computer vision has always found application on the industrial side:
  - Automated Quality Control
  - Predictive maintenance
  - Industrial Robot Guidance
  - And so on…
- Recently, it's becoming more common to use it in smart cities for various purposes:
  - Video surveillance
  - Traffic flow
  - Digital twins
  - And so on…

Computer vision for industries

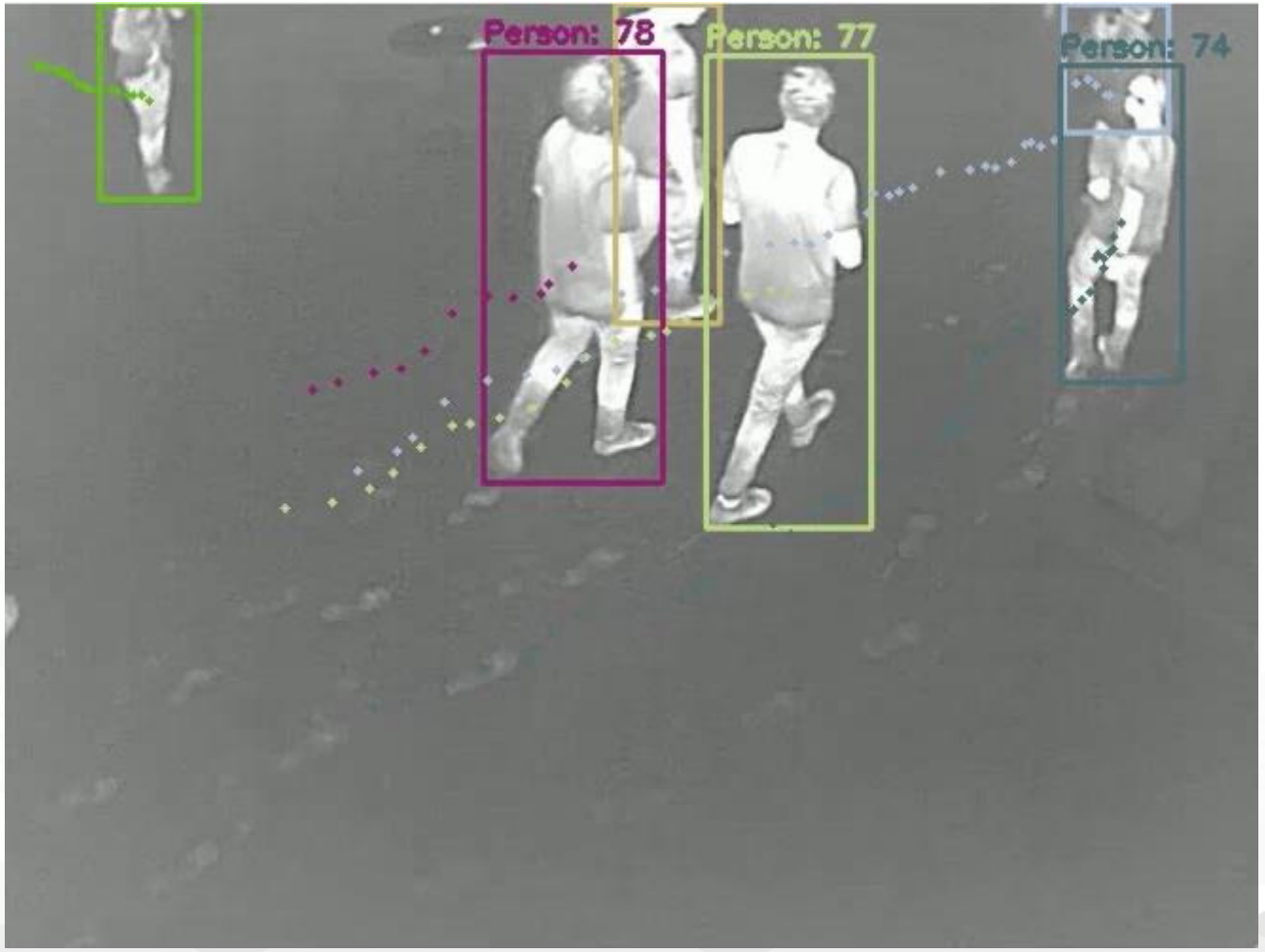
Computer vision in adversarial enviroments



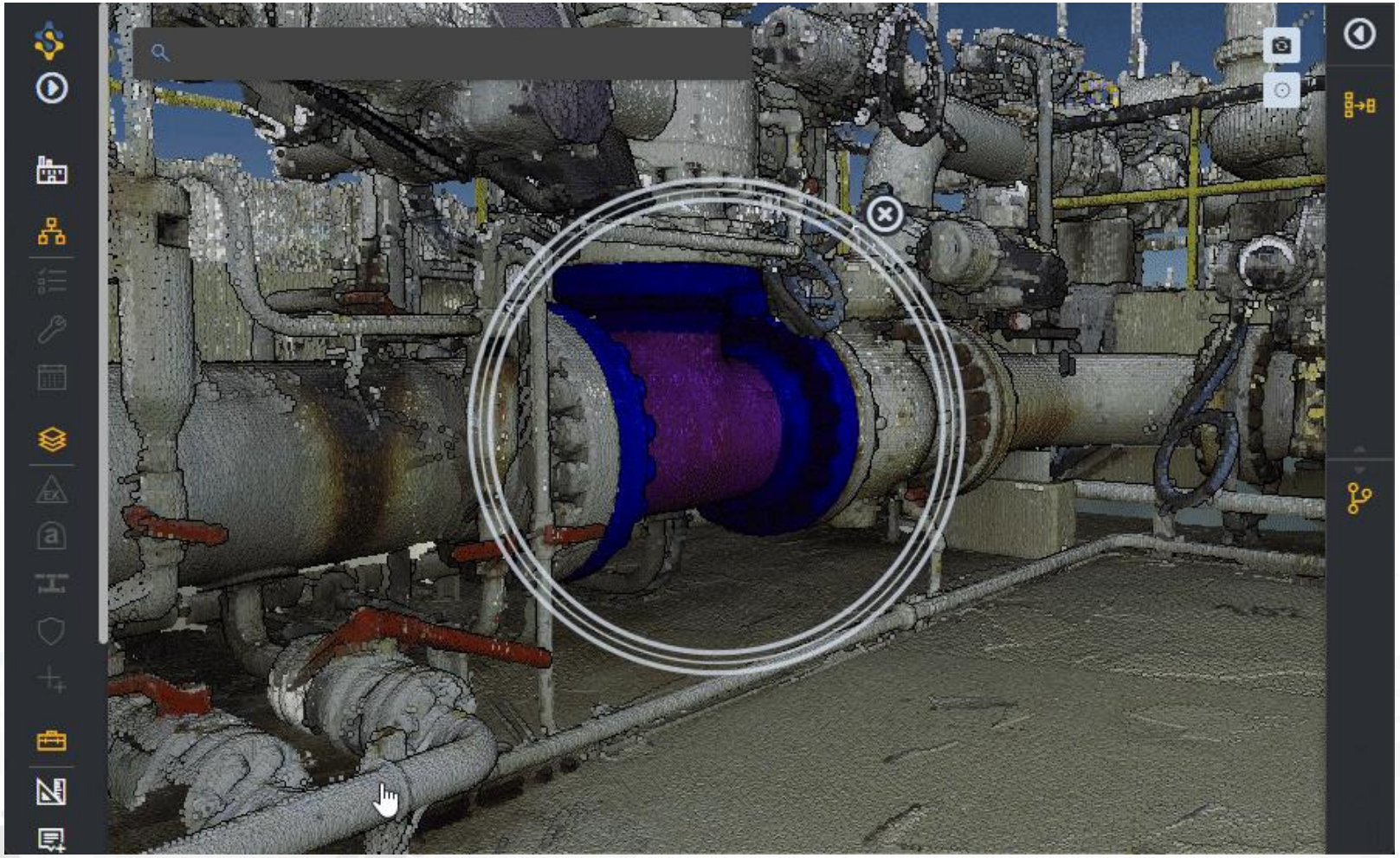




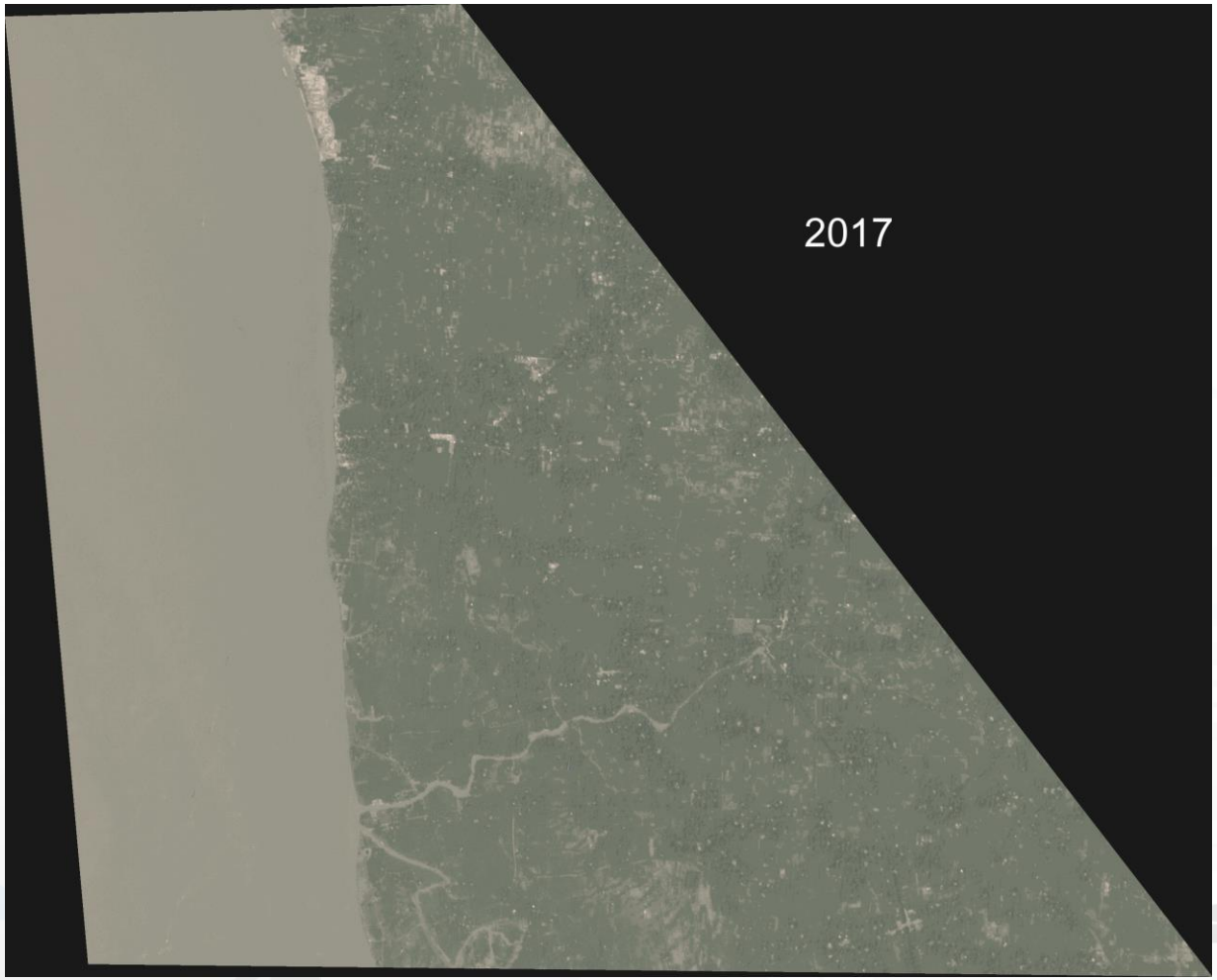
Computer vision for smart surveillance



Computer vision for digital twin

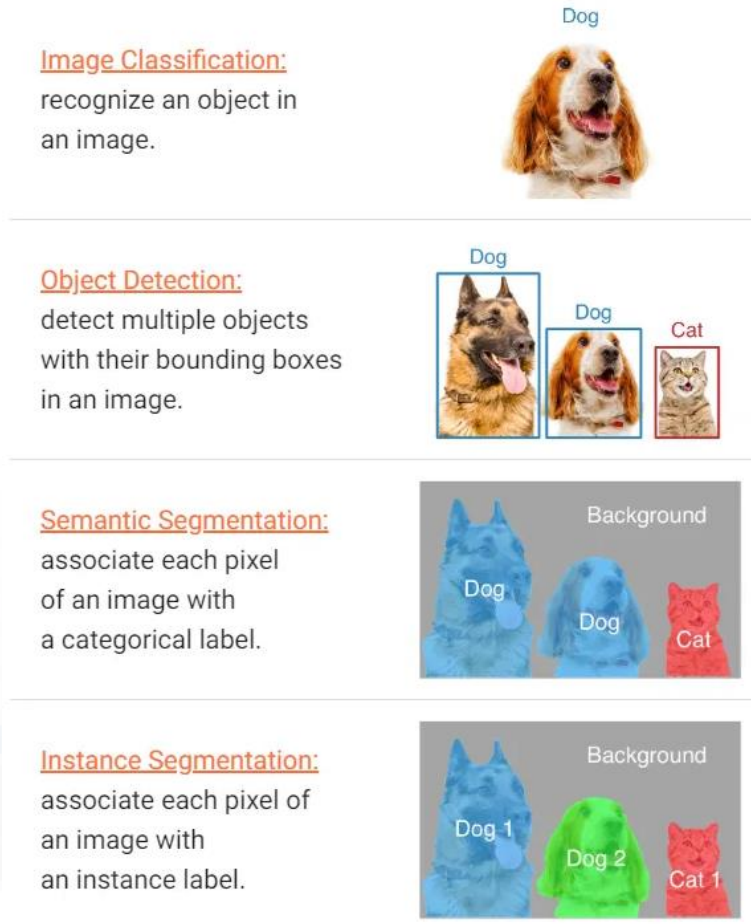


Computer vision satellite images



Computer vision tasks

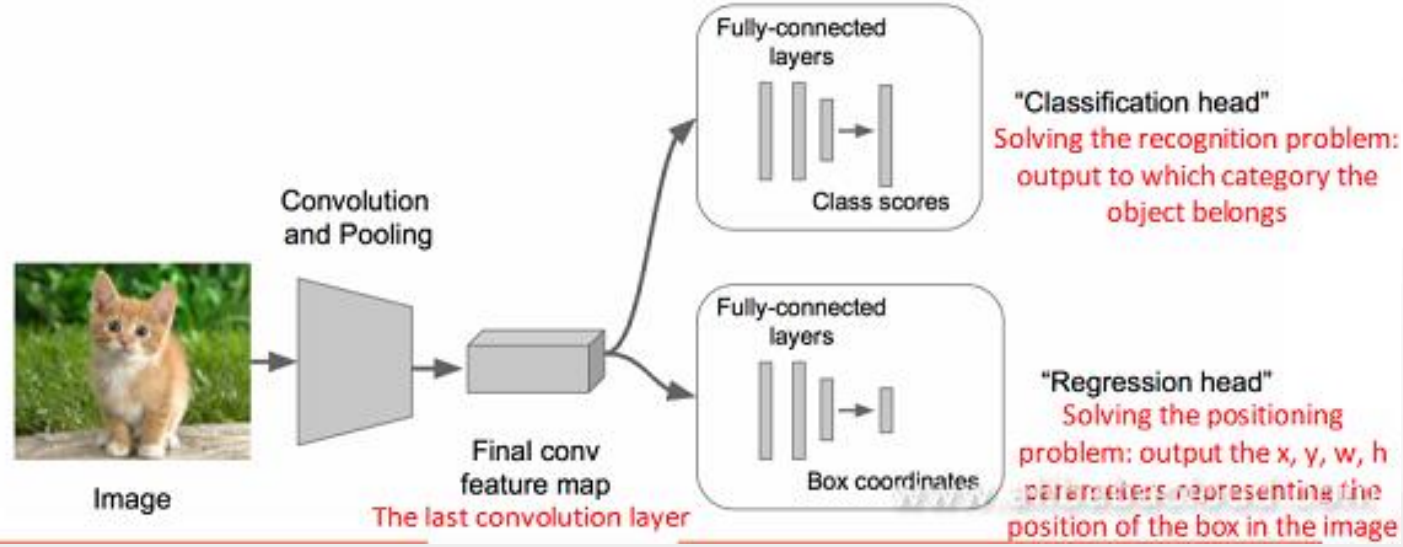
- There are a lot of computer vision tasks such as:
  - image matching
  - Image registration
  - Image stitching
  - …..
- We focus and cover for this seminary only the tasks that enable of analyzing and understanding the content of an image



Object detection

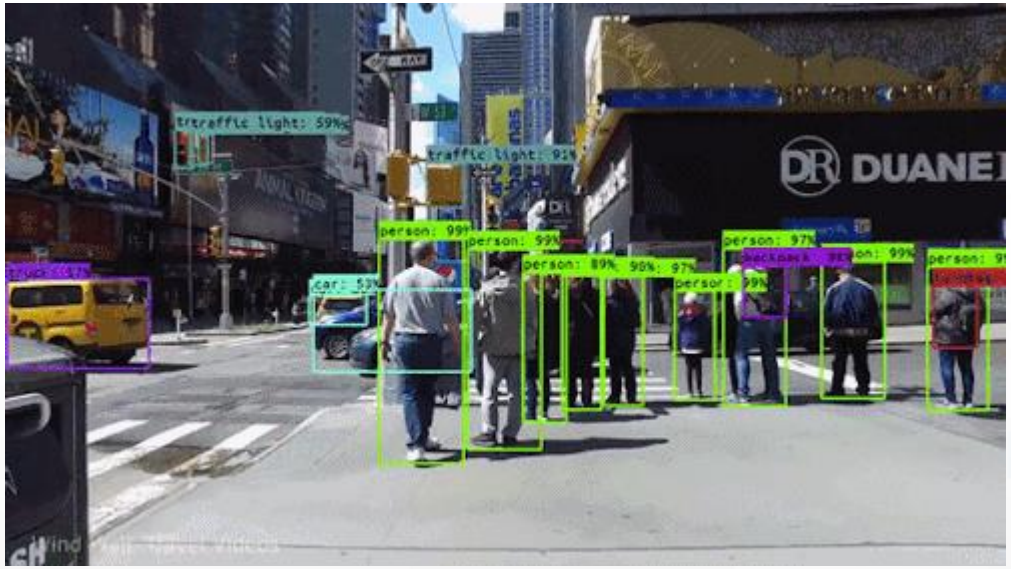
Object detection is a task that allows an algorithm to tell **what** is in an image and tell **where** it is.

- Object detection is a more complicated task than classification because it answers two questions (what, where)
- To answer these two questions it uses both classification (what) and regression (where)

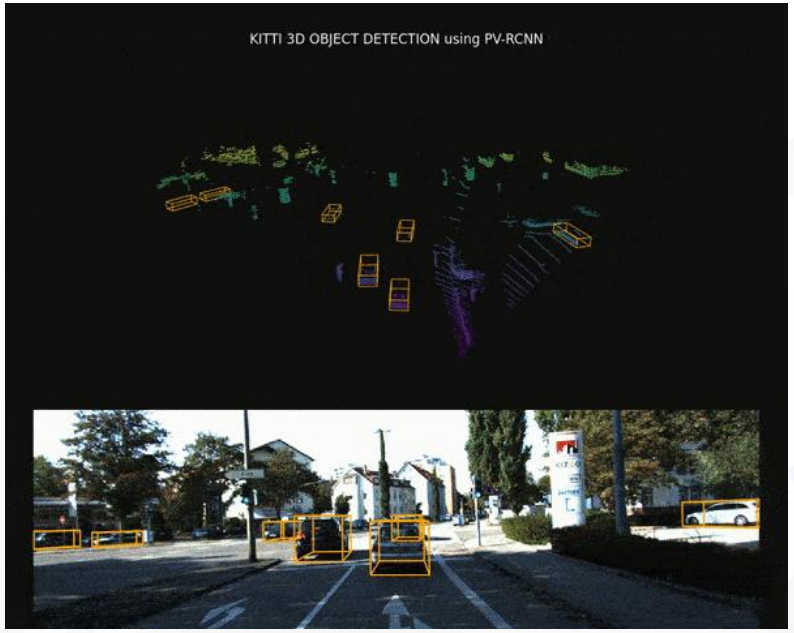


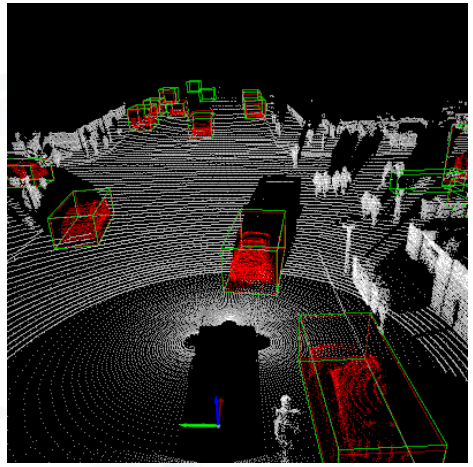
Object detection: types of detection

2D Object detection



3D Object detection



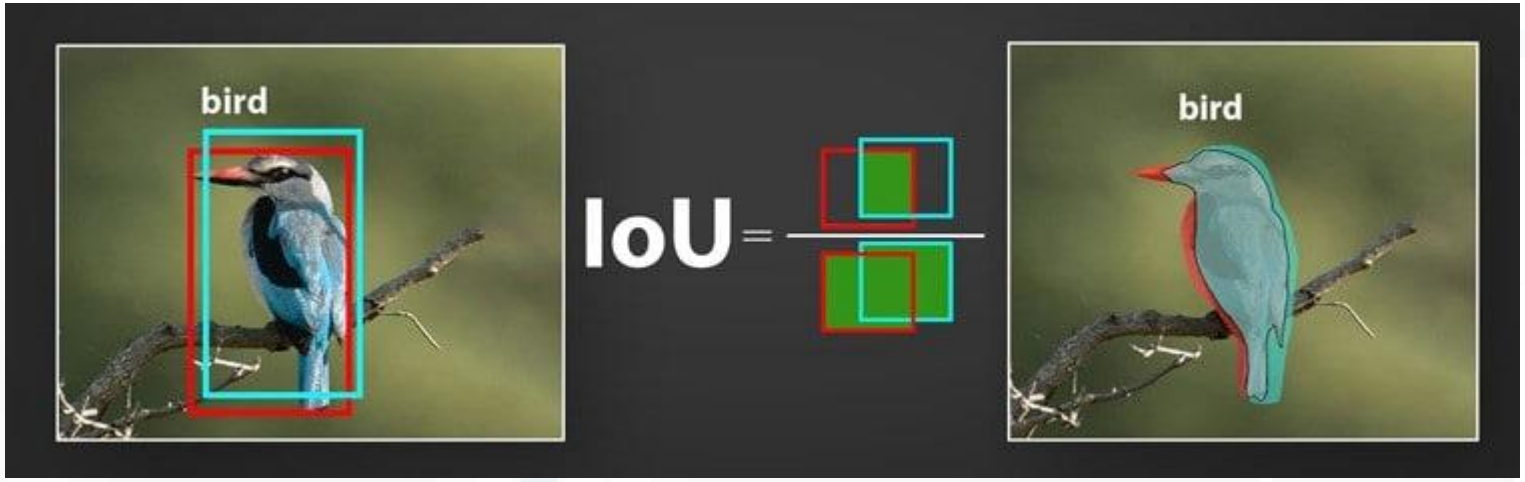


Object detection: metrics

- Like the classification and regression problem, even object detection has its metric that measure the goodness of a detector on your data:
- We see in this order:
  - Intersection over union (IoU)
  - Avarage Precision (AP)
  - Mean avarege precision (mAP)

Intersection over union (IoU)

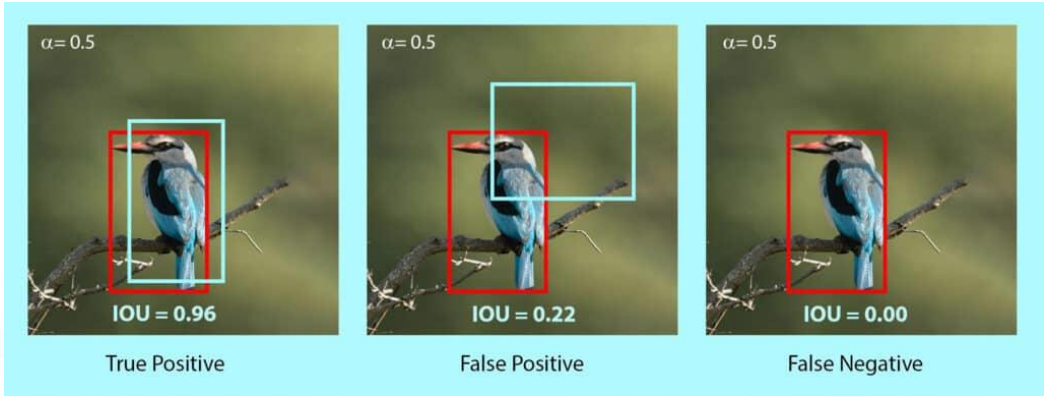
Intersection over union (IoU) is a metric that quantifies the degree of overlap between two regions.



Intersection over union (IoU)

Intersection over union (IoU) is a metric that quantifies the degree of overlap between two regions.

Confusion matrix



- With IoU:
  - A TP case occured when the the intersection between a BB predicted an BB ground truth is $\geq \alpha$
  - A FP case occured when the the intersection between a BB predicted an BB ground truth is $< \alpha$
  - A FN case is when the detection is no occured

Intersection over union (IoU)

Classification

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Object detection

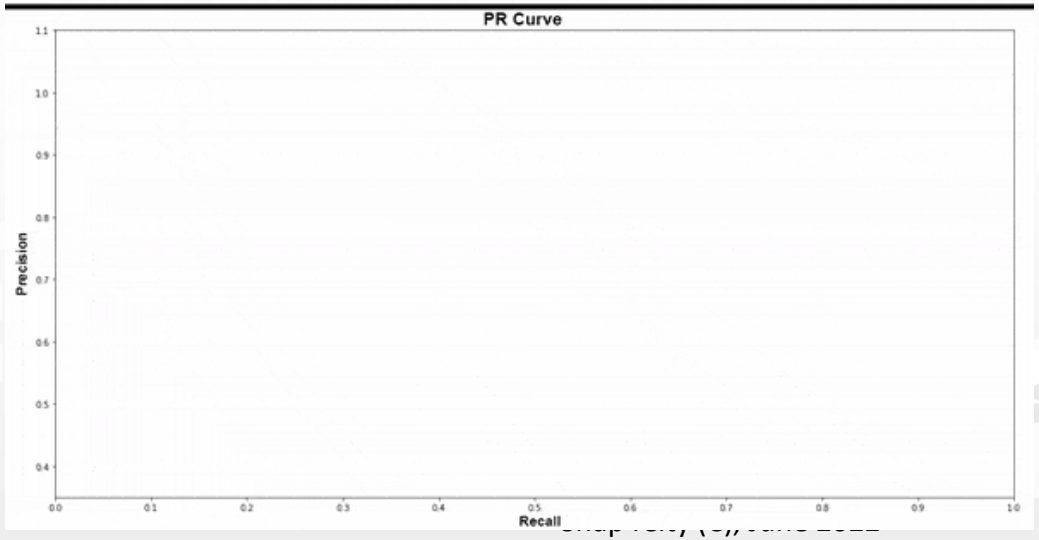
$$Precision = \frac{TP}{\sum Prediction}$$

$$Recall = \frac{TP}{\sum GT}$$

Average Precision (AP)

Average precision **is the area under the precision-recall curve**

Detections							
Precision							
Recall							

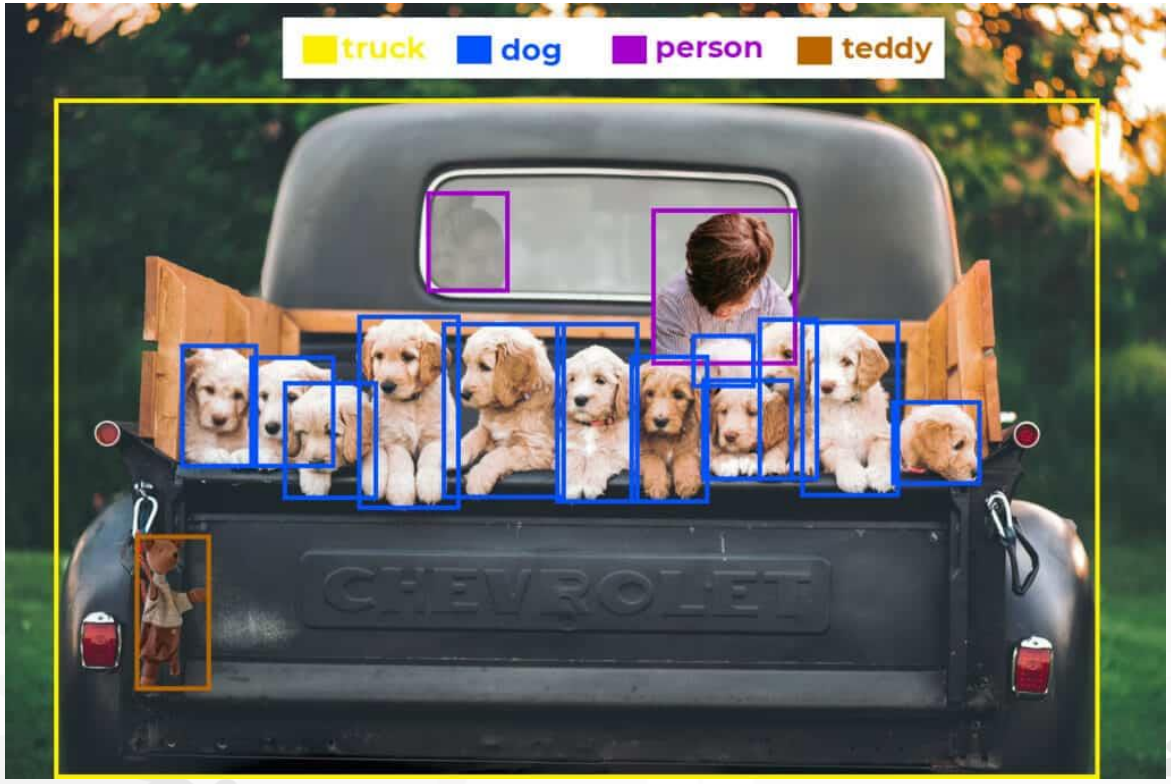


Average Precision (AP)

- Suppose you trained a detector and you have this test-image

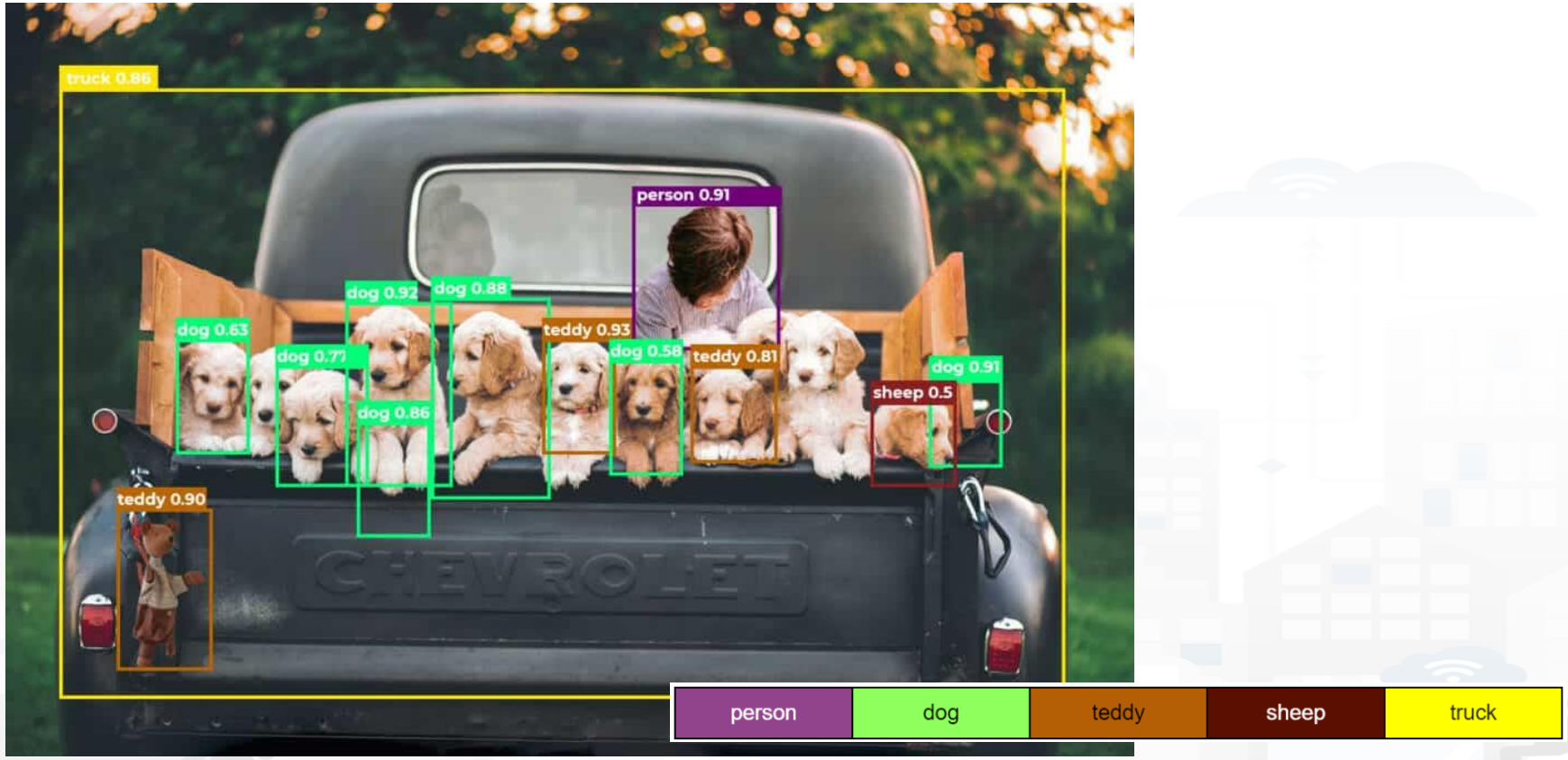
Average Precision (AP)

- For this test image you have the annotation of GT created with a tool



Average Precision (AP)

- Feed the image into detector and this is the output model



Average Precision (AP)

Detections							
Conf.	0.63	0.77	0.92	0.86	0.88	0.58	0.91
Matches GT by IoU?	TP	TP	TP	FP	TP	TP	FP

Preds.	Conf.	Matches	Cumulative TP	Cumulative FP	Precision	Recall
	0.92	TP	1	0	1/(1+0) = 1	1/16 = 0.08
	0.91	FP	1	1	1/(1+1) = 0.5	1/16 = 0.08
	0.88	TP	2	1	2/(2+3) = 0.66	2/16 = 0.16
	0.86	FP	2	2	0.5	0.16
	0.77	TP	3	2	0.6	0.25
	0.63	TP	4	2	0.66	0.33
	0.58	TP	5	2	0.71	0.41

Average Precision (AP)

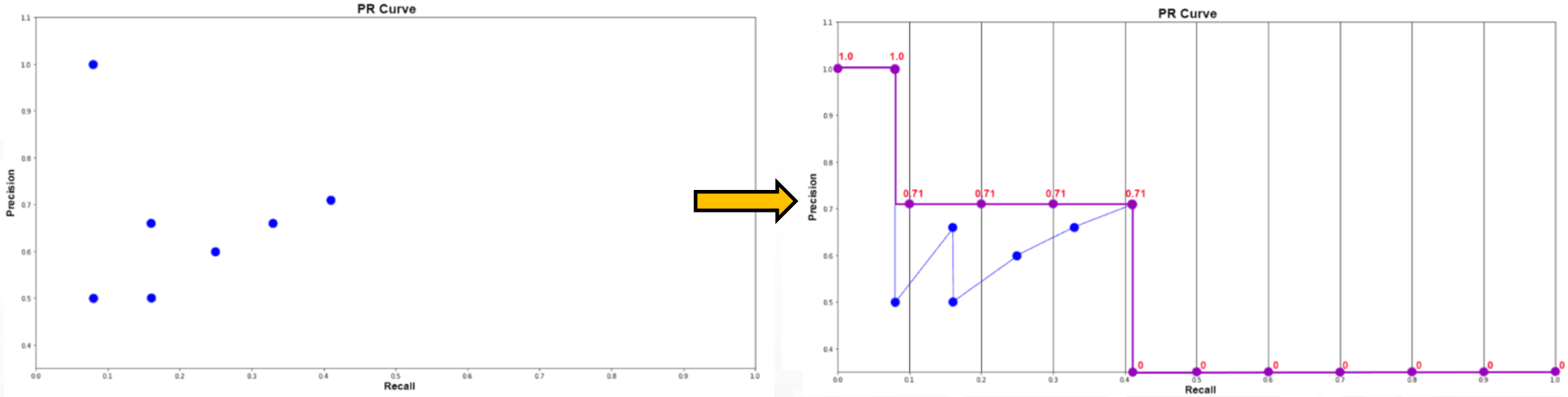
Detections							
Conf.	0.63	0.77	0.92	0.86	0.88	0.58	0.91
Matches GT by IoU?	TP	TP	TP	FP	TP	TP	FP

$$Precision = \frac{TP}{\sum Prediction}$$

$$Recall = \frac{TP}{\sum GT}$$

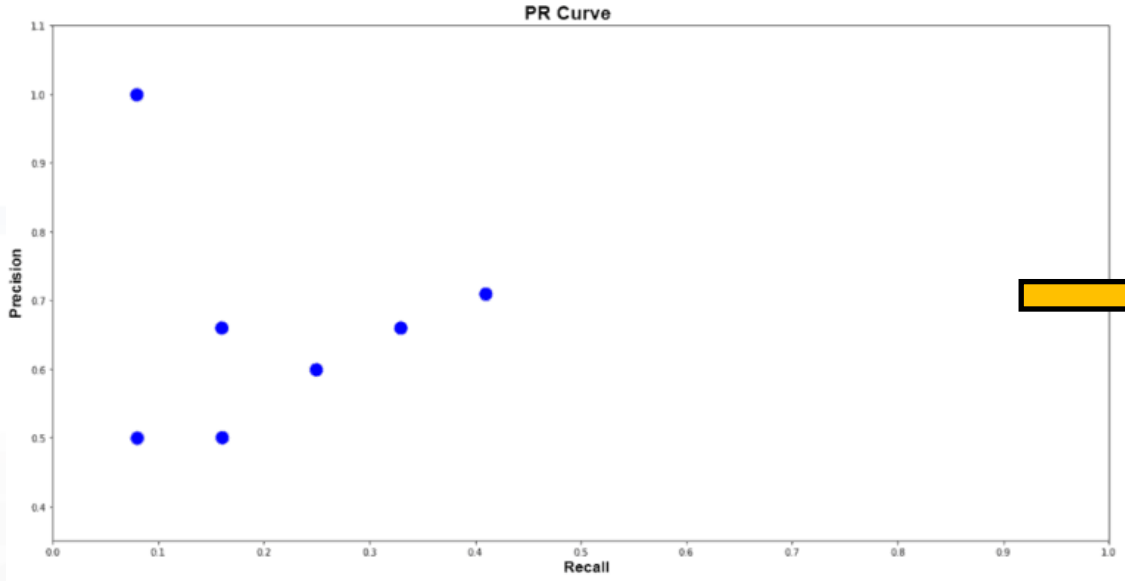
Preds.	Conf.	Matches	Cumulative TP	Cumulative FP	Precision	Recall
	0.92	TP	1	0	1/(1+0) = 1	1/16 = 0.08
	0.91	FP	1	1	1/(1+1) = 0.5	1/16 = 0.08
	0.88	TP	2	1	2/(2+3) = 0.66	2/16 = 0.16
	0.86	FP	2	2	0.5	0.16
	0.77	TP	3	2	0.6	0.25
	0.63	TP	4	2	0.66	0.33
	0.58	TP	5	2	0.71	0.41

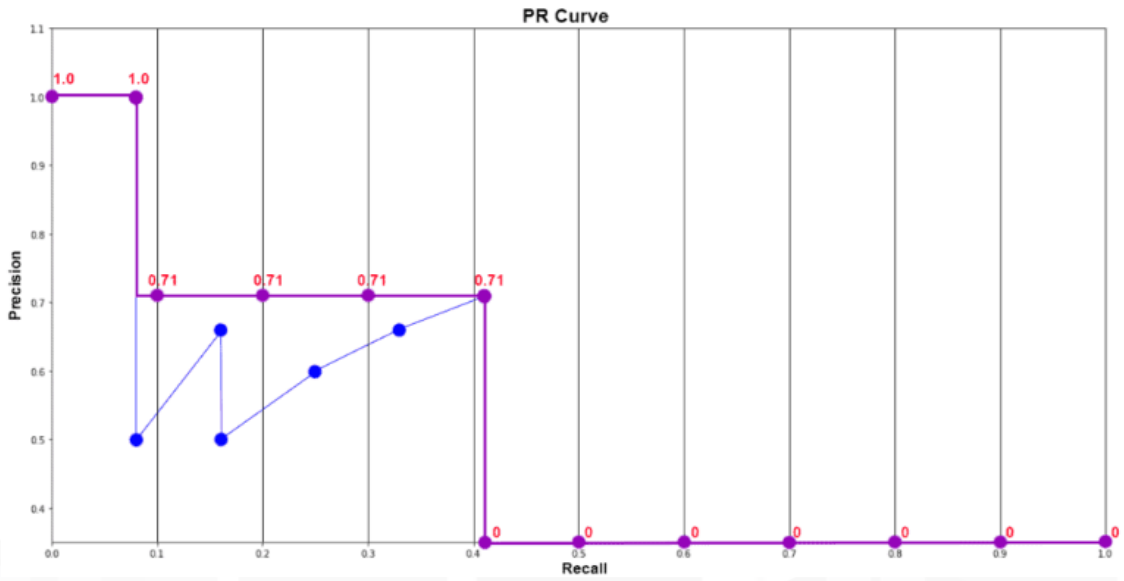
Average Precision (AP)



- Precision values are interpolated across 11 Recall values, i.e., 0, 0.1, 0.2, 0.3,…,1.0.
- The interpolated Precision is the **maximum Precision corresponding to the Recall value greater than the current Recall value**.
- The choice of 11 Recall values is done to avoid creating a noise curve

Average Precision (AP)





- $AP_{dog} = \frac{1}{11} * (1 + 4 * 0.71 + 6 * 0) = 0.349 = 34.9\%$

Mean average precision (mAP)

- AP is an indicator for one class of the performance of detector
- If we want have a global idea of the performance of detector we have to move to mean Average Precision (mAP)
- mAP take in consideration all the classes and it is very simple to compute if you know how compute AP

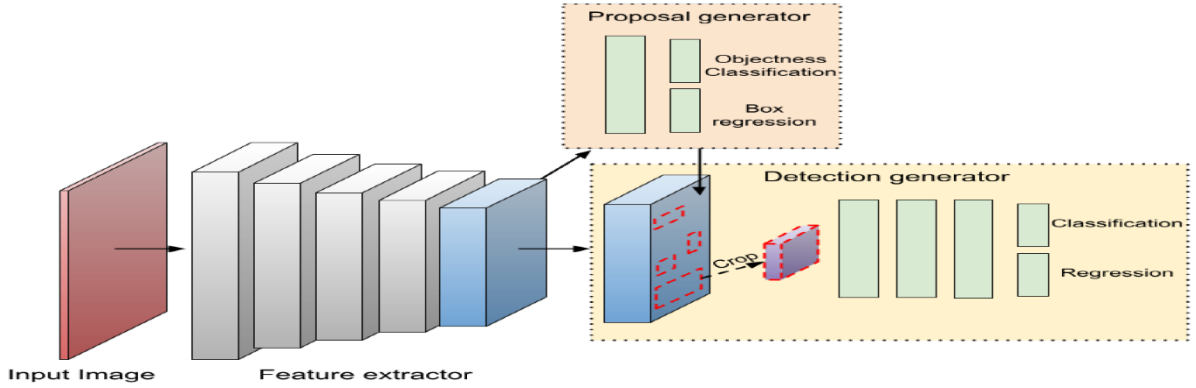
CLASS	dog	person	sheep	truck	teddy
AP	0.349	0.545	0.00	1.00	0.50

$$mAP = \frac{1}{k} * \sum_{i=1}^{k} AP_i = \frac{0.349 + .0545 + 0 + 1 + 0.5}{5} = 0.4788 = 47.88\%$$

- By varying the threshold of the IoU, different mAPs can be obtained, for example mAP@50 is obtained by imposing a threshold at 0.5, mAP@95 is obtained by imposing a threshold at 0.95, even more, one can average these mAPs and denote with mAP[.5:.95]
- the choice of threshold value depend on your application, if you want that the predicted BB correspond as much as possible to BB of ground truth you need to set an higher value for threshold otherwise 0.5 is a good and common choice

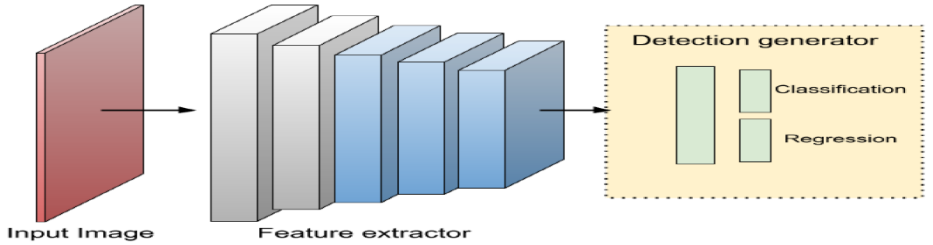
Object detection

Two-stage detector



(a) Two-stage Faster R-CNN

One-stage detector



(b) One-stage RetinaNet

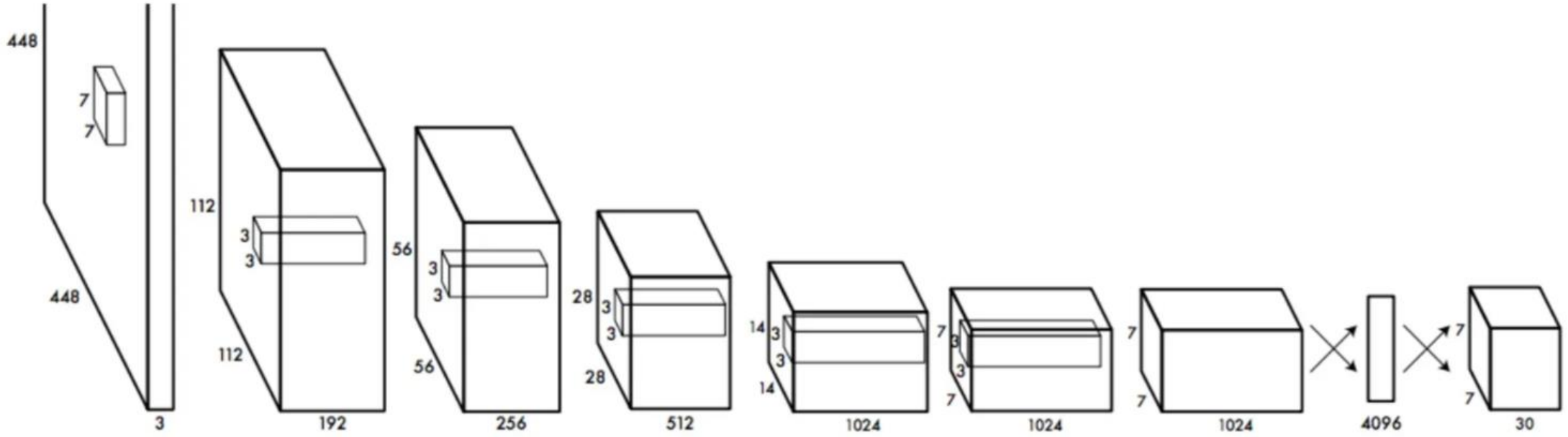
- The net is composed by two moduls:
  - A modul for region proposal
  - A module for predict the BBs
- \+ Detection with high precision
- \- Not suitable for real-time system

- the net is entirely responsible for finding the proposed regions and calculating the bounding boxes
- ‡ Detection with good precision
- \+ Suitable for real-time system

You Only Look Once (YOLO)

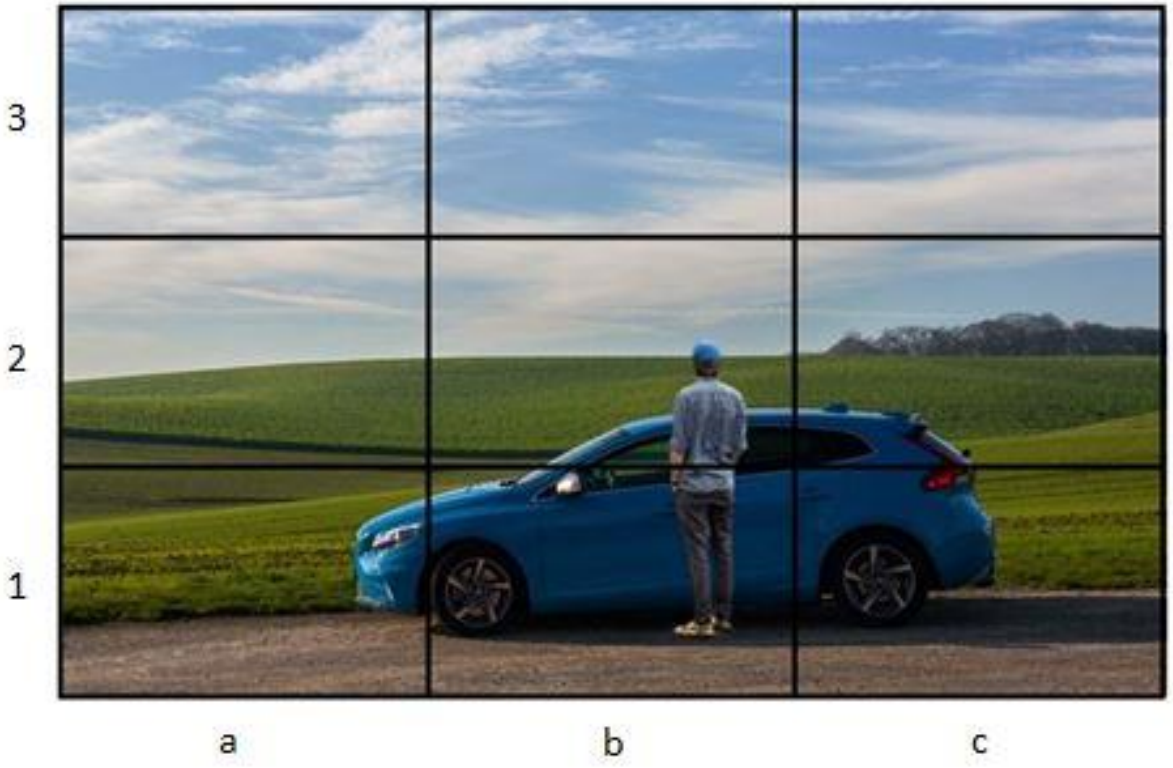
- Yolo Only Look Once best know as YOLO, it is one most famous one-stage detector
- It is not the most accurate, but it's one of the fastest when it comes to prediction times.
- YOLO is a network that has been used in many real-world projects:
  - Video surveillance
  - Drone systems
  - Mobile phone
  - Embedding systems

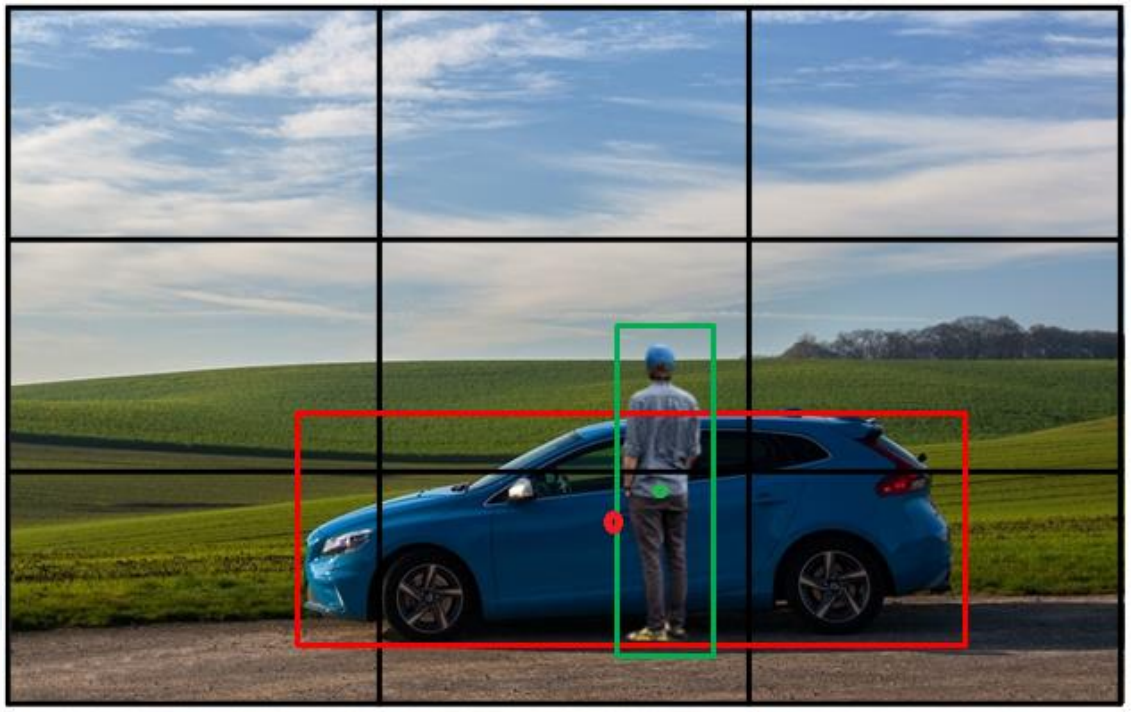
You Only Look Once (YOLO)



You Only Look Once (YOLO)

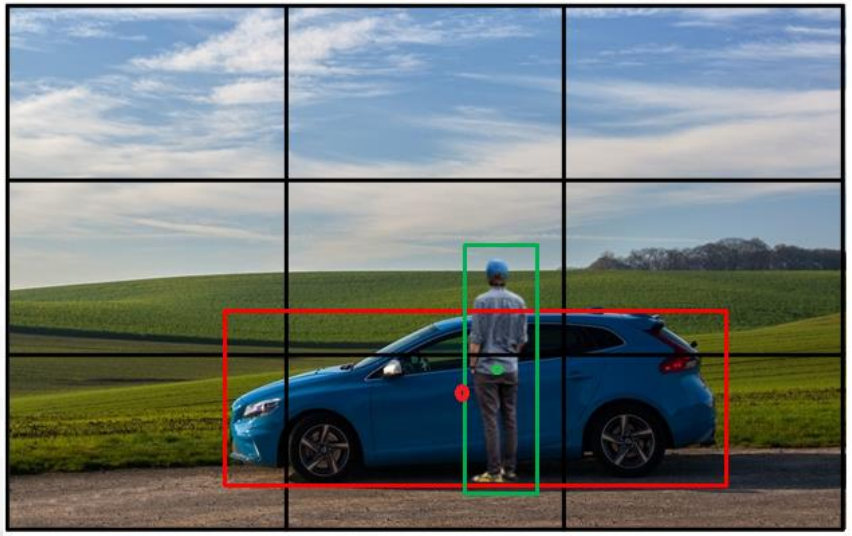
- YOLO divides the input image into an SxS (3x3) grid
- If the center of an object falls into a grid cell the cell is responsible for detecting that object whose center (the red and green dots) falls inside the grid cell





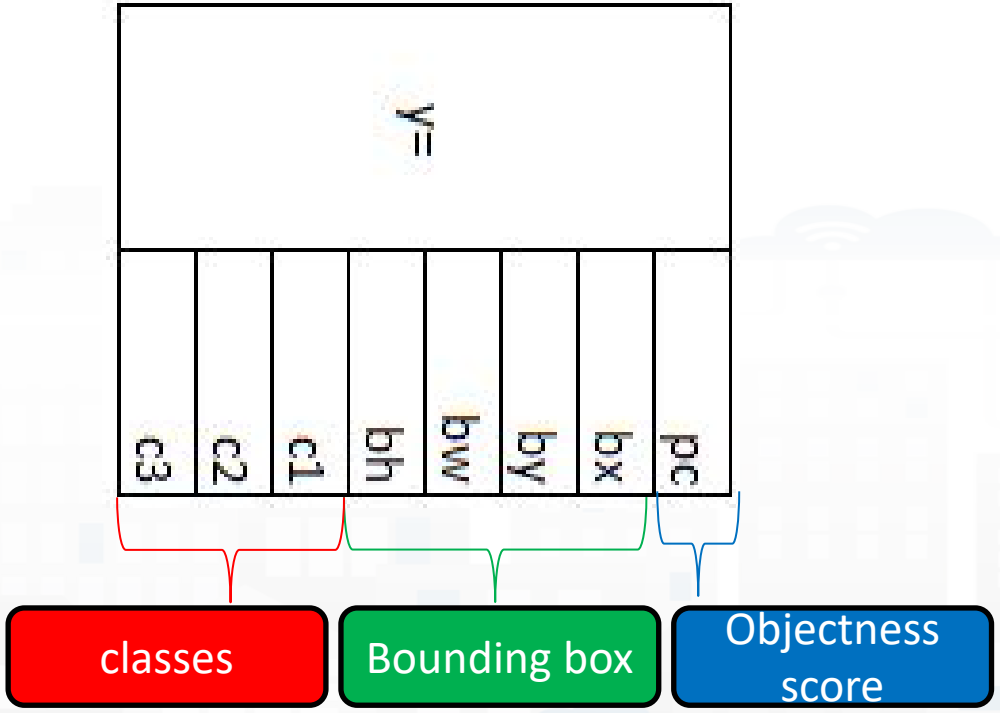
You Only Look Once (YOLO)

- Each cell predicts a fixed number B (5) of bounding boxes and confidence scores for those boxes (the one-object rule limits how close detected objects can be)
- Bounding Box coordinates(bx, by, bw, bh):
  - width bw and height bh are normalized by the image width and height
  - bx and by are offsets to the corresponding cell
- Confidence score pc: confidence that the box contains an object + box accuracy: $\Pr(Object) * IoU_{truth\_pred}$



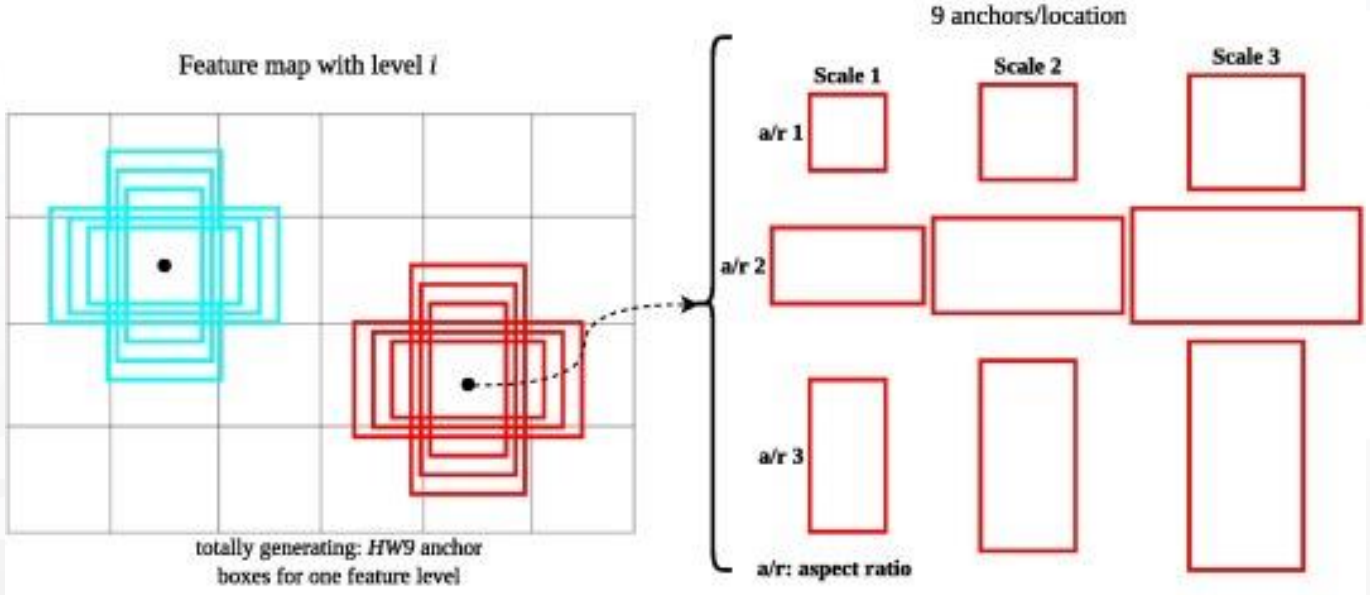
You Only Look Once (YOLO)

The output for a detection



You Only Look Once (YOLO)

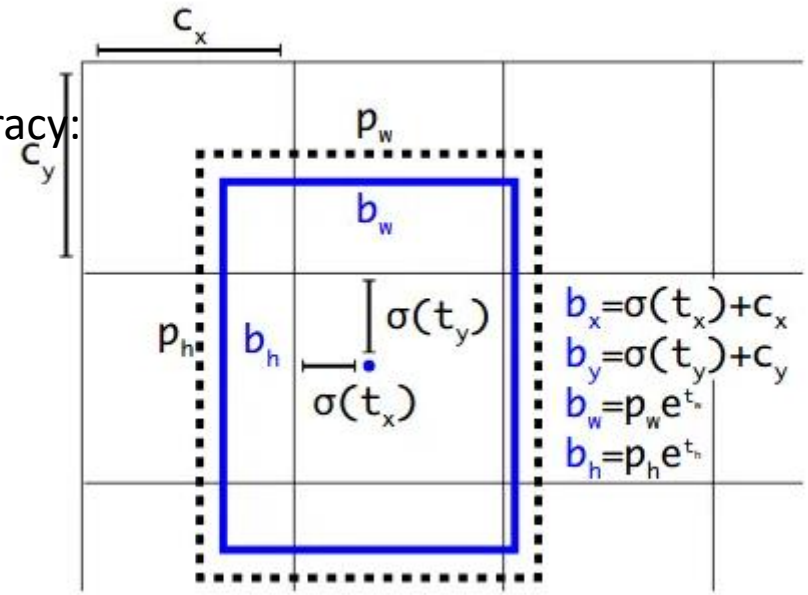
- Anchor boxes are standard bounding boxes
- There are several and they are different for aspect ratio and scale
- They are used to make bounding box predictions



You Only Look Once (YOLO)

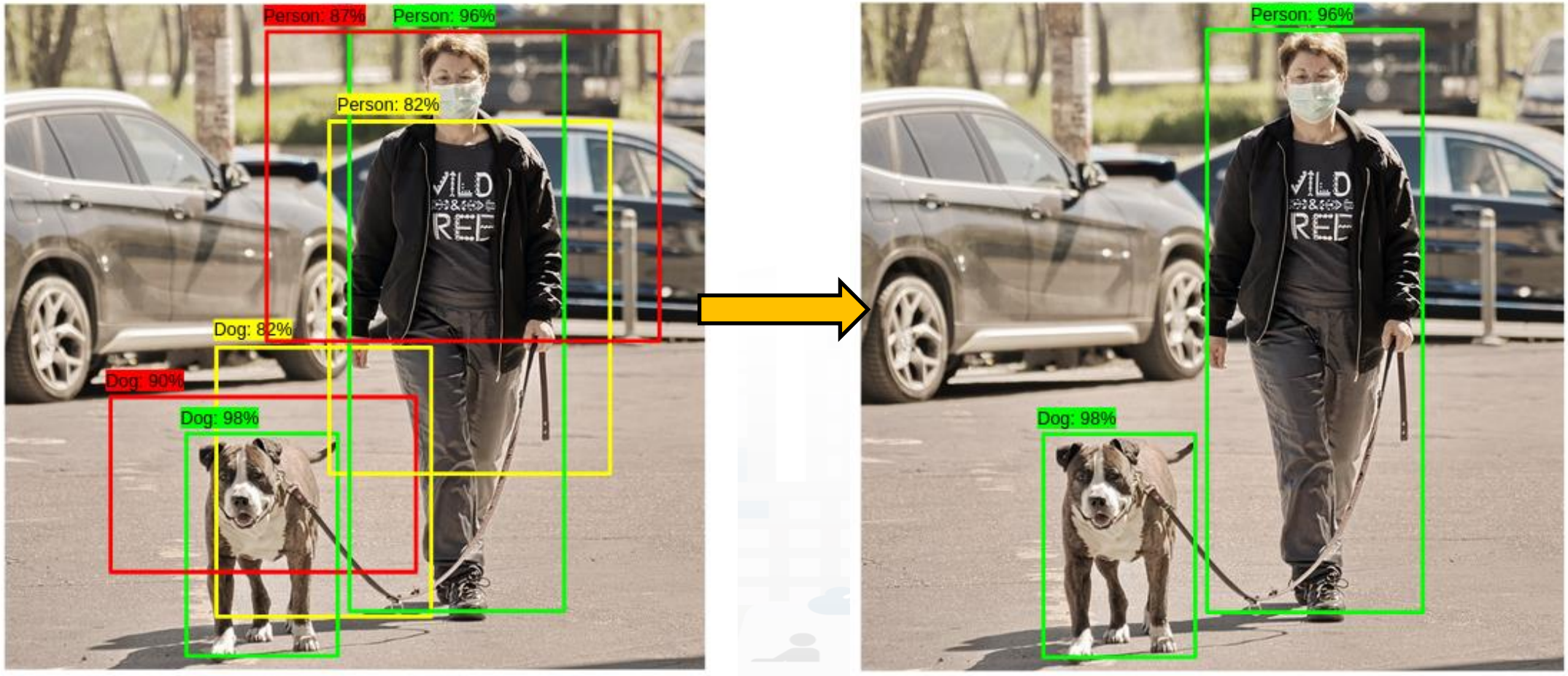
- YOLOv2 model makes use of anchor boxes like Faster R-CNN: pre-defined bounding boxes with useful shapes and sizes that are tailored during training.
- Bounding Box coordinates(x,y,w,h):
  - width w and height h are normalized by the image width and height
  - x and y are offsets to the corresponding cell
- Confidence score: confidence that the box contains an object + box accuracy:

$$
\begin{aligned}
b_x &= \sigma(t_x) + c_x \\
b_y &= \sigma(t_y) + c_y \\
b_w &= p_w e^{t_w} \\
b_h &= p_h e^{t_h} \\
Pr(\text{object}) * IOU(b, \text{object}) &= \sigma(t_o)
\end{aligned}
$$



- Cx and cy are the top-left point of the cell
- From YOLO2 the sigmoid function is used for computing the offset
- pw and ph are respectively the width and height of the anchor box that will be scaled by multiplying for pw and ph

You Only Look Once (YOLO)



You Only Look Once (YOLO)

Algorithm 1 Non-Max Suppression

```
procedure NMS(B,c)
    B_nms ← ∅                                   Initialize empty set
    for b_i ∈ B do                              Iterate over all the boxes
        discard ← False                         Take boolean variable and set it as false. This variable indicates whether b(i) should be kept or discarded
        for b_j ∈ B do                          Start another loop to compare with b(i)
            if same(b_i, b_j) > λ_nms then      If both boxes having same IOU
                if score(c, b_j) > score(c, b_i) then
                    discard ← True              Compare the scores. If score of b(i) is less than that of b(j), b(i) should be discarded, so set the flag to True.
        if not discard then                     Once b(i) is compared with all other boxes and still the discarded flag is False, then b(i) should be considered. So add it to the final list.
            B_nms ← B_nms ∪ b_i
    return B_nms                                Do the same procedure for remaining boxes and return the final list
```

You Only Look Once (YOLO)

$$L_{loc} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

$$L_{cls} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} (1_{ij}^{obj} + \lambda_{noobj}(1 - 1_{ij}^{obj}))(C_{ij} - \hat{C}_{ij})^2 + \sum_{i=0}^{S^2} \sum_{c \epsilon C} 1_i^{obj} (p_i(c) - \hat{p}_i(c))^2$$

$$L = L_{loc} + L_{cls}$$

- The yolo loss is a multi loss
  - A locational loss
  - A classification loss
- This loss must be minimised

New YOLO models

- YOLO is a family of algorithms that has evolved over time.
- One of the most widely used versions is YOLOV5 (2020) of ultralytics but the latest versions are YOLOV8 (2023) of Ultralytics and YOLO-NAS (2023) of Decima

Object Detection Performance Comparison (YOLOv8 vs YOLOv5)

Model Size	YOLOv5	YOLOv8	Difference
Nano	28	37.3	+33.21%
Small	37.4	44.9	+20.05%
Medium	45.4	50.2	+10.57%
Large	49	52.9	+7.96%
Xtra Large	50.7	53.9	+6.31%

*Image Size = 640

Model	mAP	Latency (ms)
YOLO-NAS S	47.5	3.21
YOLO-NAS M	51.55	5.85
YOLO-NAS L	52.22	7.87
YOLO-NAS S INT-8	47.03	2.36
YOLO-NAS M INT-8	51.0	3.78
YOLO-NAS L INT-8	52.1	4.78
YOLOv8 S	47.0	4.0
YOLOv8 M	50.0	6.5
YOLOv8 L	52.5	9.5

Object tracking

Object tracking is a task of computer vision that allows a system to assign an id and track an object in a video

- This task is very important in the real world because systems that can perform object tracking allows:
  - tracking objects and people for security reasons
  - counting people or objects in a room for security or data analysis purposes
  - counting people or objects in a room for security or data analysis purposes
  - understanding how far people are
  - Investigate people's movements to analyze internal mobility at a hospital or shopping centre, this analysis will yield metrics such as origin-destination matrices.

Object tracking: metrics

- Higher Order Tracking Accuracy (HOTA) combines measures of localization accuracy and identity switches to assess tracking quality incorporating both spatial accuracy and temporal consistency.
- $HOTA_{tp}$ represents true positive associations between ground truth objects and predicted tracks
- $HOTA_{fn}$ represents false negative associations, where ground truth objects are not matched with any predicted tracks
- $HOTA_{IDSW}$ represents false negative associations, where ground truth objects are not matched with any predicted tracks represents the number of identity switches, where the same object is assigned different identities or different objects are assigned the same identity.
- $HOTA_{identity}$ evaluates the accuracy of object localization by considering true positive associations and penalizing false negatives and identity switches
- $HOTA_{localization}$ measures the consistency of object identities by penalizing identity switches.

$$\mathrm{HOTA} = \sqrt{\mathrm{HOTA}_{localization} \times \mathrm{HOTA}_{identity}}$$

$$HOTA_{identity} = 1 - \frac{HOTA_{IDSW}}{HOTA_{tp} + HOTA_{IDSW}}$$

$$HOTA_{localization} = \frac{HOTA_{tp}}{HOTA_{tp} + HOTA_{fn} + HOTA_{IDSW}}$$

Object tracking: metrics

$$MOTA = 1 - \frac{\sum(\text{missed targets} + \text{false positive} + \text{missmatch})}{\sum(\text{ground truth targets})}$$

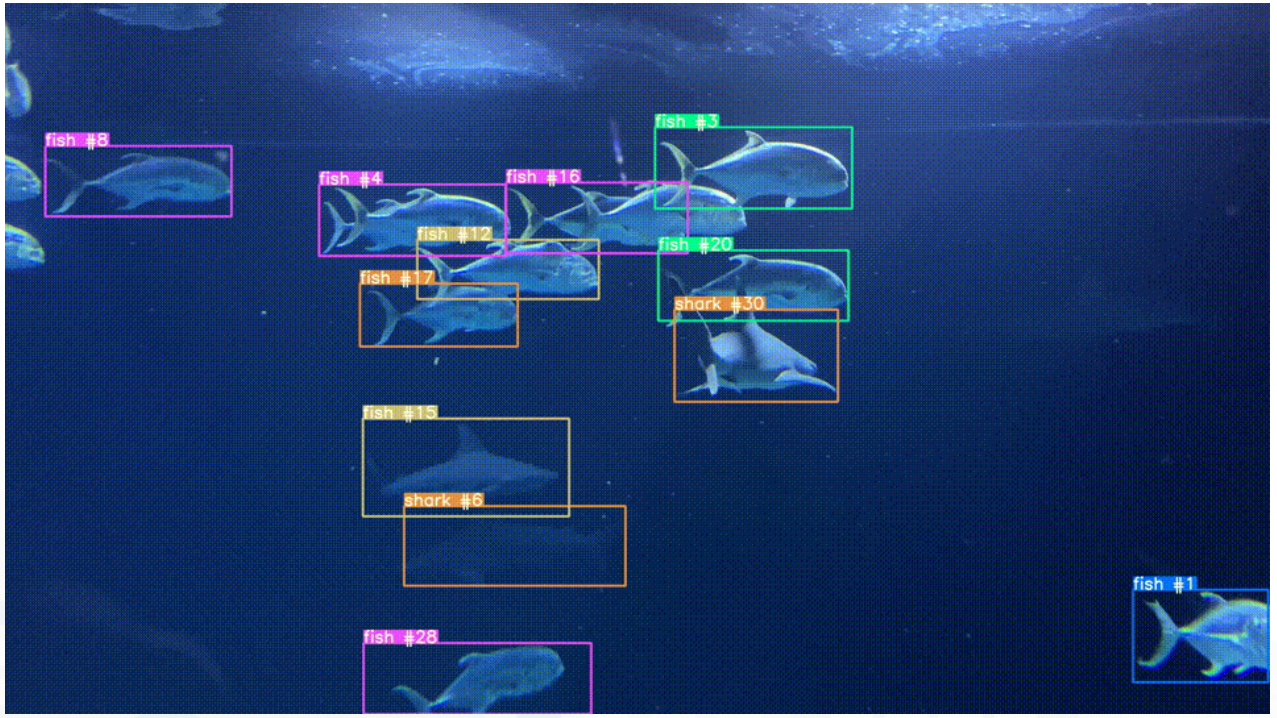
- Multiple Object Tracking Accuracy (MOTA) provides an overall assessment of tracking accuracy by considering various aspects of tracking errors.
- Here, the numerator represents the sum of missed targets (objects that were not detected), false positives (incorrectly detected objects), and mismatches (incorrectly associated objects). The denominator represents the total number of ground truth targets. A MOTA score close to 1 indicates accurate tracking with minimal errors, while a score close to 0 or negative values indicates poor tracking performance.

Object tracking

- The next algorithms are all based on a scheme called SORT and can be schematised as follows
  1. From the initial frame extract bounding boxes with a detector
  2. Use bounding boxes to extract patches of objects that have a detection in the frame
  3. Use a cnn network to calculate feature maps for each patch
  4. Initialises the database with feature maps and assigns each map an ID
  5. Take the next frame f_i from the source
     1. From the f_i extract bounding boxes with a detector
     2. Use bounding boxes to extract patches of objects that have a detection in f_i
     3. Use a cnn network to calculate feature maps for each patch
        1. For fm in feature_maps
           1. matching =false
           2. For id in db.ids
              1. fm_track = db[id]
              2. if matching(fm, fm_track) >= thresh then matching= True assign_id_to_bb(id)
           3. If not matching
              1. new_id = db.last_id +1
              2. insert_to_db(fm, new_id)

DeepSort and OC-Sort

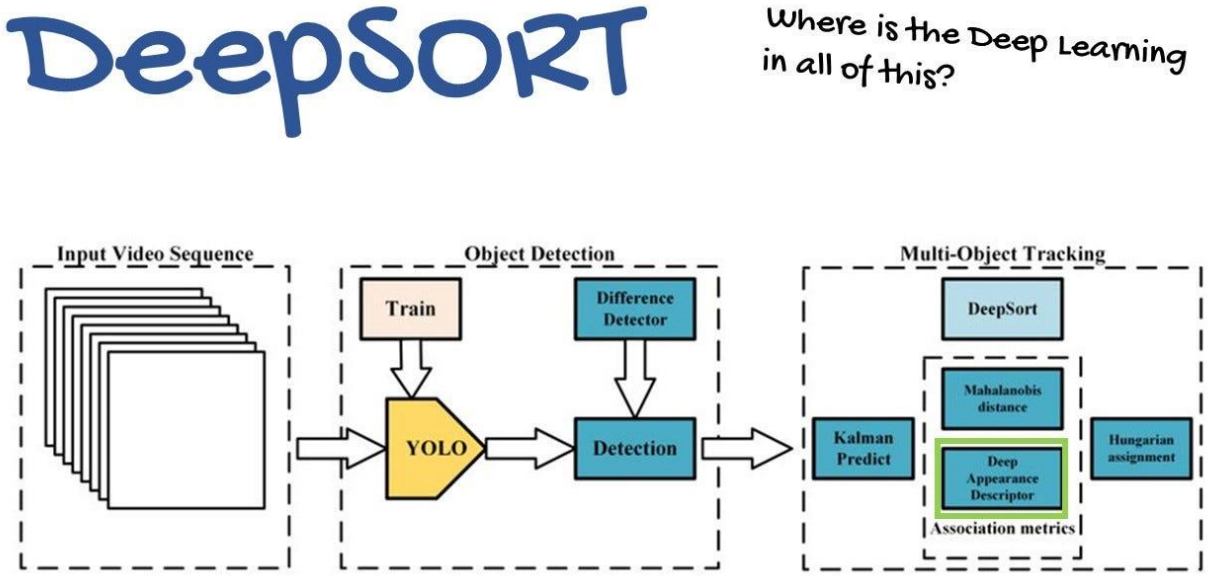
- DeepSort by Wojke et. al 2017 is a machine learning approch for tracking people, assigning IDs to each person.
- Traditionally the tracking algorithms presents the limitation that if a person hid behind an object and then reappeared, it is assigned a different ID.
- DeepSort solves this problem by using an AI model that compares similarity between people, thus reducing the issue of switching people's identities.



DeepSort and OC-Sort

DeepSort steps for tracking detections:

1. Compute bounding boxes using a detector
2. Use Sort (Kalman filter) and ReID (identification model) to link bounding boxes and tracks
3. If no link can be made, a new ID is assigned and it is newly added to tracks.



DeepSort and OC-Sort

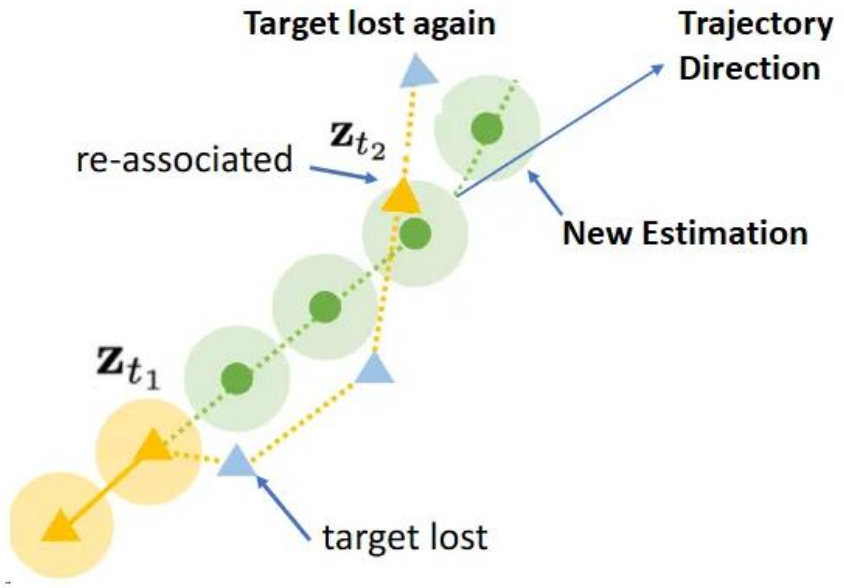


Figure 1. Scenario where target is lost due to large Error uncertainties. The trajectory excludes the most recent track state. Modified from Source.

- OC-SORT Cao et al. 2022 is a novel algorithm based on SORT
- SORT typically fails to maintain track when objects are lost/occluded or when non-linear motion occurs. To overcome this, OC-SORT proposes three solutions:
  - Observation Centric Re-Update (ORU)
  - Observation Centric Momentum (OCM)
  - Observation Centric Recovery (OCR)

DeepSort and OC-Sort

- ORU is activated when a lost track is re-associated
- This simply involves replacing the previous estimations with improved data, thereby reducing the magnitude of the accumulated error
- This 'better data' is referred to as a virtual trajectory.
- The virtual trajectory begins at the track's last observed point in its history (z_{t1}) and concludes with the most recent observation, which is the last association (z_{t2})
- The virtual trajectory is generated in this mannnor:

$$\tilde{\mathbf{z}}_t = \mathbf{z}_{t_1} + \frac{t - t_1}{t_2 - t_1}(\mathbf{z}_{t_2} - \mathbf{z}_{t_1}), t_1 < t < t_2$$

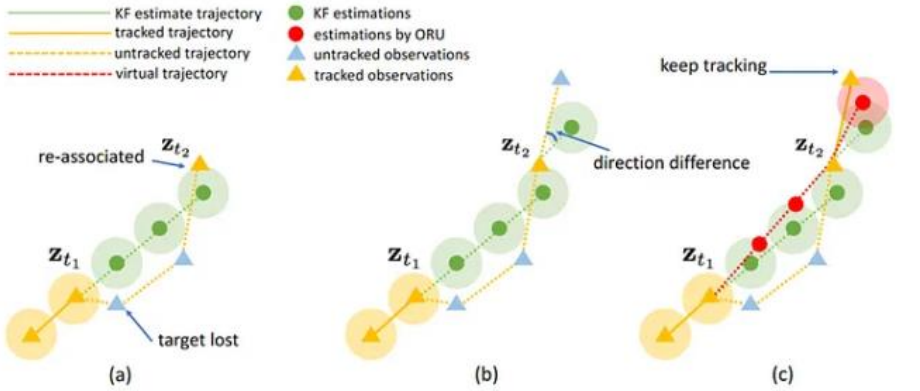
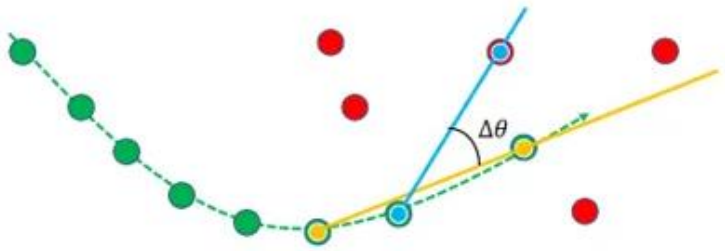


Illustration of how ORU changes the behaviors of SORT after an untracked track is re-associated to an observation. The circle area with shadow indicates the range that an estimate can be associated with observations close enough to it. (a). The track is re-associates with an observation z_{t_2} at the step t_2 after being untracked since the time step t_1. (b). Without ORU, on the next step of re-association, even though the KF state is updated by z_{t_2}, there is still a direction difference between the true object trajectory and the KF estimates. Therefore, the track is unmatched with detections again (in blue). (c). With ORU, we get a more significant change in the state, especially the motion direction by updating velocity. Now, the state estimate (in red) is closer to the state observation and they can be associated again.

DeepSort and OC-Sort



Calculation of motion direction difference in OCM. The green line indicates an existing track and the dots are the observations on it. The red dots are the new observations to be associated. The blue link and the yellow link form the directions of θ^{track} and $\theta^{intention}$ respectively. The included angle is the difference of direction $\Delta\theta$.

- The linear velocity assumption implies that an object has a consistent direction of motion.
- It turns out that the direction of motion can actually be used to more accurately associate tracks.
- In general, we can compute the motion in radians by considering the relative displacement between two bounding boxes

$$\theta = \arctan\left(\frac{v_1 - v_2}{u_1 - u_2}\right)$$

- Where (u_1, v_1) and (u_2, v_2) are two different bounding box positions. To understand how OCM uses motion to associate tracks, we need to understand two angles: the potential track angle θ^{track} and the intention angle $\theta^{intention}$
  - Potential Track angle: θ^{track} links two observations on a track
  - - Previous track observations are linked with every new detection
  - - Each track forms an θ^{track} with every new detection
  - Intention Angle: $\theta^{intention}$ links a past observation with a new observation
  - - A Past track historical observation is linked with a new associated observation
  - - Each track only has one $\theta^{intention}$
- OCM compares the angles of the track (θ^{track}) and intention ($\theta^{intention}$) to get a measure of direction: $\Delta\theta = |\theta^{track} - \theta^{intention}|$, this is called the consistency cost. Similar to the IOU cost matrix C_{iou}, the consistency cost matrix C_v contains all combinations of $\Delta\theta$. We can express the full association cost as:

$$C(\hat{\mathbf{X}}, \mathbf{Z}) = C_{\text{IoU}}(\hat{\mathbf{X}}, \mathbf{Z}) + \lambda C_v(\mathcal{Z}, \mathbf{Z})$$

- Where λ determines how much to weight the consistency cost C_v. In general this provides a more robust association metric that accounts for direction as well as IOU.

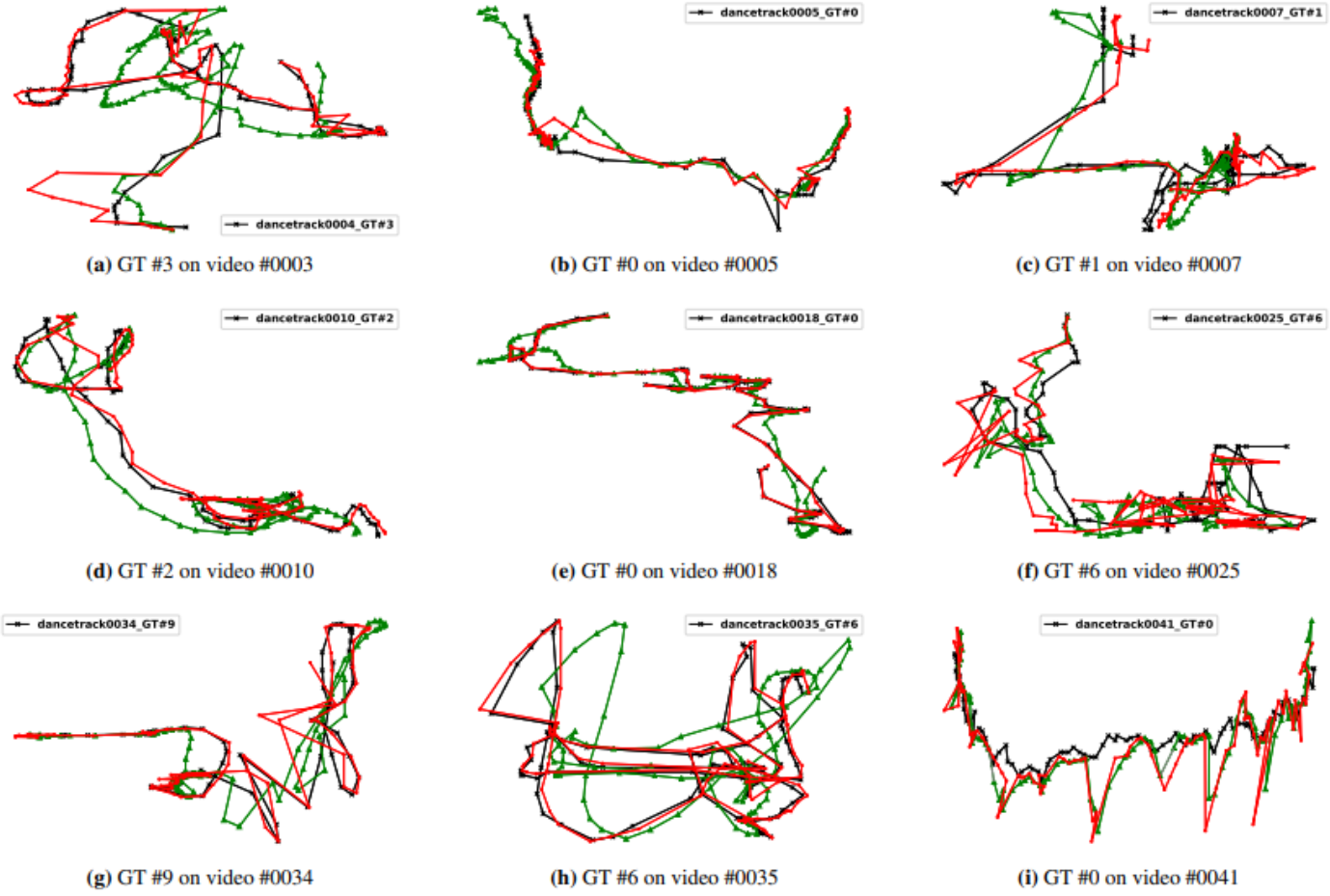
DeepSort and OC-Sort

- In cases where a track has come to a halt or is partially obscured, the motion model may cease to function correctly, resulting in the track not being associated with any detections. Even though we have established techniques to address lost tracks, we still aim to obtain an observation whenever possible.

- OCR is a heuristic technique that can assist in recovering a lost track by utilizing its most recent known observation. It links the last known observation of each unpaired track to all unpaired observations. This constitutes an IOU-based association that occurs after the primary track/detection association.

DeepSort and OC-Sort

- Black cross gt
- Red cross oc-sort
- green cross sort



(a) GT #3 on video #0003 (b) GT #0 on video #0005 (c) GT #1 on video #0007

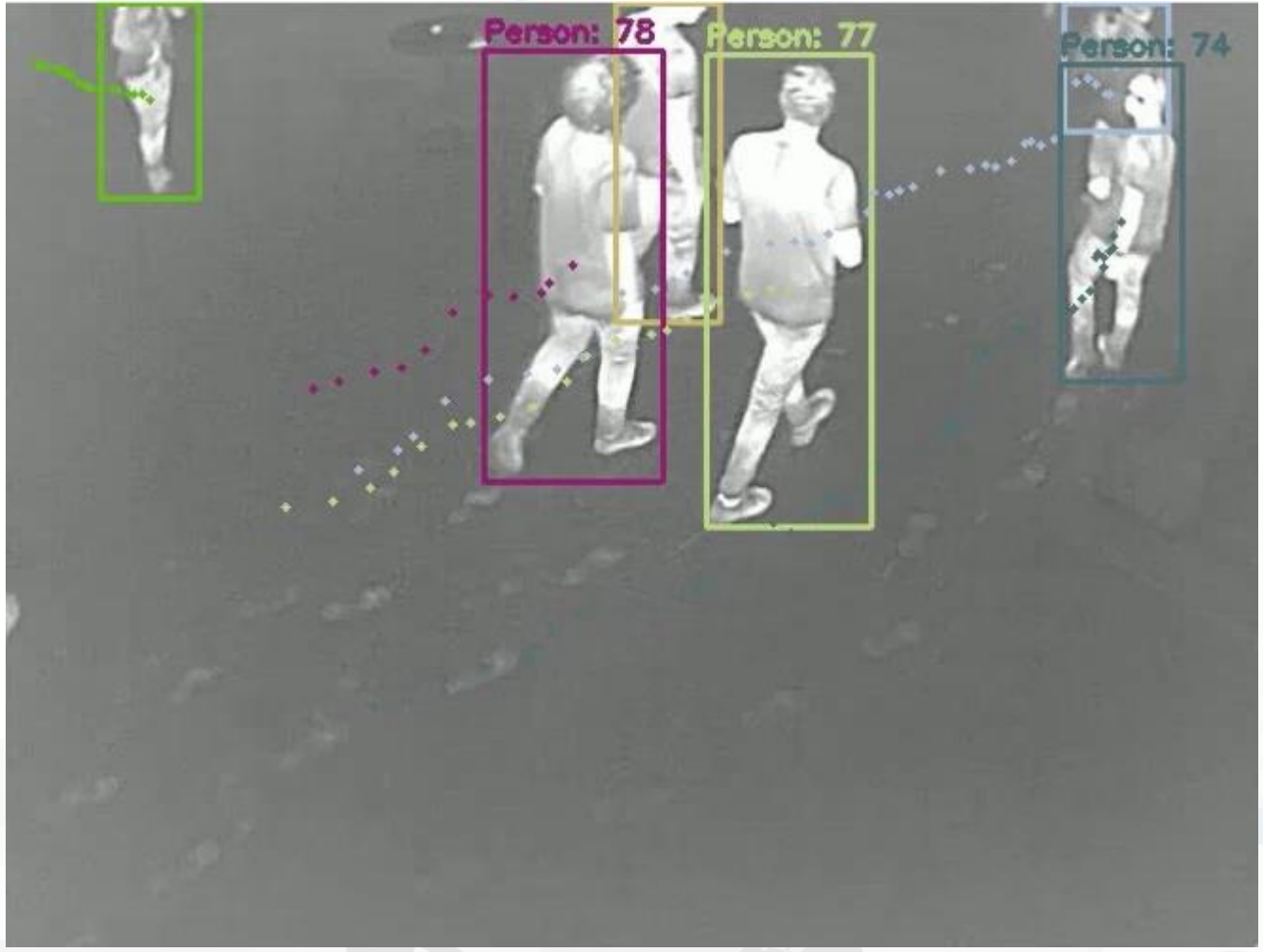
(d) GT #2 on video #0010 (e) GT #0 on video #0018 (f) GT #6 on video #0025

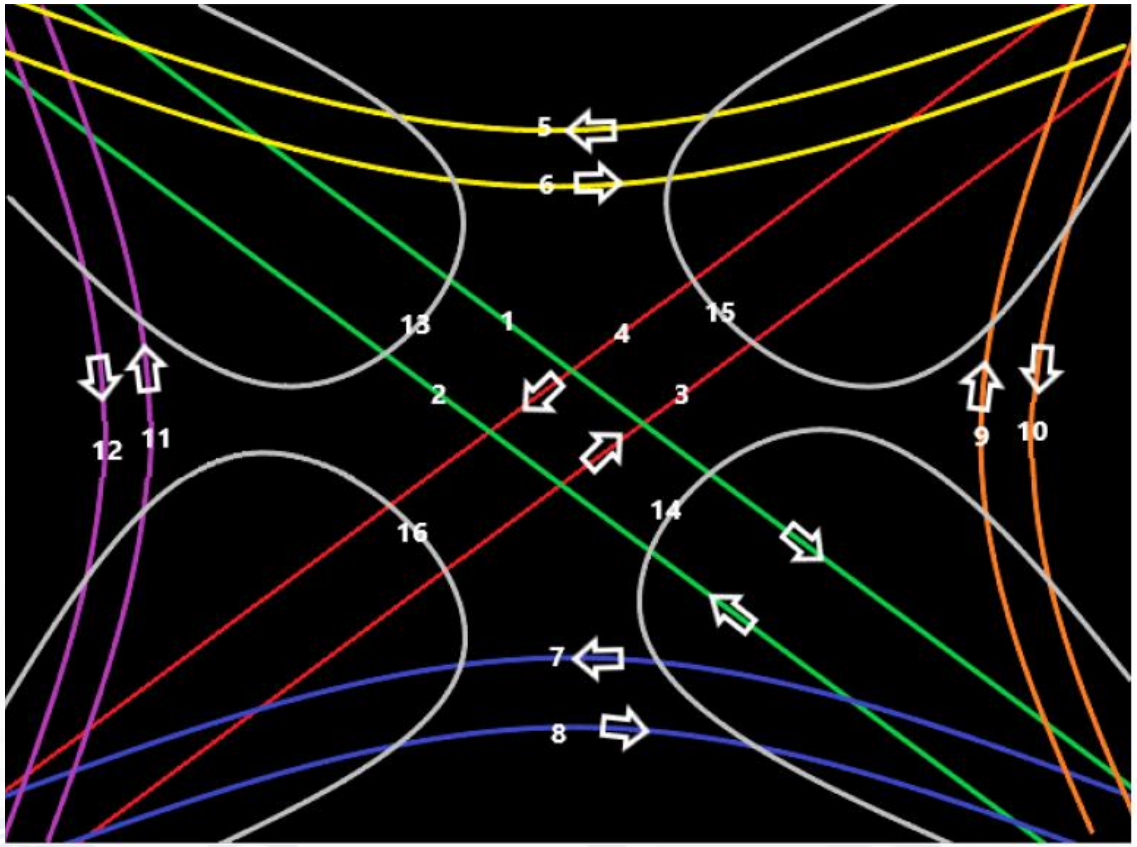
(g) GT #9 on video #0034 (h) GT #6 on video #0035 (i) GT #0 on video #0041

Case study: Barcelona

- We used YOLOv5 and OC-Sort to track and count people using a thermal imaging camera that does not raise privacy concerns (GDPR) and performed well in terms of mAP@50, MOTA and HOTA metrics.
- People were counted in 16 directions and we performed well in terms of weighted F1-score.
- We early publish the paper on

Case study: Barcelona





Let's make a bit of practices

- Please join the colab exercise on yolo
- We will see how training and inference with yolov8 and yolo-nas and tracking object with yolov8 and several trackers
- Follow menti please!

Satellite images

- A satellite image is a visual representation of portions of the earth's surface or other celestial bodies obtained from an orbiting satellite.
- Satellite images can come from a variety of satellites, including Earth observation satellites (such as those of NASA and ESA), weather satellites, communication satellites and even spacecraft sent to other planets or moons.

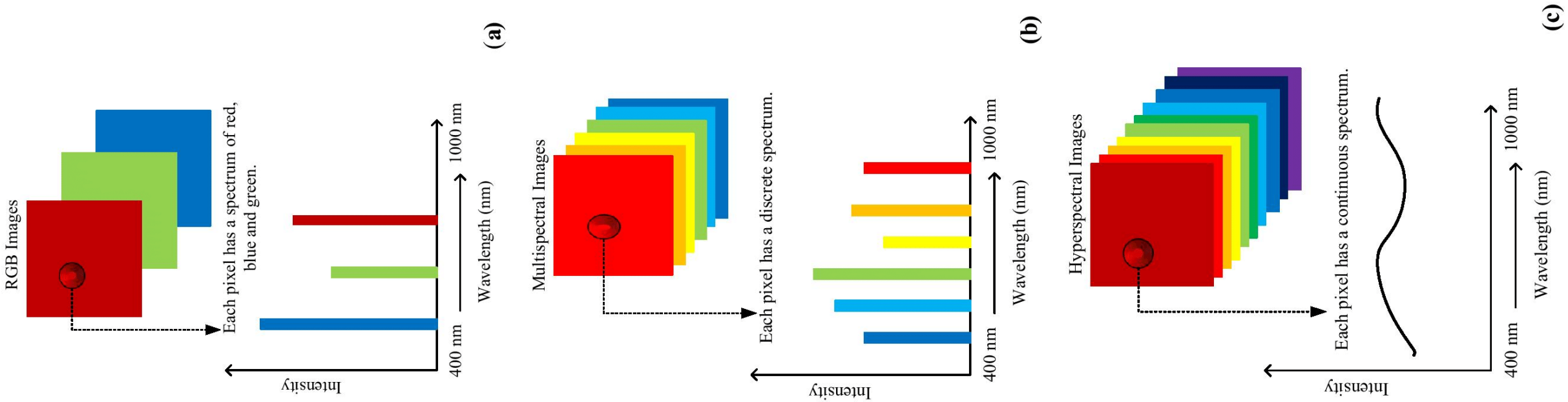
Satellite Imaging Applications

- Using satellite images we can study the development of an area over time
- Study an area e.g. indicate the amount of forest, agricultural, urban areas
- Studying pollution or environmental disasters
- studying coastal erosion
- Support for planning urban develpment

Satellite Imaging

- Satellite images are available in different formats and serve different purposes
- These images can be used to visualise features of the Earth using real colours or to identify a set of features using parts of the spectrum invisible to the human eye.
- Unlike the sample image, most satellite data comes with each channel separated into a separate file for each one. These channels are called bands and comprise a range of the electromagnetic spectrum visible or not visible to the human eye.

Satellite Imaging



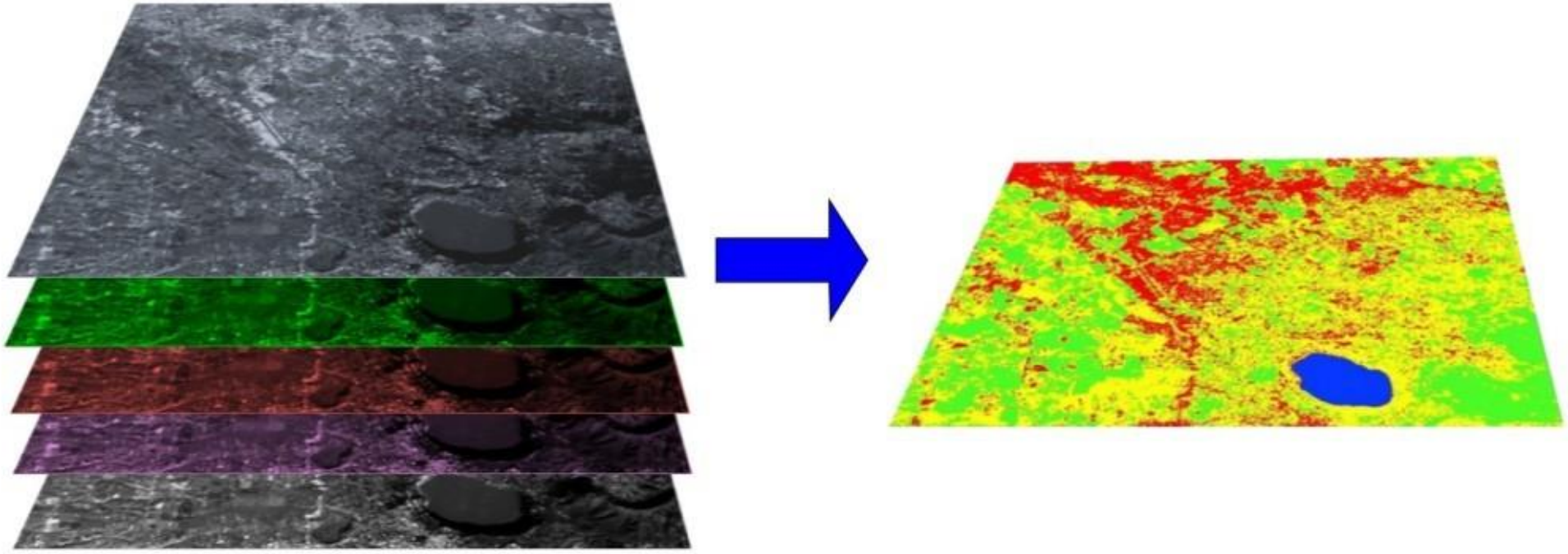
Rgb images -> 3 channel and discrete value

Multi-spectral images -> C channel and discrete value

Hyper-spectral images -> C channel and continuos value

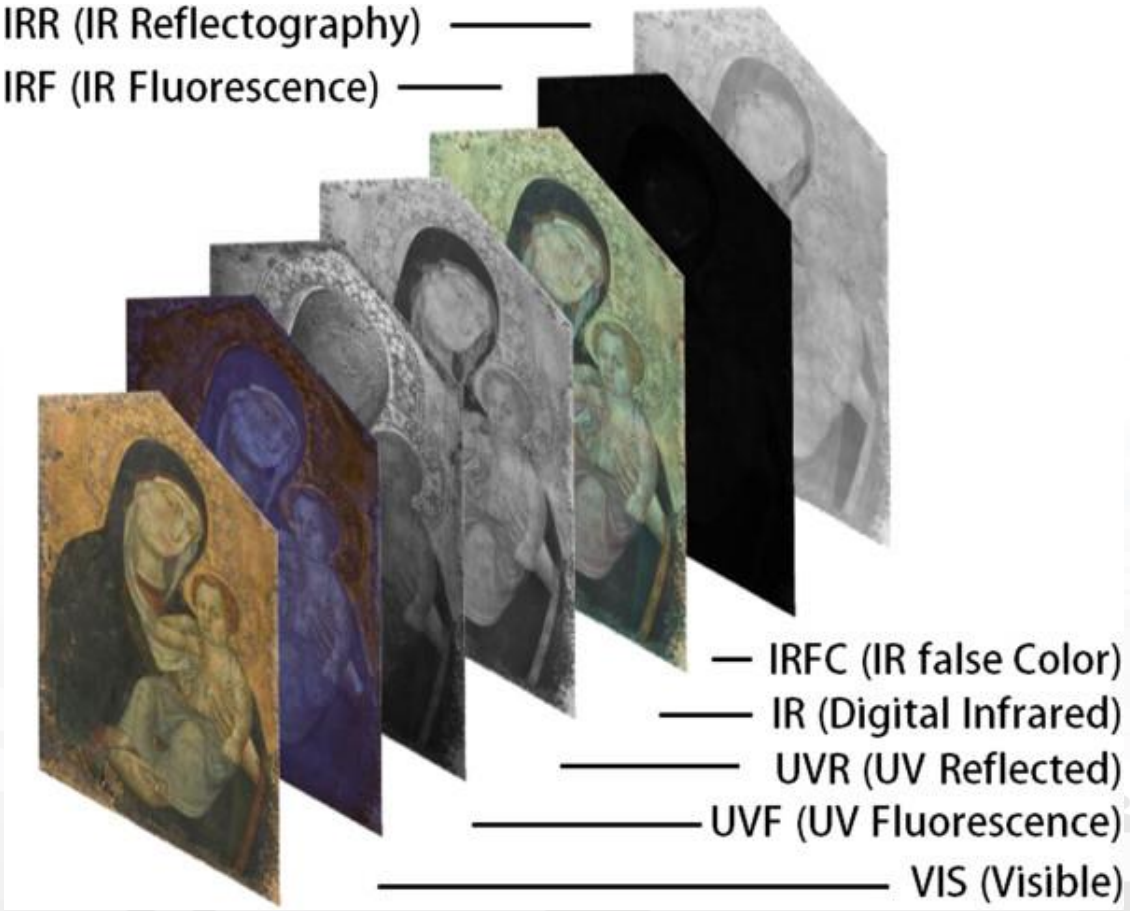
Satellite Imaging

- A satellite image is a multi-spectral image



Satellite Imaging

- Multi-spectral images are more common than you might expect



Satellite Imaging

- Satellites use a combination of sensors and on-board cameras to capture images.
- The sensors may be able to detect different bands of the electromagnetic spectrum, including visible light, infrared and radar.
- For example landast-8 satellite has two sensors: the Operational Land Imager (OLI) and the Thermal Infrared Sensor (TIRS).
- OLI will collect images using nine spectral bands in different wavelengths of visible, near-infrared, and shortwave light to observe a 185 kilometer (115 miles) wide swath of the Earth in 15-30 meters resolution covering wide areas of the Earth's landscape while providing sufficient resolution to distinguish features like urban centers, farms, forests, and other land uses.

- TIRS was added to the satellite mission when it became clear that state water resource managers rely on the highly accurate measurements of Earth's thermal energy obtained by LDCM's predecessors, Landsat 5 and Landsat 7, will track how land and water is being used. With nearly 80 percent of the fresh water in the Western U.S. being used to irrigate crops, TIRS will become an invaluable tool for managing water consumption.

Satellite Imaging

Bands	Wavelength (micrometers)	Resolution (meters)	Common uses
Band 1—Coastal aerosol	0.43 - 0.45	30	Shallow coastal water studies and estimation of the concentration of aerosols in the atmosphere
Band 2—Blue	0.45 - 0.51	30	Visible blue channel, distinguish soil from vegetation
Band 3—Green	0.53 - 0.59	30	Visible green channel
Band 4—Red	0.64 - 0.67	30	Visible red channel
Band 5—Near Infrared (NIR)	0.85 - 0.88	30	Biomass estimation
Band 6—SWIR 1	1.57 - 1.65	30	Soil moisture
Band 7—SWIR 2	2.11 - 2.29	30	Soil moisture
Band 8—Panchromatic	0.50 - 0.68	15	Sharper resolution
Band 9—Cirrus	1.36 - 1.38	30	Detection of cirrus cloud contamination
Band 10—Thermal Infrared (TIRS) 1	10.60 - 11.19	30	Thermal mapping and estimating soil moisture
Band 11—Thermal Infrared (TIRS) 2	11.50 - 12.51	30	Thermal mapping and estimating soil moisture

Satellite Imaging

- Processing satellite images (or other remote sensing data) is a computational challenge for two reasons: usually the images are large (many megabytes or gigabytes) and many images are needed in combination to produce the desired information.

- Opening and processing many large images can consume a lot of computer memory. This condition places a tight limit on what the user can do before running out of memory.

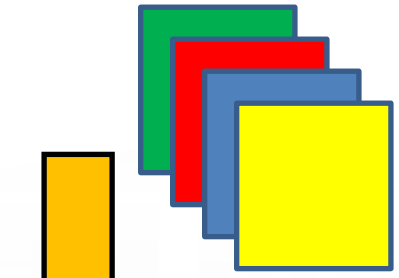
Satellite images

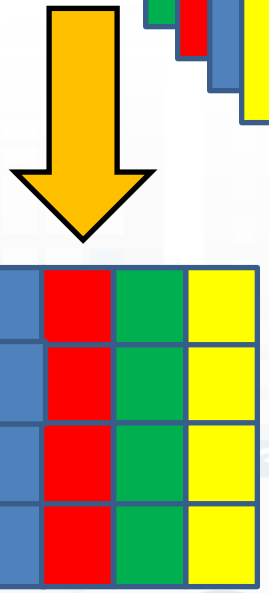
- Computer vision can be used to understand how an area is being used by humans, this task is called land cover
- The idea is to exploit clustering techniques to perform non-supervised pixel-by-pixel classification
- Obviously features are important and the right channels must be selected before proceeding in preparing the data for clustering algorithms

Satellite images

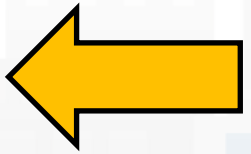
- Let see how to prepare the data

$X \in N^{c,h,w}$





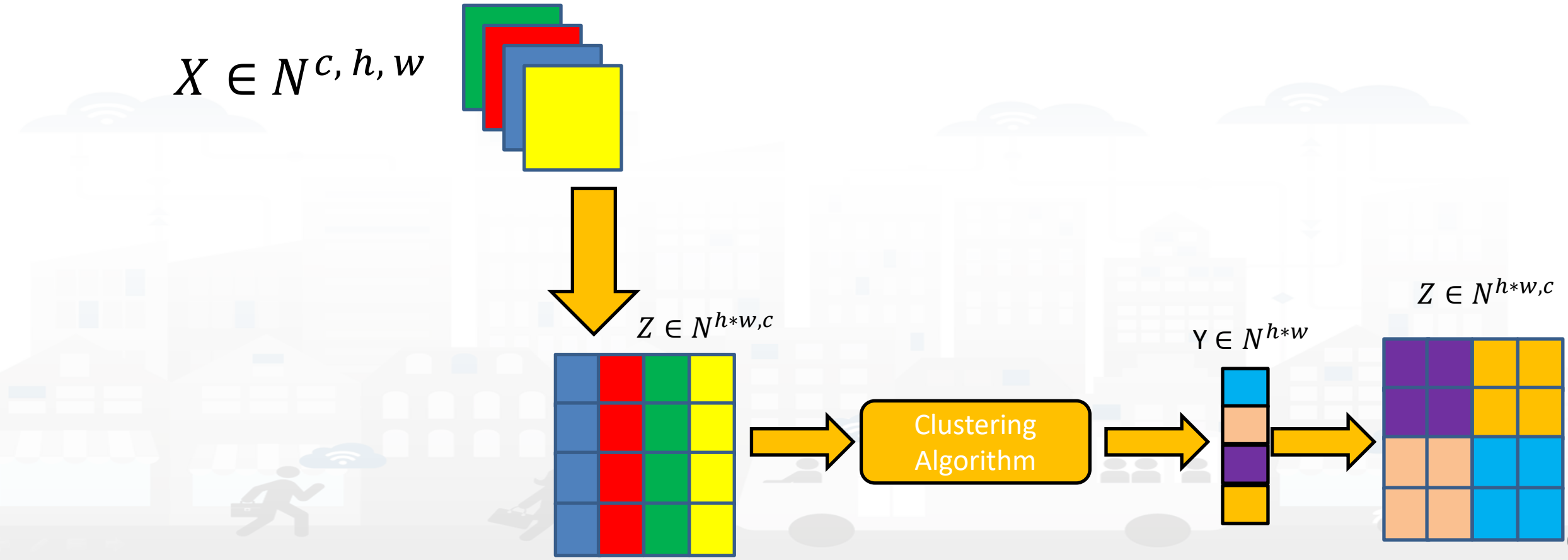
$Z \in N^{h*w,c}$



flatten each channel and then concatenate the channels

Satellite images

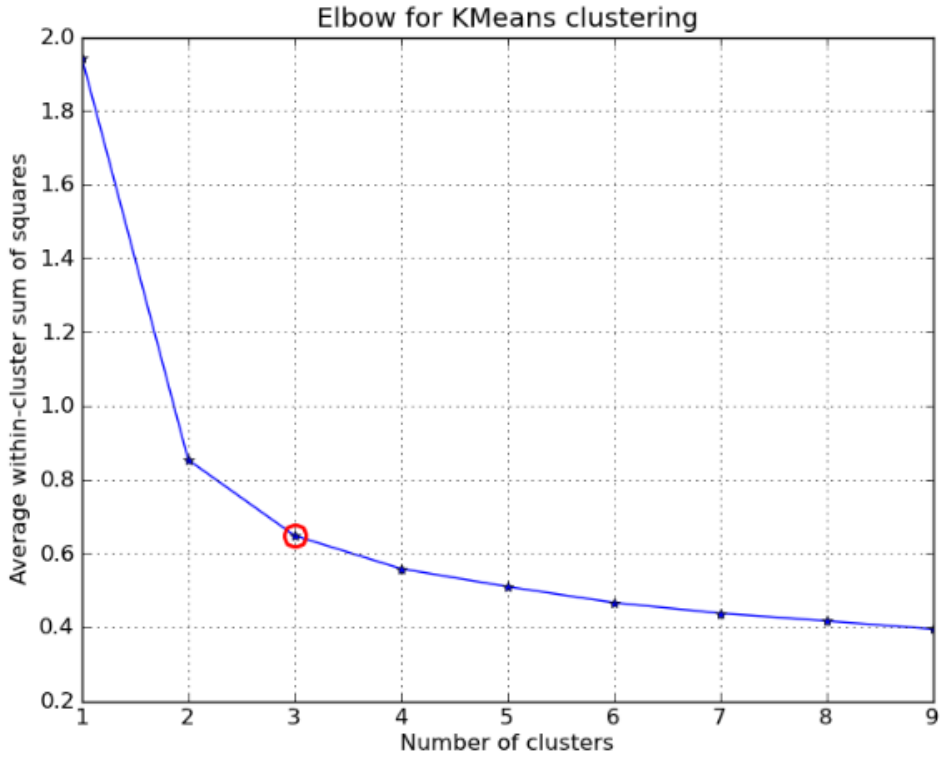
- Let see how to prepare the data



Satellite images

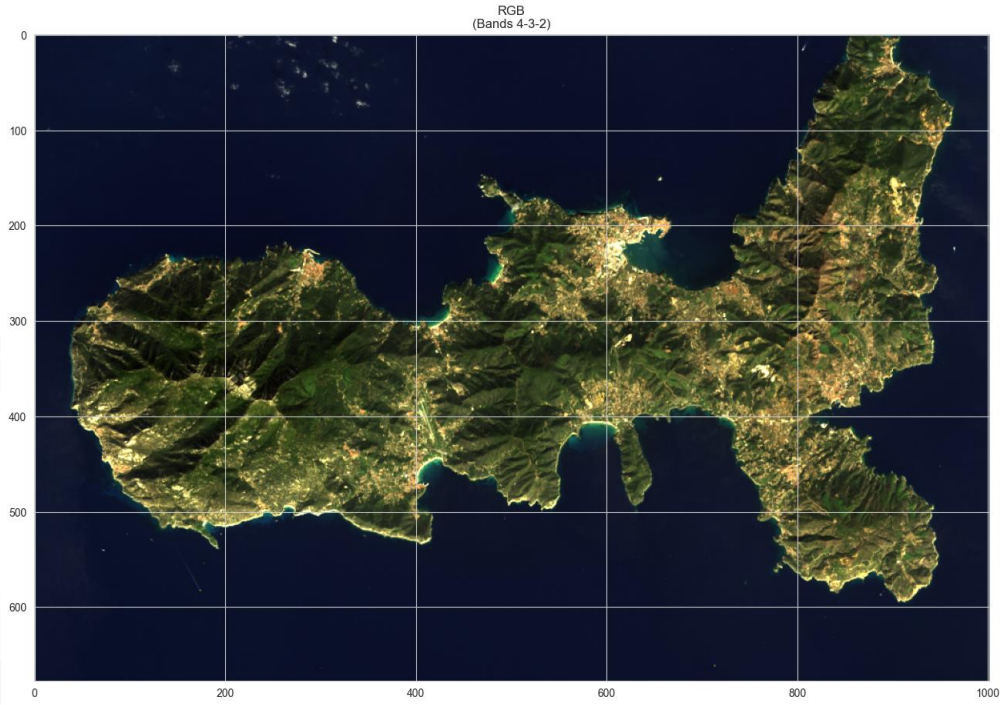
- Now I ask you something about it

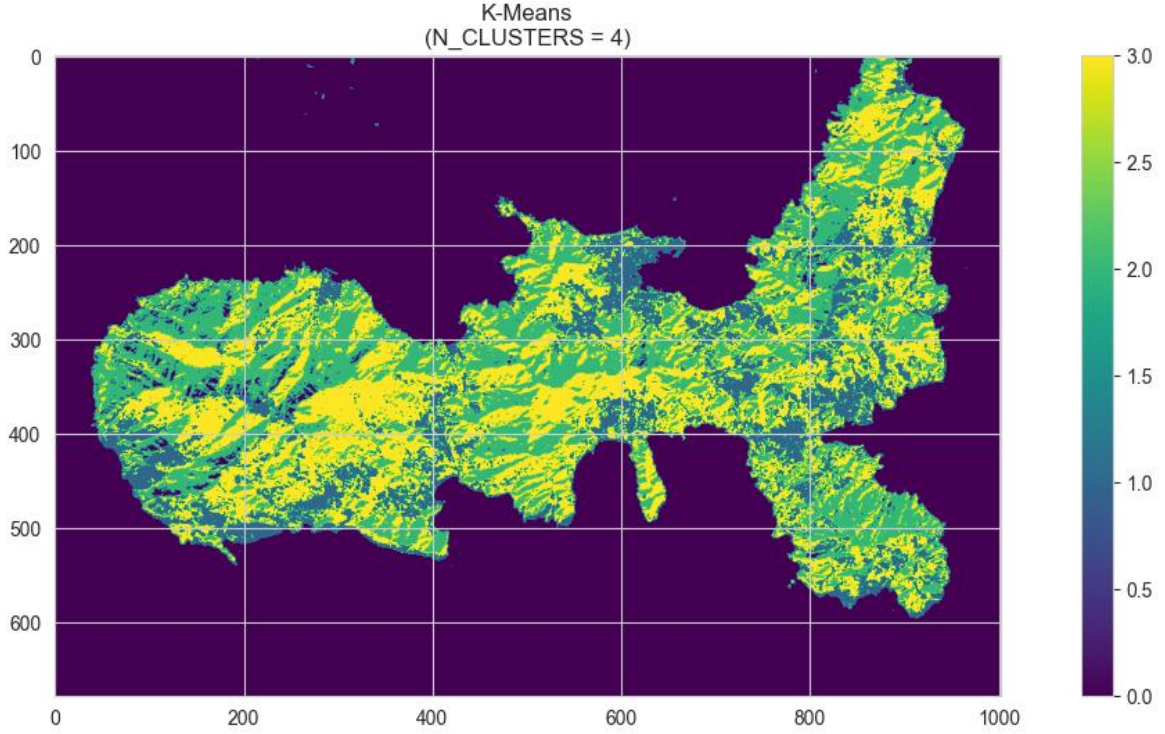
Satellite images



- We can use the k-means algorithm beacuse it is fast and is not memory greedy
- You can choice the number of cluster using one of internal measure:
  - Elbow method
  - Silhouette score
  - Davies–Bouldin index
  - Calinski-Harabasz index
  - Gap statistics
  - Correlation index

Satellite images

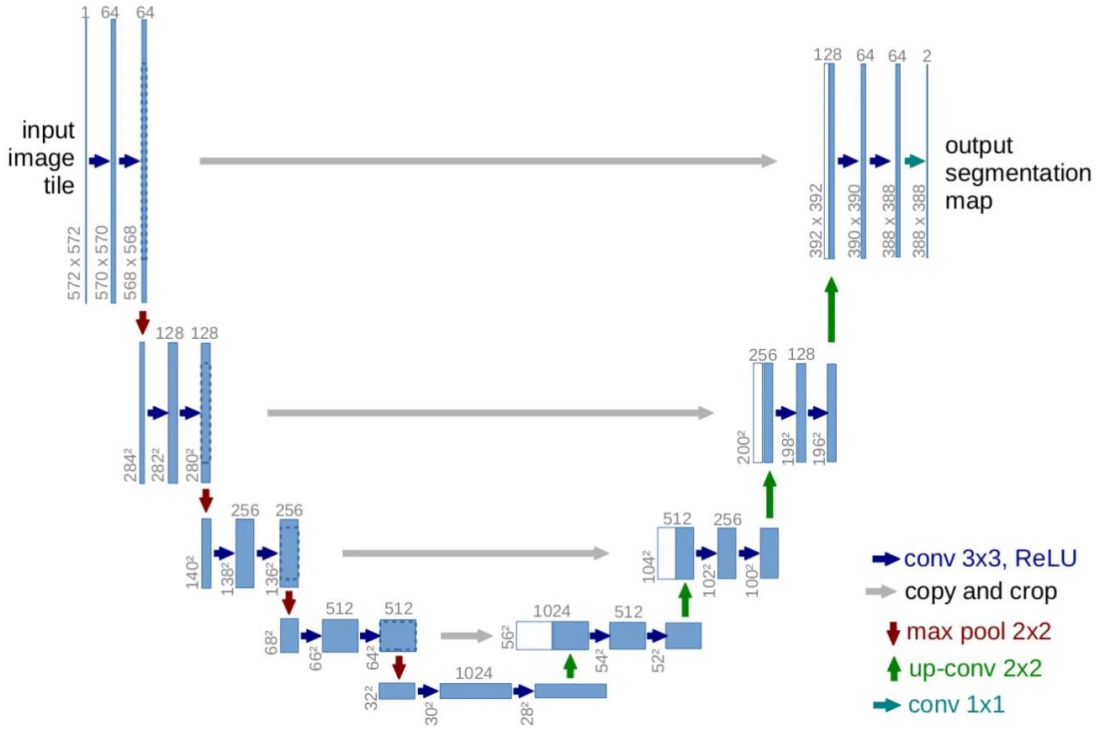




Satellite images

- Classical machine learning techniques are simple and straightforward to use but do not provide accurate results
- Deep learning-based solutions are more powerful and less immediate to use
- Solutions based on deep learning are more powerful and less immediate to use but provide accurate results

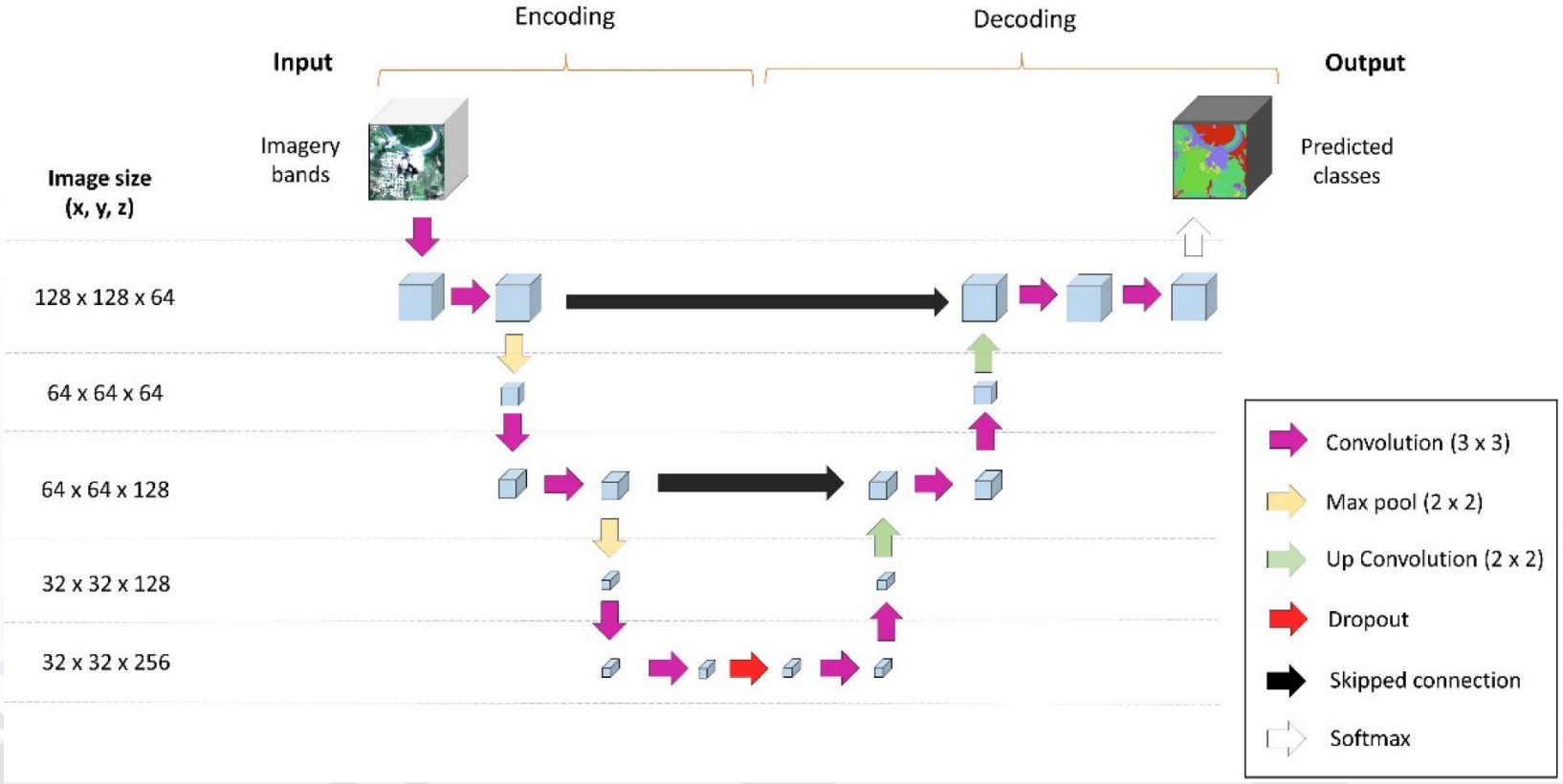
Satellite images



- Using deep learning the U-net networks (Ronneberger et. Al 2015) are the standard for image segmentation
- U-Net is a convolutional neural network that was developed for biomedical image segmentation
- U-net are model encoder-decoder networks:
  - Encoder extract the features
  - Decoder use the skip connection to reconstruct the information by allucination using deconvolution and upsampling layers
- The inputs and outputs of a u-net is an image, so the ground truths are also images

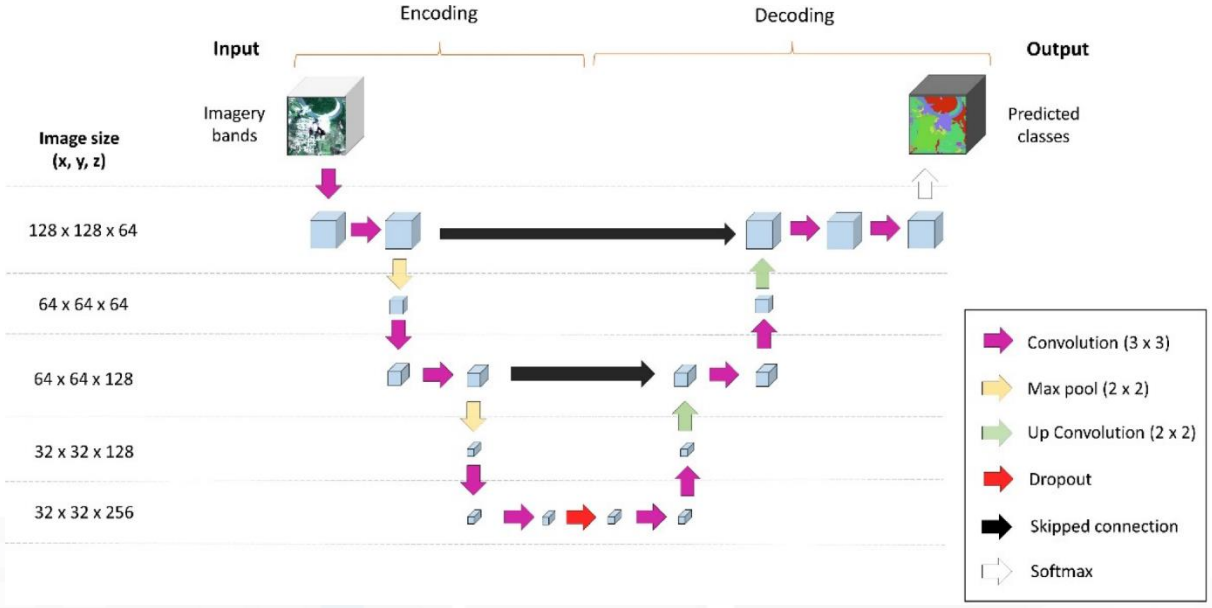
Satellite images

- Solórzano et. al in the 2021 propose a novel approach for the task of cover land using a U-net architecture



Satellite images

- The typical U-net architecture comprises of five hidden layers however, in this study, due to memory restrictions, a simpler version of the U-net was used
- Two outputs are obtained from the U-net: the LULC classification map and the probability map. The typical U-net architecture comprises of five hidden layers
- The loss function is a cross entropy for multiclass and Adam is the optimizer method

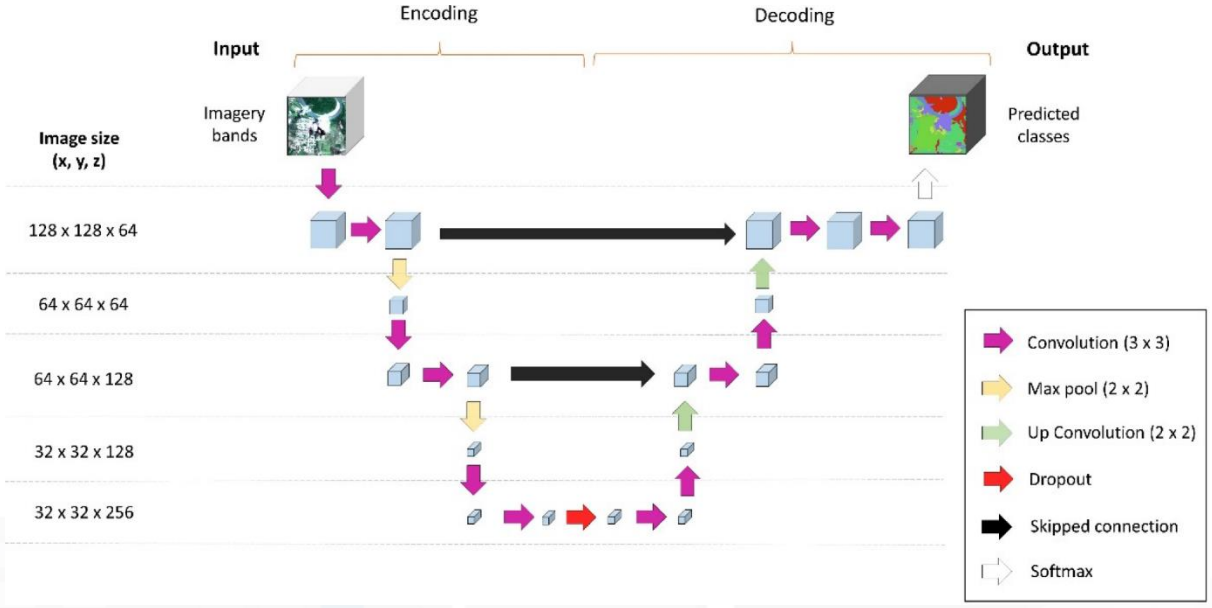


Satellite images

$$avgF_1 = \frac{1}{C}\sum_{c=1}^{C} 2\frac{pr}{p+r}$$

$$p = \frac{TP}{TP + FP}$$

$$r = \frac{TP}{TP + FN}$$



Satellite images

Figure 2. Schematic representation of the complete procedure used in the study.

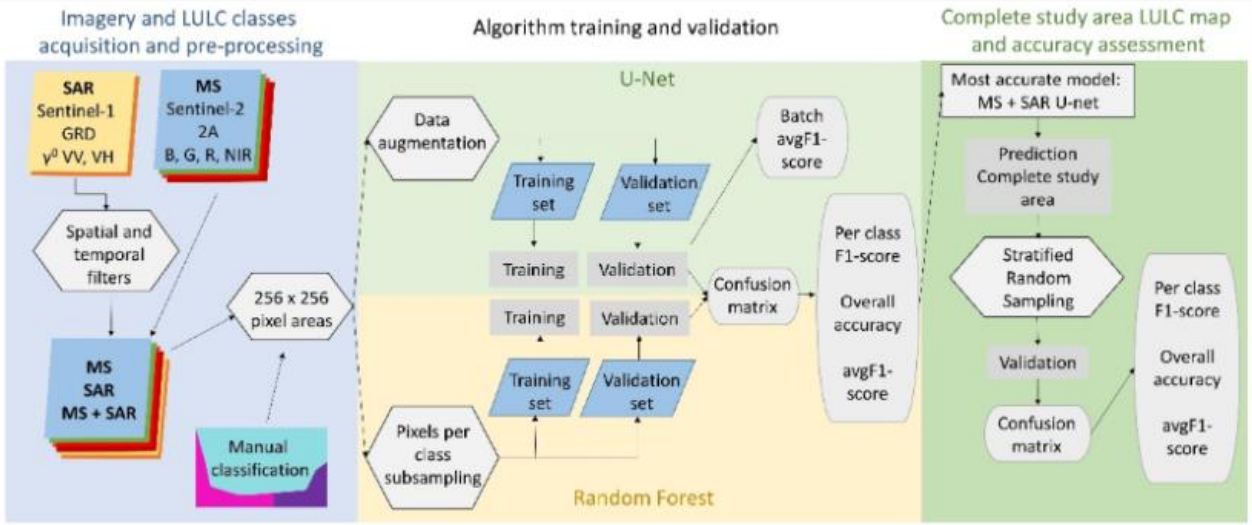
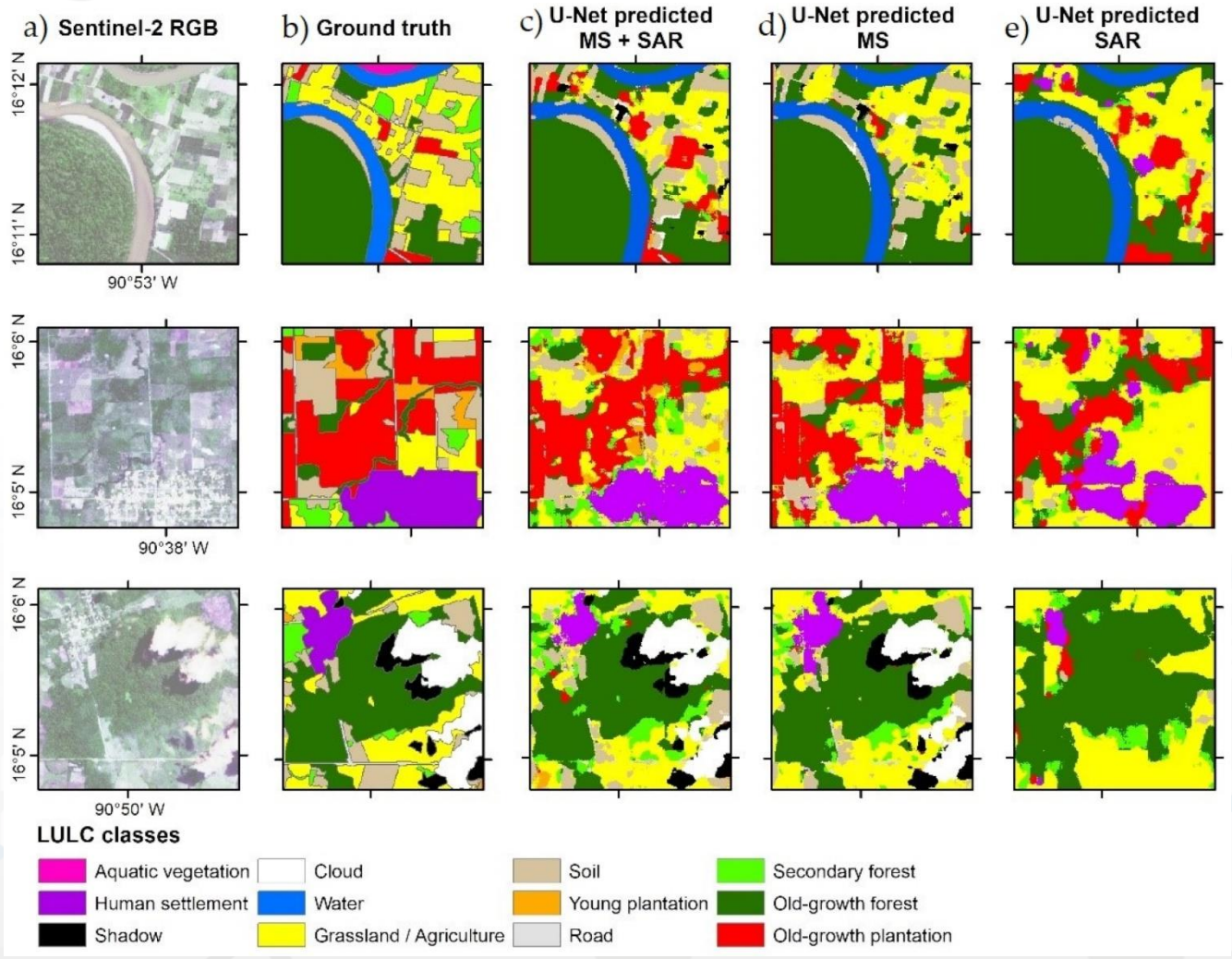


Table 1. F1-score and the difference from the highest score (ΔF1-score) for each class for the MS + SAR, MS, SAR U-net and MS + SAR Random Forests. MS: Multispectral bands, SAR: synthetic aperture radar bands, RF: Random forests.

Class	U-Net						RF	
	MS + SAR		MS		SAR		MS + SAR	
	F1-Score	ΔF1-Score	F1-Score	ΔF1-Score	F1-Score	ΔF1-Score	F1-Score	ΔF1-Score
Aquatic vegetation	0.15	0	0.10	0.05	0	0.15	0.04	0.11
Grassland/Agriculture	0.78	0	0.77	0.01	0.69	0.09	0.63	0.15
Human settlements	0.87	0	0.87	0	0.45	0.42	0.29	0.58
Old-growth forest	0.86	0	0.86	0	0.79	0.07	0.69	0.17
Old-growth plantations	0.62	0	0.54	0.08	0.47	0.15	0.43	0.19
Roads	0.35	0.08	0.43	0	0	0.43	0.23	0.2
Secondary forest	0.45	0	0.34	0.11	0.20	0.25	0.33	0.12
Soil	0.65	0	0.64	0.01	0.30	0.35	0.57	0.08
Water	0.96	0.01	0.97	0	0.94	0.03	0.94	0.03
Young plantations	0.11	0	0.03	0.08	0.01	0.1	0.11	0

Satellite images



Satellite images

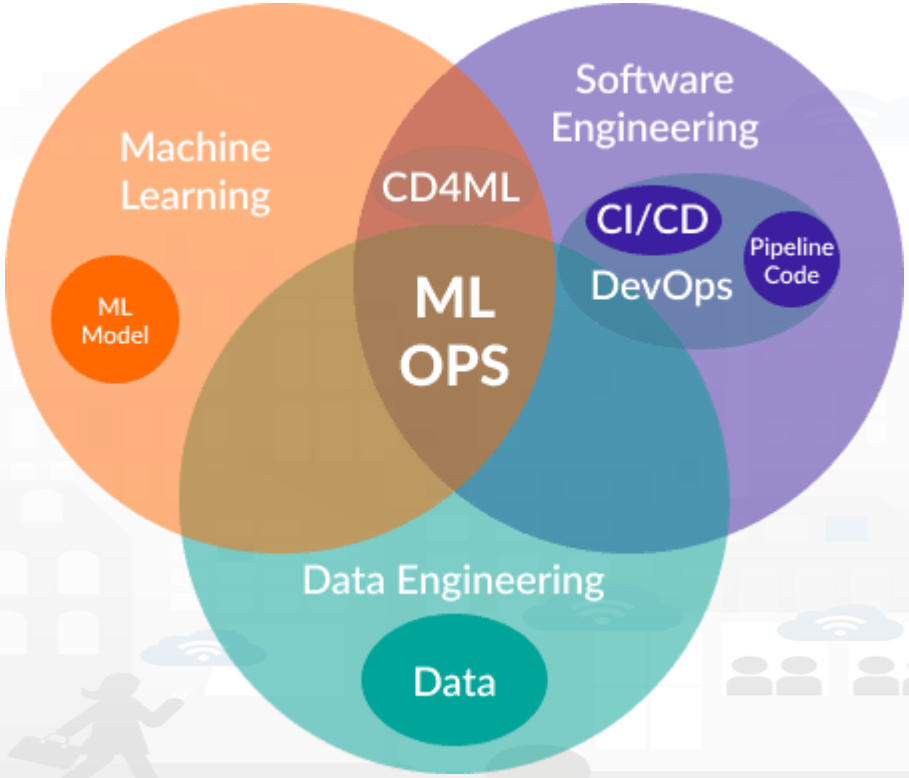
- Please join to menti and follow the instruction for the tutorial on land cover with k-means

Machine learning operations (MLOPS)

- We are at the end of this of the seminar!
- I want to share with you some tips and tricks to better develop some Machine Learning and Deep Learning projects
- the time has come to do things right for your sanity and that of your colleagues
- Join on menti for some questions

Machine learning operations (MLOPS)

Machine Learning Operations (MLOPS) is a practice that combines machine learning (ML) developments with software engineering operations and best practices to enable the efficient implementation, management and maintenance of machine learning models in production.



Machine learning operations (MLOPS)

- As DEVOPS even MLOPS tries to push for a sustainable, collaborative and fast development of a software solution, in this case an MLOPS solution
- There would be so much to talk about, but for reasons of time one must select the material to be discussed, so we will discuss the following tools (see next slide)

Machine learning operations (MLOPS)

COOKIECUTTER

Poetry

DVC

HYDRA

Black

W&B

Flake

Streamlit

OPTUNA

mypy

Cookiecutter

- A good ML project starts from its foundations
- If its structure is unclear and the people working with you go crazy searching for code, tests, docs, plots and data, then you should most probably consider reviewing the project structure
- Lucky for you there is cookiecutter, which automates the structure of a software project on the basis of a template

Cookiecutter

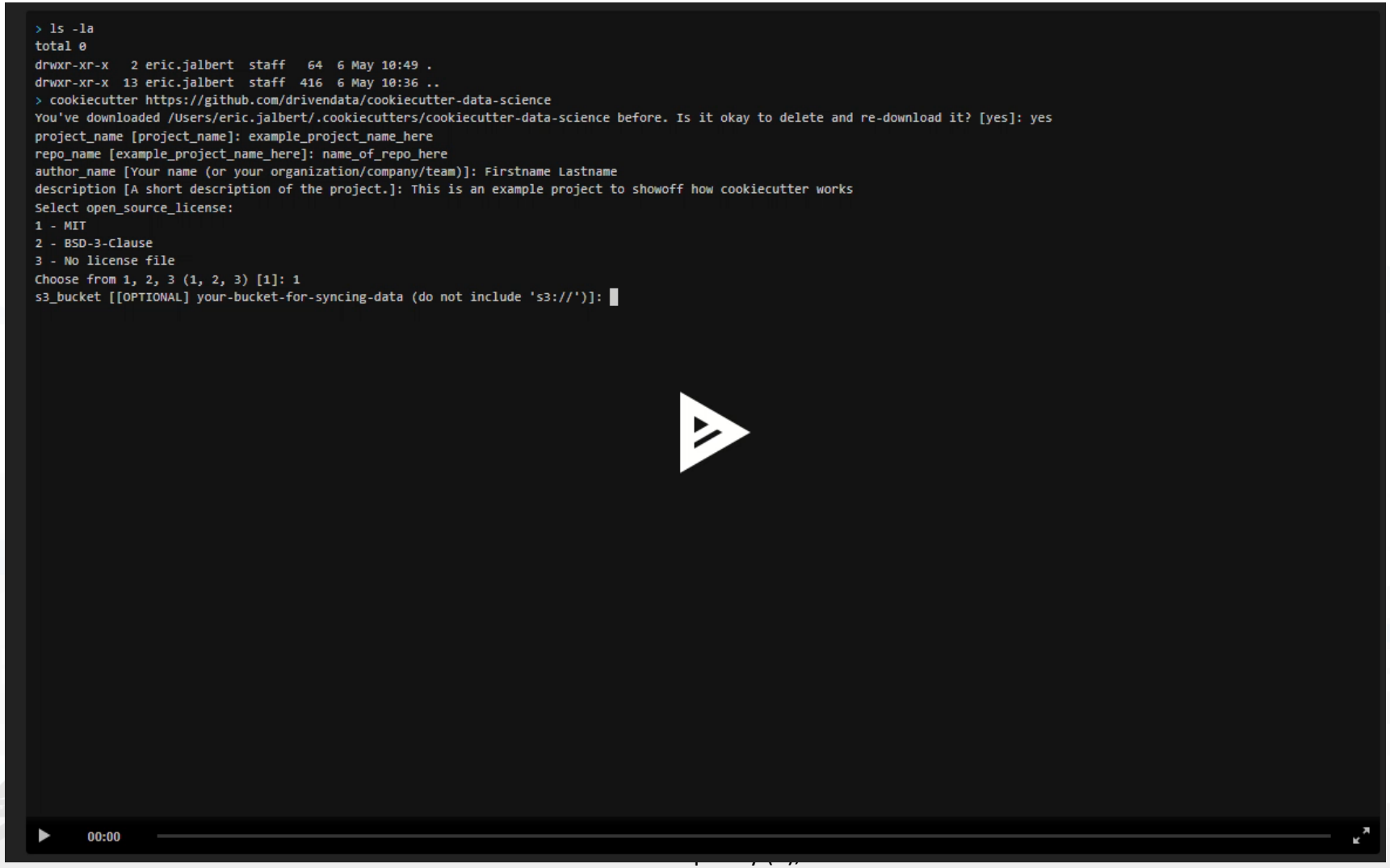
- For install cookiecutter from comand line digits:

With pip: pip3 install cookicutter
With conda:conda install –c conda-forge cookicutter

- To create a new project digits:

cookiecutter -c v1 https://github.com/drivendata/cookiecutter-data-science

Cookiecutter



Poetry

- If you use python I take it for granted that you know its package manager, pip.
- Sometimes it happens that two or more packages are in conflict and you have to handle this do it by hand, or use anaconda, which not only manages environments and installs packages, but also resolves conflicts between them
- Sometimes resolving conflicts by hand is difficult and anaconda takes a long time to resolve dependencies when there are many packages

Poetry

- The solution is to use poetry
- Poetry is a tool that manages the dependencies of your code and also manages their versions
- Poetry also allows you to create readable dependency files (pyproject.toml), so you can quickly know which dependencies are being used in both test and production phases

Poetry

- For create a new project types

```
poetry new poetry-demo
```

- For a pre-existing project types

```
cd pre-existing-project
poetry init
```

- If you want to add dependencies to your project, you can modify the yaml file in the section tool.poetry.dependencies like this

```
[tool.poetry.dependencies]
pandas = "^2.1"
```

- Or simply types on command line

```
poetry add pandas
```

- To delete a dependencie you can delete the dependecie from the section tool.poetry.dependencies
- or simply types on command line

```
poetry remove pandas
```

- Poetry can show your project dependencies as a tree

```
poetry show --tree
```

Hydra

- If we work with Machine Learning then we must also work with data
- Usually we are used to writing the data path in the code (hard coding), this although immediate is not practical from the point of view of code usability, because if the data path changes all references in the project must then be searched and changed
- Hard coding should be avoided with Hydra

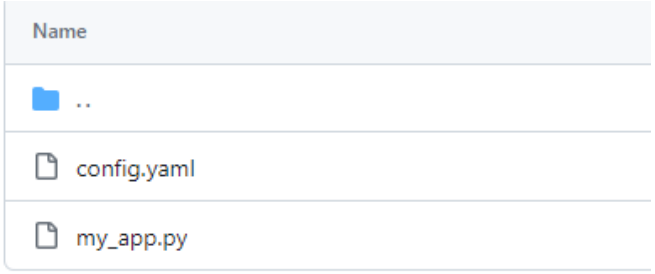
Hydra

- Hydra is a tool that allows you to work with configuration files in python
- A configuration file is a special file where we can define global values that can be useful at multiple points in our project
- In a machine learning project a configuration file can be used for write the path to data, write the hyper-parameters for the algorithms (es learning rate, batch size, k ecc…) and other global variable

Hydra

- Type on command line `pip3 install hydra-core --upgrade`
- In your project, create a folder called config.
- In the folder, create a file called config.yaml
- Omegaconf is a library that is installed together with hydra, it is used to define the hint type for function arguments that must use a configurator object.

Hydra

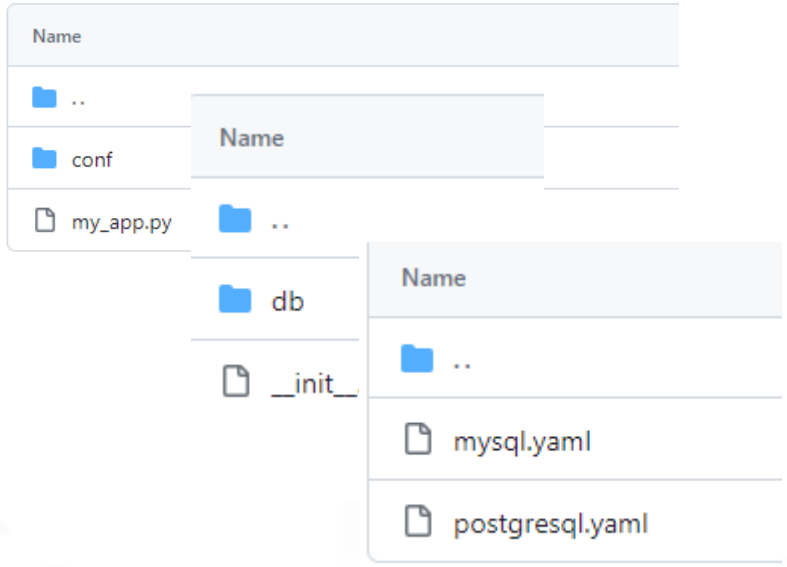



```yaml
db:
  driver: mysql
  user: omry
  password: secret
```

```python
from omegaconf import DictConfig, OmegaConf

import hydra


@hydra.main(version_base=None, config_path=".", config_name="config")
def my_app(cfg: DictConfig) -> None:
    print(OmegaConf.to_yaml(cfg))


if __name__ == "__main__":
    my_app()
```

- Omegaconf is a library that is installed together with hydra, it is used to define the hint type for function arguments that must use a configurator object
- @hydra.main(version_base = None, config_path=".", config_name="config") is a decorator method that allow the function my_app to access to the content of configuration file

Hydra



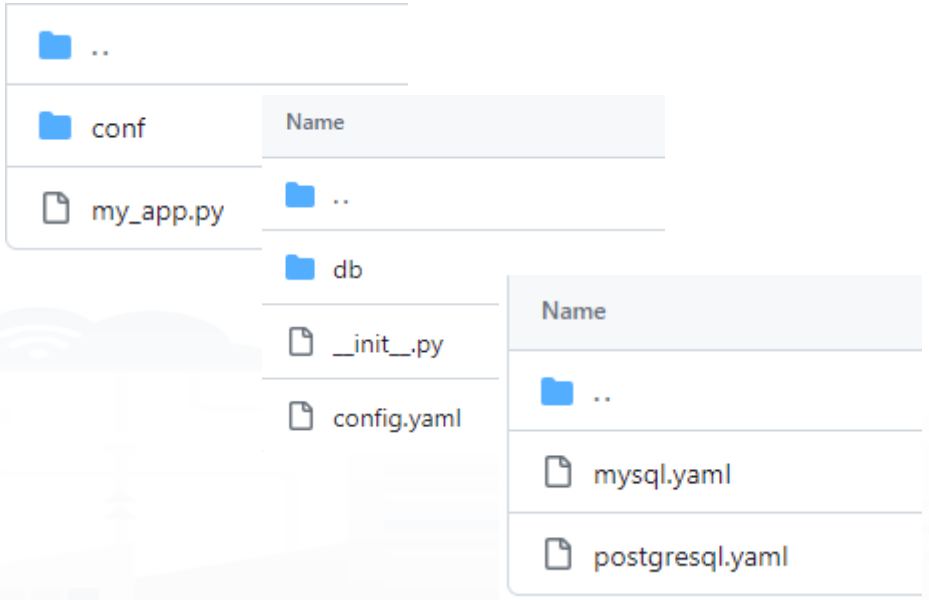

```yaml
db:
  driver: mysql
  user: omry
  password: secret
```

```yaml
driver: postgresql
user: postgres_user
password: drowssap
timeout: 10
```

```python
from omegaconf import DictConfig, OmegaConf

import hydra


@hydra.main(version_base=None, config_path=".", config_name="config")
def my_app(cfg: DictConfig) -> None:
    print(OmegaConf.to_yaml(cfg))


if __name__ == "__main__":
    my_app()
```

- it is possible to have more than one configuration file if there is a need, the files must be grouped in a folder and the folder name must be indicated in the decorator method
- To select the configuration file during the execution type python my_app.py +db=postgresql o type python my_app.py +db=mysql

Hydra




```python
from omegaconf import DictConfig, OmegaConf

import hydra


@hydra.main(version_base=None, config_path="conf", config_name="config")
def my_app(cfg: DictConfig) -> None:
    print(OmegaConf.to_yaml(cfg))


if __name__ == "__main__":
    my_app()
```

```yaml
defaults:
  - db: mysql
```

```yaml
driver: mysql
user: omry
pass: secret
```

```yaml
driver: postgresql
user: postgres_user
pass: drowssap
timeout: 10
```

- You can select a default configuration file that tells Hydra how to compose the final configuration file
- With a default list you don't need to type python my_app.py +db=mysql you just type python my_app.py for mysql
- Postgresql is still available typing python my_app.py +db=postgresql

Hydra

- When using hydra, the latter creates an output folder in the format DD-MM-YYY and within it, depending on how many executions it creates several sub-folders with the name HH:mm:ss
- Basically, hydra creates a log of our executions
- If you try to do a path print within the code where hydra is used with os or pathlib, the latter instead of returning the path to the script being executed will print the path to the output folder: example

Hydra

- *### main.py*

import hydra
from omegaconf import DictConfig

@ hydra.main(config_path = "config", config_name = "config")
def **func** (cfg: DictConfig):
    working_dir = os.getcwd()
    print (f"La corrente la directory di lavoro è {working_dir}")

    # Per accedere agli elementi della configurazione
    print (f"La dimensione del batch è {cfg.batch_size}")
    print (f"Il tasso di apprendimento è {cfg['lr']}")

if __name__ == "__main__":
    func()

- When you run src/main.py, hydra moves this file to src/outputs/2021-03-13/16-22-21/main.pye then runs it.
- The current working directory is src/outputs/2021-03-13/16-22-21 when you run python main.py

Hydra

- Another useful feature of Hydra is the possibility of creating log files.
- It is also possible to indicate the log level in order to identify and understand the type of printing if it refers to debug printing, for example...
- By installing a plug-in, it is possible to associate colours to the log levels to increase readability.

Hydra

- For installing the color plug-in types: pip install hydra_colorlog –upgrade
- Override hydra/job_logging and hydra/hydra_logging in your config like this:
  - defaults:
    - override hydra/job_logging: colorlog
    - override hydra/hydra_logging: colorlog

```
(hydra36) [17:01] omry-mbp:omry@~/dev/hydra/plugins/hydra_colorlog $ python example/my_app.py  -m
[2019-10-22 17:01:43,751][HYDRA] Launching 1 jobs locally
[2019-10-22 17:01:43,751][HYDRA] Sweep output dir : multirun/2019-10-22/17-01-43
[2019-10-22 17:01:43,751][HYDRA]        #0 :
[2019-10-22 17:01:43,816][__main__][INFO] - Info level message
[2019-10-22 17:01:43,816][__main__][WARNING] - Warning level message
[2019-10-22 17:01:43,816][__main__][ERROR] - Error level message
[2019-10-22 17:01:43,816][__main__][CRITICAL] - Critical level message
```

```python
import logging

from omegaconf import DictConfig

import hydra

# A logger for this file
log = logging.getLogger(__name__)


@hydra.main(version_base=None)
def my_app(_cfg: DictConfig) -> None:
    log.info("Info level message")
    log.debug("Debug level message")


if __name__ == "__main__":
    my_app()
```

Black

- Good python programmers can be recognised from bad ones because they respect the PEP8 formatting style
- If you follow PEP8 you do yourself and your colleagues who have to read or work on your code a favour
- Black is a python library that automatically formats code contained in script files

Black

main.py 01.ipynb example_01.py

```
1  def a_long_functionn_that_will_break_the_pep8_style_guide(param1:int, param2:int, param3:int, param4:int, param5:int, param6:int, param7:int, param8:int):
2      pass
```

Black

```
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$ black --diff src/example_01.py
--- src/example_01.py   2023-11-07 10:45:38.564368+00:00
+++ src/example_01.py   2023-11-07 10:46:50.177603+00:00
@@ -1,2 +1,11 @@
-def a_long_functionn_that_will_break_the_pep8_style_guide(param1:int, param2:int, param3:int, param4:int, param5:int, param6:int, param7:int, param8:int):
-    pass
\ No newline at end of file
+def a_long_functionn_that_will_break_the_pep8_style_guide(
+    param1: int,
+    param2: int,
+    param3: int,
+    param4: int,
+    param5: int,
+    param6: int,
+    param7: int,
+    param8: int,
+):
+    pass
would reformat src/example_01.py
```

Black


Terminal
Local
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$ black src/example_01.py
reformatted src/example_01.py
All done! ✨ 🍰 ✨
1 file reformatted.
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$
torchGeo > src > example_01.py

Black

main.py 01.ipynb example_01.py

```
1  def a_long_functionn_that_will_break_the_pep8_style_guide(
2      param1: int,
3      param2: int,
4      param3: int,
5      param4: int,
6      param5: int,
7      param6: int,
8      param7: int,
9      param8: int,
10 ):
11     pass
12
```

Flake-8

- We all make mistakes and don't realise we have made them
- Luckily for us, there's Flake-8 to point them out
- Flake-8 will report unused variables, formatting errors and violations of the PEP8 standard
- For installing Flake-8 pip3 install flake8

Flake-8

main.py | 01.ipynb | example_01.py | example_02.py

```
1  def a_function_that_will_be_mad_flake8(param1: int,param2: int,param3: int
2                                         ,param4: int,param5: int,param6: int,param7: int,param8: int):
3      print("Hello I'm a function that will be mad by flake8")
4      print(param1,param2,param3,param4,param5,param6,param7,param8)
```

Flake-8

```
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$ flake8 src/example_02.py
src/example_02.py:1:51: E231 missing whitespace after ','
src/example_02.py:1:63: E231 missing whitespace after ','
src/example_02.py:1:75: E203 whitespace before ','
src/example_02.py:2:40: E231 missing whitespace after ','
src/example_02.py:2:52: E231 missing whitespace after ','
src/example_02.py:2:64: E231 missing whitespace after ','
src/example_02.py:2:76: E231 missing whitespace after ','
src/example_02.py:2:80: E501 line too long (101 > 79 characters)
src/example_02.py:2:88: E231 missing whitespace after ','
src/example_02.py:4:17: E231 missing whitespace after ','
src/example_02.py:4:24: E231 missing whitespace after ','
src/example_02.py:4:31: E231 missing whitespace after ','
src/example_02.py:4:38: E231 missing whitespace after ','
src/example_02.py:4:45: E231 missing whitespace after ','
src/example_02.py:4:52: E231 missing whitespace after ','
src/example_02.py:4:59: E231 missing whitespace after ','
src/example_02.py:4:67: W292 no newline at end of file
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$
```

Flake-8

```
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$ clear
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$ flake8 src/example_02.py
src/example_02.py:1:51: E231 missing whitespace after ','
src/example_02.py:1:63: E231 missing whitespace after ','
src/example_02.py:1:75: E203 whitespace before ','
src/example_02.py:2:40: E231 missing whitespace after ','
src/example_02.py:2:52: E231 missing whitespace after ','
src/example_02.py:2:64: E231 missing whitespace after ','
src/example_02.py:2:76: E231 missing whitespace after ','
src/example_02.py:2:80: E501 line too long (101 > 79 characters)
src/example_02.py:2:88: E231 missing whitespace after ','
src/example_02.py:4:17: E231 missing whitespace after ','
src/example_02.py:4:24: E231 missing whitespace after ','
src/example_02.py:4:31: E231 missing whitespace after ','
src/example_02.py:4:38: E231 missing whitespace after ','
src/example_02.py:4:45: E231 missing whitespace after ','
src/example_02.py:4:52: E231 missing whitespace after ','
src/example_02.py:4:59: E231 missing whitespace after ','
src/example_02.py:4:67: W292 no newline at end of file
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$ black src/example_02.py
reformatted src/example_02.py

All done! ✨ 🍰 ✨
1 file reformatted.
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$ flake8 src/example_02.py
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$
```

Flake-8

main.py 01.ipynb example_01.py example_02.py

```python
def a_function_that_will_be_mad_flake8(
    param1: int,
    param2: int,
    param3: int,
    param4: int,
    param5: int,
    param6: int,
    param7: int,
    param8: int,
):
    print("Hello I'm a function that will be mad by flake8")
    print(param1, param2, param3, param4, param5, param6, param7, param8)

```

Flake-8

main.py | 01.ipynb | example_01.py | example_02.py

```
1  def a_function_that_will_be_mad_flake8(param1: int,param2: int,param3: int
2                                         ,param4: int,param5: int,param6: int,param7: int,param8: int):
3      print("Hello I'm a function that will be mad by flake8")
4      print(param1,param2,param3,param4,param5,param6,param7,param8)
```

I-sort

- When writing python code, it is good practice to write all imports at the beginning of the script file
- It sometimes happens that you write an import anywhere in the script file
- This, although python allows it, is best avoided
- I-sort sorts all imports back to the beginning of the script
- Types on command line `pip3 install isort`

I-sort

main.py 01.ipynb example_01.py example_02.py example_03.py

```
1  import pandas as pd
2  import numpy as np
3  import seaborn as sns
4  from sklearn.model_selection import train_test_split
5  import matplotlib.pyplot as plt
6  from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score, log_loss, confusion_matrix
7  from sklearn.tree import DecisionTreeClassifier, export_graphviz
8  import itertools
9  from sklearn import tree
10 from sklearn.preprocessing import StandardScaler
11
```

I-sort

```
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$ isort src/example_03.py --diff
--- /home/matteo/PycharmProjects/torchGeo/src/example_03.py:before      2023-11-07 14:40:20.671603
+++ /home/matteo/PycharmProjects/torchGeo/src/example_03.py:after       2023-11-07 14:42:25.231273
@@ -1,11 +1,12 @@
+import itertools
+
+import matplotlib.pyplot as plt
+import numpy as np
 import pandas as pd
-import numpy as np
 import seaborn as sns
+from sklearn import tree
+from sklearn.metrics import (accuracy_score, confusion_matrix, f1_score,
+                             log_loss, precision_score, recall_score)
 from sklearn.model_selection import train_test_split
-import matplotlib.pyplot as plt
-from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score, log_loss, confusion_matrix
+from sklearn.preprocessing import StandardScaler
 from sklearn.tree import DecisionTreeClassifier, export_graphviz
-import itertools
-from sklearn import tree
-from sklearn.preprocessing import StandardScaler
```

I-sort

main.py 01.ipynb example_01.py example_02.py example_03.py

```
1  import itertools
2
3  import matplotlib.pyplot as plt
4  import numpy as np
5  import pandas as pd
6  import seaborn as sns
7  from sklearn import tree
8  from sklearn.metrics import (accuracy_score, confusion_matrix, f1_score,
9                               log_loss, precision_score, recall_score)
10 from sklearn.model_selection import train_test_split
11 from sklearn.preprocessing import StandardScaler
12 from sklearn.tree import DecisionTreeClassifier, export_graphviz
```

My[py]

- Unlike languages such as Java or C++ which have a static type-checker, Python has a dynamic type-checker
- Having a dynamic type-checker allows Python to be more flexible in evaluating expressions
- However, this flexibility can backfire when analysing code and writing functions or methods
- Mypy counters the type of the expression as if it were a static type-checker language
- Types on command line pip3 install mypy

My[py]

main.py | 01.ipynb | example_01.py | example_04.py | example_02.py | example_03.py

```
1
   2 usages
2  def greeting(name: str):
3      return 'Hello' + name
4
5
6  if __name__ == '__main__':
7      print(greeting('World'))
8      print(greeting(123))
9      number = input('Insert a number: ')
10     print(number+1)
```

My[py]

```
Terminal    Local  ×
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$ mypy src/example_04.py
src/example_04.py:8: error: Argument 1 to "greeting" has incompatible type "int"; expected "str"  [arg-type]
src/example_04.py:10: error: Unsupported operand types for + ("str" and "int")  [operator]
Found 2 errors in 1 file (checked 1 source file)
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$
```

My[py]

main.py | 01.ipynb | example_01.py | example_04.py | example_02.py | example_03.py

```
1
    2 usages
2   def greeting(name: str):
3       return 'Hello' + name
4
    2 usages
5   def greeting_new(name):
6       return 'Hello' + name
7
8   if __name__ == '__main__':
9       print(greeting('World'))
10      print(greeting(123))
11      number = input('Insert a number: ')
12      print(greeting_new('World'))
13      print(greeting_new(123))
14      print(number+1)
```

My[py]

```
Terminal   Local ×
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$ mypy src/example_04.py
src/example_04.py:8: error: Argument 1 to "greeting" has incompatible type "int"; expected "str"  [arg-type]
src/example_04.py:10: error: Unsupported operand types for + ("str" and "int")  [operator]
Found 2 errors in 1 file (checked 1 source file)
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$ mypy src/example_04.py
src/example_04.py:10: error: Argument 1 to "greeting" has incompatible type "int"; expected "str"  [arg-type]
src/example_04.py:14: error: Unsupported operand types for + ("str" and "int")  [operator]
Found 2 errors in 1 file (checked 1 source file)
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$
```

Interrogate

- I know writing documentation is boring, but someone has to do it
- When a project starts to grow, it happens that one forgets to write documentation for certain classes or functions
- Interrogate goes in search of discovered documentation codes
- Types on command line pip3 install interrogate

Interrogate

```
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$ interrogate src/
RESULT: FAILED (minimum: 80.0%, actual: 0.0%)
```

Interrogate

```
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$ interrogate -v src/
============================================================================== Coverage for /home/matteo/PycharmProjects/torchGeo/src/ ==============================================================================
------------------------------------------------------------------------------------------------------------------------ Summary -------------------------------------------------------------------------------------------------------------------------
| Name                                                 |                                Total |                                 Miss |                                Cover |                                Cover% |
|------------------------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|---------------------------------------|
| example_01.py                                        |                                    2 |                                    2 |                                    0 |                                    0% |
| example_02.py                                        |                                    2 |                                    2 |                                    0 |                                    0% |
| example_03.py                                        |                                    1 |                                    1 |                                    0 |                                    0% |
|------------------------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|---------------------------------------|
| TOTAL                                                |                                    5 |                                    5 |                                    0 |                                 0.0% |
------------------------------------------------------------------------------------------------------------------------ RESULT: FAILED (minimum: 80.0%, actual: 0.0%) -------------------------------------------------------------------------------------------------------------------------
```

Interrogate

```
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$ interrogate -vv src/
================================ Coverage for /home/matteo/PycharmProjects/torchGeo/src/ ================================
---------------------------------------------- Detailed Coverage ----------------------------------------------
| Name                                                                      |                          Status |
|---------------------------------------------------------------------------|---------------------------------|
| example_01.py (module)                                                    |                          MISSED |
|   a_long_functionn_that_will_break_the_pep8_style_guide (L1)              |                          MISSED |
|---------------------------------------------------------------------------|---------------------------------|
| example_02.py (module)                                                    |                          MISSED |
|   a_function_that_will_be_mad_flake8 (L1)                                 |                          MISSED |
|---------------------------------------------------------------------------|---------------------------------|
| example_03.py (module)                                                    |                          MISSED |
|---------------------------------------------------------------------------|---------------------------------|

---------------------------------------------------- Summary ----------------------------------------------------
| Name                     |                Total |                 Miss |                Cover |               Cover% |
|--------------------------|----------------------|----------------------|----------------------|----------------------|
| example_01.py            |                    2 |                    2 |                    0 |                   0% |
| example_02.py            |                    2 |                    2 |                    0 |                   0% |
| example_03.py            |                    1 |                    1 |                    0 |                   0% |
|--------------------------|----------------------|----------------------|----------------------|----------------------|
| TOTAL                    |                    5 |                    5 |                    0 |                 0.0% |
------------------------------------ RESULT: FAILED (minimum: 80.0%, actual: 0.0%) ------------------------------------
(torchGeo) matteo@matteo-GS66-Stealth-10SF:~/PycharmProjects/torchGeo$
```

DVC

- Have you ever tried using git to version your data or your model?
- It should not surprise you that git has a limit on the size of resources to version
- When you want to use versioning datasets and models, you have to use the data git Data Version Control DVC
- With DVC you can track structured data such as csv, json but also unstructured data such as images or text
- The commands are identical to those in git

DVC

- The installation of dvc depends on your operating system and I recommend you follow the documentation on the DVC website

1. After doing git init type dvc init
2. The second step is to create the remote repository (let's use google drive)Create a folder on the google drive that follows this structure DVC->data->repo_name
3. Copy the last part of the link address link_copyType in terminal dvc remote add -d storage drive://link_copy
4. Commit to the git git commit .dvc/config -m "configure remote storage
5. Perform push with dvc dvc push
6. Give permission to dvc to access, edit and delete files on the repository

DVC

- We inform dvc which files or folders to keep track of with:

```
dvc add <resource_name>
```

- dvc add resource which generates a file with the same name but which has the extension .dvc resource_name.dvc which is nothing more than a file containing metadata
- Afterwards you have to inform git as suggested by dvc:

```
git add data/.gitignore resource_name.dvc
```

- Run git commit -m 'add resource data'

DVC

- To send new data to data repository type
- To pull on your local machine the data type

dvc push

dvc pull

WandB

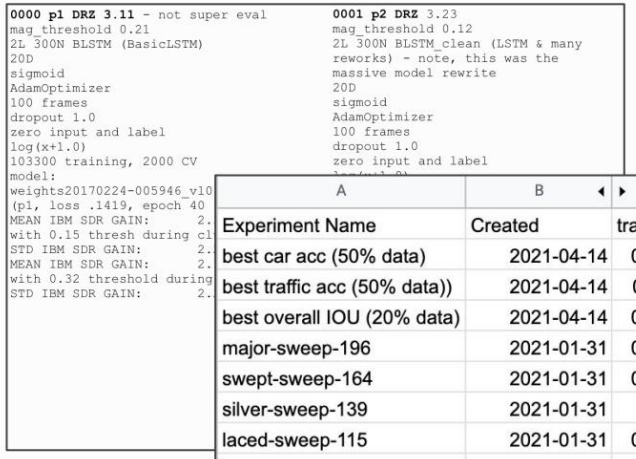
- When developing a Machine Learning project, we do not just develop one model but a series of models

0000 p1 DRZ 3.11 - not super eval
mag_threshold 0.21
2L 300N BLSTM (BasicLSTM)
20D
sigmoid
AdamOptimizer
100 frames
dropout 1.0
zero input and label
log(x+1.0)
103300 training, 2000 CV
model:
weights20170224-005946_v10.1419
(p1, loss .1419, epoch 40 [task0])
MEAN IBM SDR GAIN: 2.324 -
with 0.15 thresh during cluster
STD IBM SDR GAIN: 2.276
MEAN IBM SDR GAIN: 2.110 -
with 0.32 threshold during cluster
STD IBM SDR GAIN: 2.254

0001 p2 DRZ 3.23
mag_threshold 0.12
2L 300N BLSTM_clean (LSTM & many reworks) - note, this was the massive model rewrite
20D
sigmoid
AdamOptimizer
100 frames
dropout 1.0
zero input and label
log(x+1.0)
10330 training, 2000 CV
model:
weights20170224-032054_v10.1418
(p2, loss .1418, epoch 40 [task0])
MEAN IBM SDR GAIN: 2.056 -
with 0.15 thresh during cluster
STD IBM SDR GAIN: 2.214
MEAN IBM SDR GAIN: 2.068 -
with 0.32 threshold during cluster
STD IBM SDR GAIN: 2.205
MEAN IBM SDR GAIN: 2.108 - with fancy best SDR of the two system (eval_sdr2.py)
STD IBM SDR GAIN: 4.114
MEAN IBM SDR GAIN: 5.915 - using EXACT script measuring both voices gain (SUPER_EVAL)
STD IBM SDR GAIN: 4.349

WandB

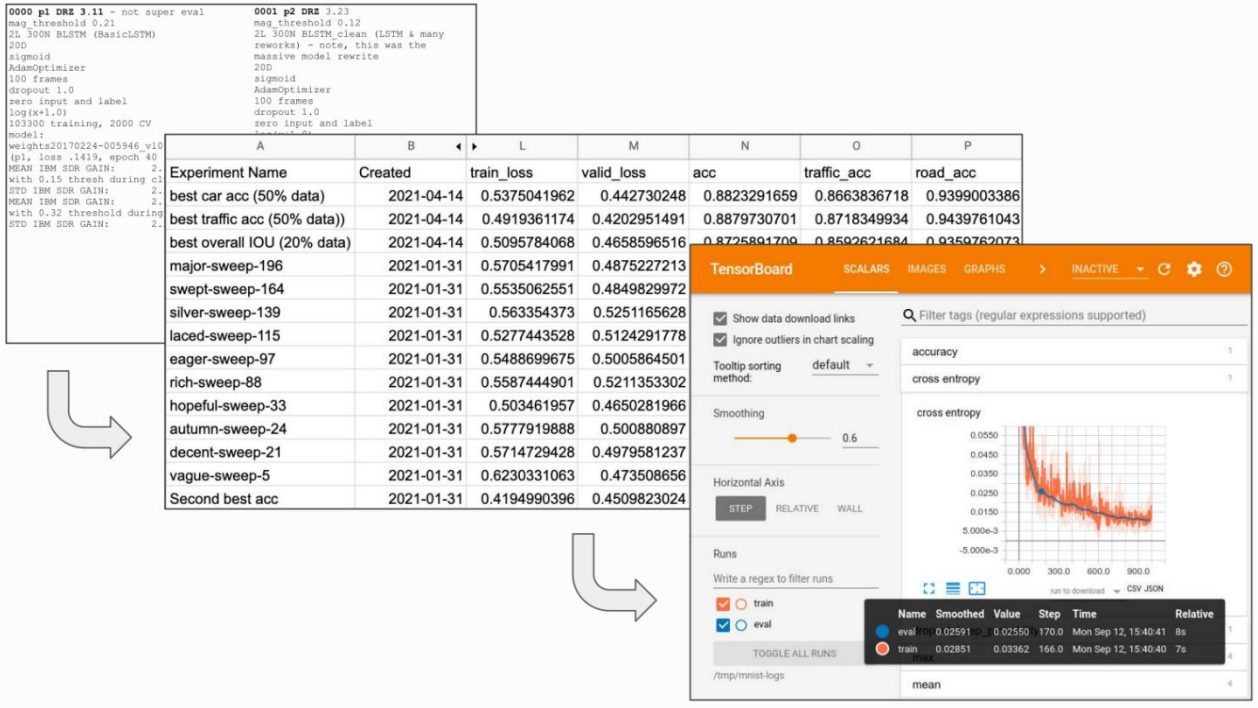
- All these models arise from different experiments where each experiment differs from the other due to a change of hyper-parameters, new data, a change of architecture, etc....



A	B	L	M	N	O	P
Experiment Name	Created	train_loss	valid_loss	acc	traffic_acc	road_acc
best car acc (50% data)	2021-04-14	0.5375041962	0.442730248	0.8823291659	0.8663836718	0.9399003386
best traffic acc (50% data))	2021-04-14	0.4919361174	0.4202951491	0.8879730701	0.8718349934	0.9439761043
best overall IOU (20% data)	2021-04-14	0.5095784068	0.4658596516	0.8725891709	0.8592621684	0.9359762073
major-sweep-196	2021-01-31	0.5705417991	0.4875227213	0.8698127866	0.8570468426	0.9454026222
swept-sweep-164	2021-01-31	0.5535062551	0.4849829972	0.8701210618	0.8567070365	0.9204238057
silver-sweep-139	2021-01-31	0.563354373	0.5251165628	0.871628046	0.846842885	0.9262287617
laced-sweep-115	2021-01-31	0.5277443528	0.5124291778	0.8705932498	0.8521561027	0.9389513731
eager-sweep-97	2021-01-31	0.5488699675	0.5005864501	0.8738754392	0.8612990975	0.913561523
rich-sweep-88	2021-01-31	0.5587444901	0.5211353302	0.8785927892	0.8512274623	0.9295567274
hopeful-sweep-33	2021-01-31	0.503461957	0.4650281966	0.8706912994	0.8560319543	0.9387732744
autumn-sweep-24	2021-01-31	0.5777919888	0.500880897	0.8755427003	0.8561192751	0.9181208611
decent-sweep-21	2021-01-31	0.5714729428	0.4979581237	0.8745227456	0.8490597606	0.9463140965
vague-sweep-5	2021-01-31	0.6230331063	0.473508656	0.8732874393	0.8601382971	0.9297611117
Second best acc	2021-01-31	0.4194990396	0.4509823024	0.8873019218	0.8705806732	0.9445936084

WandB

- It doesn't matter at some point these experiments will become too many and your boss will ask you for that experiment where that 97% f1-score was reached and you have no idea which experiment it corresponds to because you stopped tracking the experiments.

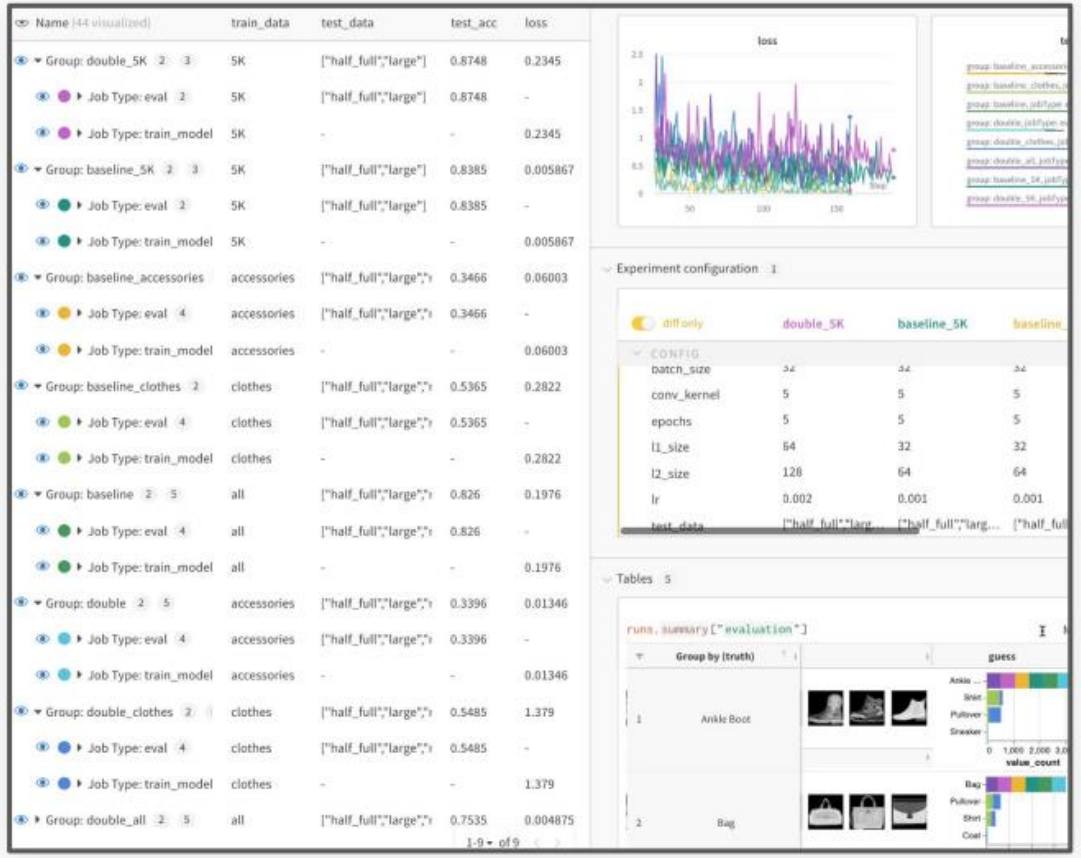


Experiment Name	Created	train_loss	valid_loss	acc	traffic_acc	road_acc
best car acc (50% data)	2021-04-14	0.5375041962	0.442730248	0.8823291659	0.8663836718	0.9399003386
best traffic acc (50% data))	2021-04-14	0.4919361174	0.4202951491	0.8879730701	0.8718349934	0.9439761043
best overall IOU (20% data)	2021-04-14	0.5095784068	0.4658596516	0.8725891709	0.8592621684	0.9359762073
major-sweep-196	2021-01-31	0.5705417991	0.4875227213			
swept-sweep-164	2021-01-31	0.5535062551	0.4849829972			
silver-sweep-139	2021-01-31	0.563354373	0.5251165628			
laced-sweep-115	2021-01-31	0.5277443528	0.5124291778			
eager-sweep-97	2021-01-31	0.5488699675	0.5005864501			
rich-sweep-88	2021-01-31	0.5587444901	0.5211353302			
hopeful-sweep-33	2021-01-31	0.503461957	0.4650281966			
autumn-sweep-24	2021-01-31	0.5777919888	0.500880897			
decent-sweep-21	2021-01-31	0.5714729428	0.4979581237			
vague-sweep-5	2021-01-31	0.6230331063	0.473508656			
Second best acc	2021-01-31	0.4194990396	0.4509823024			

WandB

- Weight and Bias (Wandb) is a tool that allows you to track and visualise your experiments
- WandB can run locally by downloading the docker image or as a service on a remote server
- It began as a tool for tracking and sharing experiments, but over time has evolved and can now be used for data and model versioning and optimisation

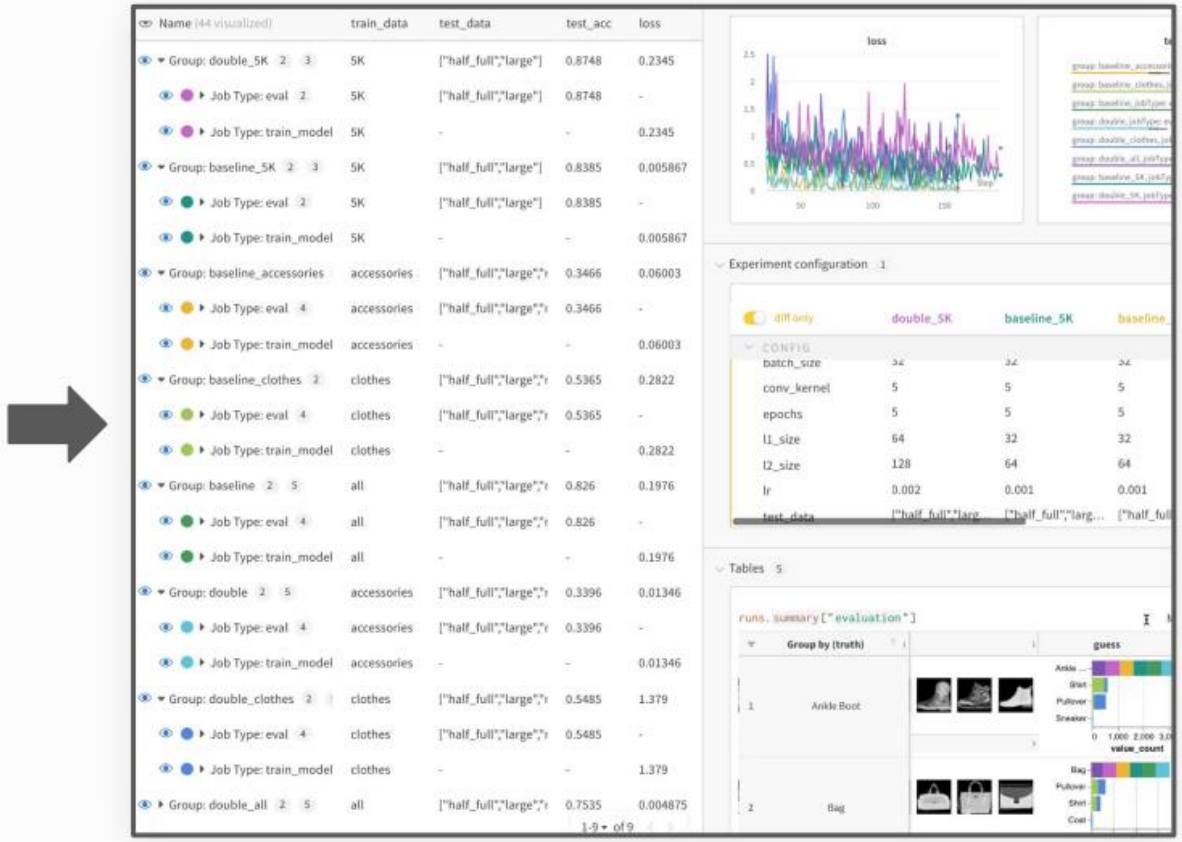
WandB


Name (44 visualized)
train_data
test_data
test_acc
loss
Group: double_5K 2 3
Job Type: eval 2
Job Type: train_model
Group: baseline_5K 2 3
Job Type: eval 2
Job Type: train_model
Group: baseline_accessories
Job Type: eval 4
Job Type: train_model
Group: baseline_clothes 2
Job Type: eval 4
Job Type: train_model
Group: baseline 2 5
Job Type: eval 4
Job Type: train_model
Group: double 2 5
Job Type: eval 4
Job Type: train_model
Group: double_clothes 2
Job Type: eval 4
Job Type: train_model
Group: double_all 2 5
loss
Experiment configuration 1
diff only
double_5K
baseline_5K
CONFIG
batch_size
conv_kernel
epochs
l1_size
l2_size
lr
test_data
Tables 5
runs.summary["evaluation"]
Group by (truth)
guess
Ankle Boot
Bag
value_count

WandB

Get started in 60 seconds

```
!pip install wandb    # Install W&B

wandb.init()          # Start experiment
wandb.log(metrics)    # Log metrics + more!
```



Optuna

- To push a machine learning model to its limits on a dataset, you have to look for the best values for the hyper-parameters in your model
- Surely you have used grid search and random search at least once, so you are not new to parameter tuning and know the pitfalls of this practice
- A best way to perfom the search of the hyper-parameters is to use Optuna

Optuna

- Optuna is a state-of-the-art hyperparameter optimization library
- Optuna enables users to adopt state-of-the-art algorithms for sampling hyperparameters and pruning unpromising trials. This helps to speed up optimization time and performance greatly compared to traditional methods such as grid search o random search
- Optuna uses the Bayesian optimization theory to search the best value for the hyper-parameters

Optuna

- Bayesian optimization methods search for global optimization by iteratively building a probabilistic model of the function mapping from hyperparameter values to the objective function
- The probabilistic model captures beliefs about the behavior of the function to form a posterior distribution over the objective function
- Using Bayesian theory, Optuna almost always manages to do better than traditional search methods without the need for expensive hardware and in a fairly time.

Optuna

- Using Bayesian theory, Optuna almost always manages to do better than traditional search methods without the need for expensive hardware and in a fairly time.
- To install optuna type `pip3 install optuna`
- To start use optuna we need of two things:
  - Define a function called objective
  - Define a study object

Optuna

```python
import optuna

import sklearn.datasets
import sklearn.ensemble
import sklearn.model_selection
import sklearn.svm


# FYI: Objective functions can take additional arguments
# (https://optuna.readthedocs.io/en/stable/faq.html#objective-func-additional-args).
def objective(trial):
    iris = sklearn.datasets.load_iris()
    x, y = iris.data, iris.target

    classifier_name = trial.suggest_categorical("classifier", ["SVC", "RandomForest"])
    if classifier_name == "SVC":
        svc_c = trial.suggest_float("svc_c", 1e-10, 1e10, log=True)
        classifier_obj = sklearn.svm.SVC(C=svc_c, gamma="auto")
    else:
        rf_max_depth = trial.suggest_int("rf_max_depth", 2, 32, log=True)
        rf_n_estimators = trial.suggest_int("rf_n_estimators", 10, 1000)
        rf_max_features = trial.suggest_categorical("rf_max_features", [None, "sqrt", "log2"])
        classifier_obj = sklearn.ensemble.RandomForestClassifier(
            max_depth=rf_max_depth, n_estimators=rf_n_estimators, max_features=rf_max_features
        )

    score = sklearn.model_selection.cross_val_score(classifier_obj, x, y, n_jobs=-1, cv=3)
    accuracy = score.mean()
    return accuracy

```

Optuna

```
study = optuna.create_study(direction="maximize")
study.optimize(objective, n_trials=1000)
print(study.best_trial)
```

Optuna

```
[I 2023-11-07 22:45:01,369] Trial 991 finished with value: 0.98 and parameters: {'classifier': 'SVC', 'svc_c': 313461.4404857984}. Best is trial 54 with value: 0.9866666666666667.
[I 2023-11-07 22:45:01,395] Trial 992 finished with value: 0.32 and parameters: {'classifier': 'SVC', 'svc_c': 7.405993726189809e-05}. Best is trial 54 with value: 0.9866666666666667.
[I 2023-11-07 22:45:01,419] Trial 993 finished with value: 0.32 and parameters: {'classifier': 'SVC', 'svc_c': 8.001872134399831e-06}. Best is trial 54 with value: 0.9866666666666667.
[I 2023-11-07 22:45:01,444] Trial 994 finished with value: 0.96 and parameters: {'classifier': 'SVC', 'svc_c': 0.43020420921240865}. Best is trial 54 with value: 0.9866666666666667.
[I 2023-11-07 22:45:01,467] Trial 995 finished with value: 0.9666666666666667 and parameters: {'classifier': 'SVC', 'svc_c': 23.670244354705204}. Best is trial 54 with value: 0.9866666666666667.
[I 2023-11-07 22:45:01,493] Trial 996 finished with value: 0.32 and parameters: {'classifier': 'SVC', 'svc_c': 1.2529630995378923e-05}. Best is trial 54 with value: 0.9866666666666667.
[I 2023-11-07 22:45:01,519] Trial 997 finished with value: 0.9666666666666667 and parameters: {'classifier': 'SVC', 'svc_c': 1.4943274563640283}. Best is trial 54 with value: 0.9866666666666667.
[I 2023-11-07 22:45:01,544] Trial 998 finished with value: 0.98 and parameters: {'classifier': 'SVC', 'svc_c': 5.201471756437384}. Best is trial 54 with value: 0.9866666666666667.
[I 2023-11-07 22:45:01,574] Trial 999 finished with value: 0.9666666666666667 and parameters: {'classifier': 'SVC', 'svc_c': 41.284067456447524}. Best is trial 54 with value: 0.9866666666666667.

FrozenTrial(number=54, state=TrialState.COMPLETE, values=[0.9866666666666667], datetime_start=datetime.datetime(2023, 11, 7, 22, 44, 18, 665588), datetime_complete=datetime.datetime(2023, 11, 7, 22, 44, 18, 682112),
params={'classifier': 'SVC', 'svc_c': 4.129630462785546}, user_attrs={}, system_attrs={}, intermediate_values={}, distributions={'classifier': CategoricalDistribution(choices=('SVC', 'RandomForest')), 'svc_c': FloatDistribution
(high=10000000000.0, log=True, low=1e-10, step=None)}, trial_id=54, value=None)
```

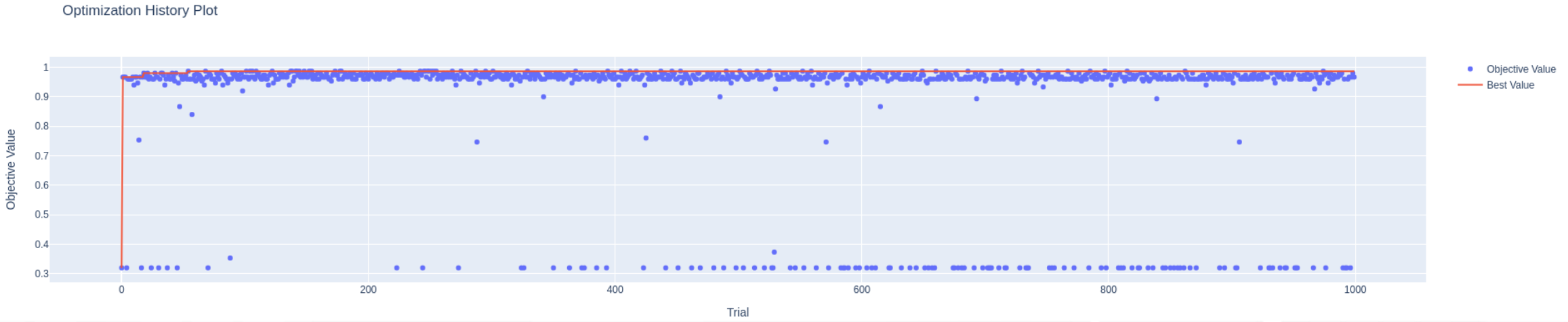
```
study.best_params
```
Executed at 2023.11.07 22:45:01 in 21ms

```
{'classifier': 'SVC', 'svc_c': 4.129630462785546}
```

Optuna

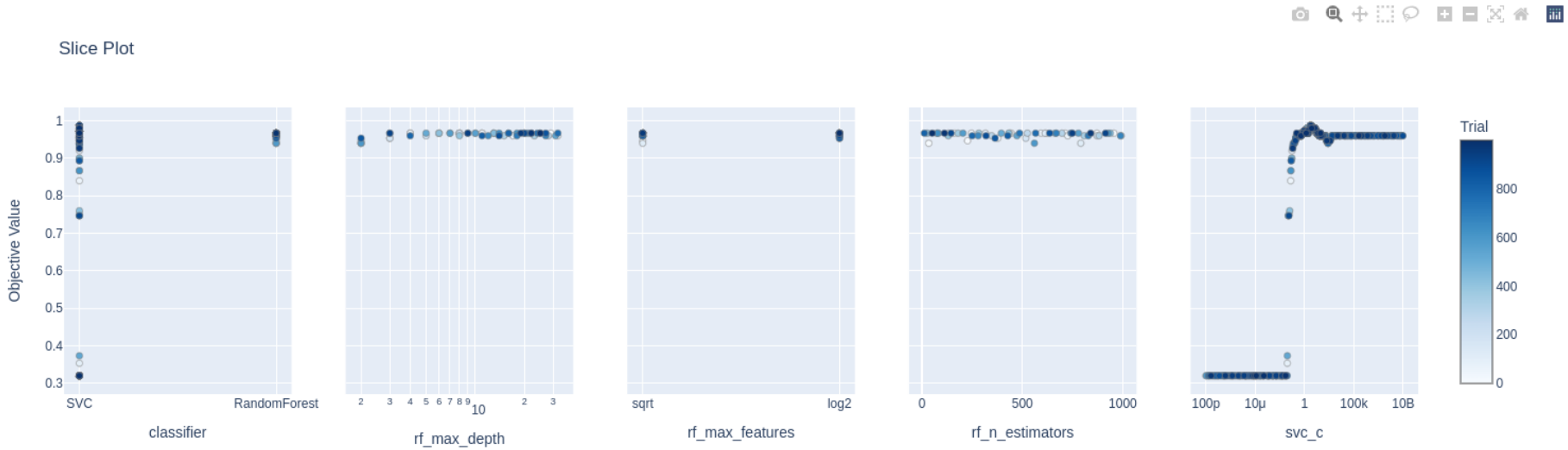
```
1 plot_optimization_history(study)
```

Executed at 2023.11.07 22:45:02 in 338ms



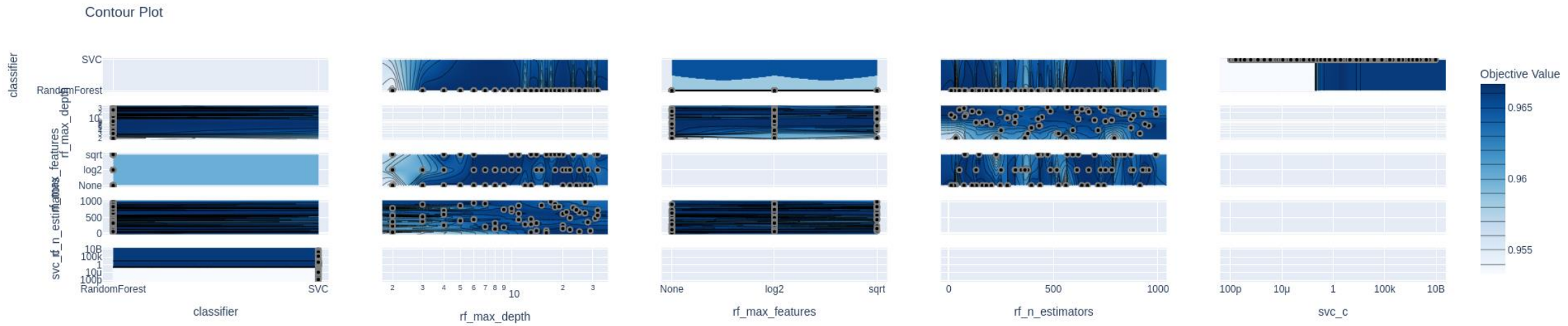
Optuna

```
In 7  1  plot_slice(study)
```
Executed at 2023.11.07 22:45:02 in 114ms



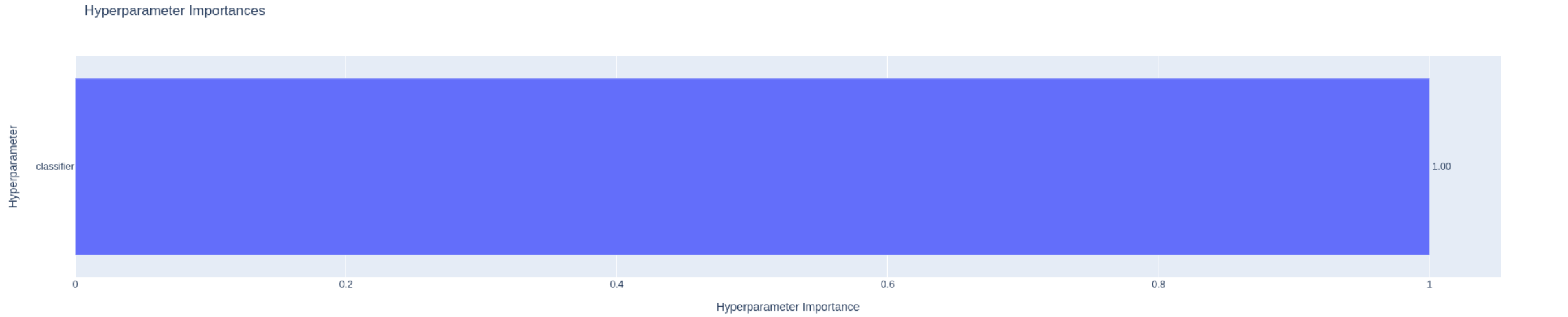
Optuna

```
In 8  1  plot_contour(study)
Executed at 2023.11.07 22:45:03 in 986ms
```



Optuna

```
In 9  1  plot_param_importances(study)
```
Executed at 2023.11.07 22:45:03 in 68ms



Optuna

```
In 14  1  study.trials_dataframe()
```

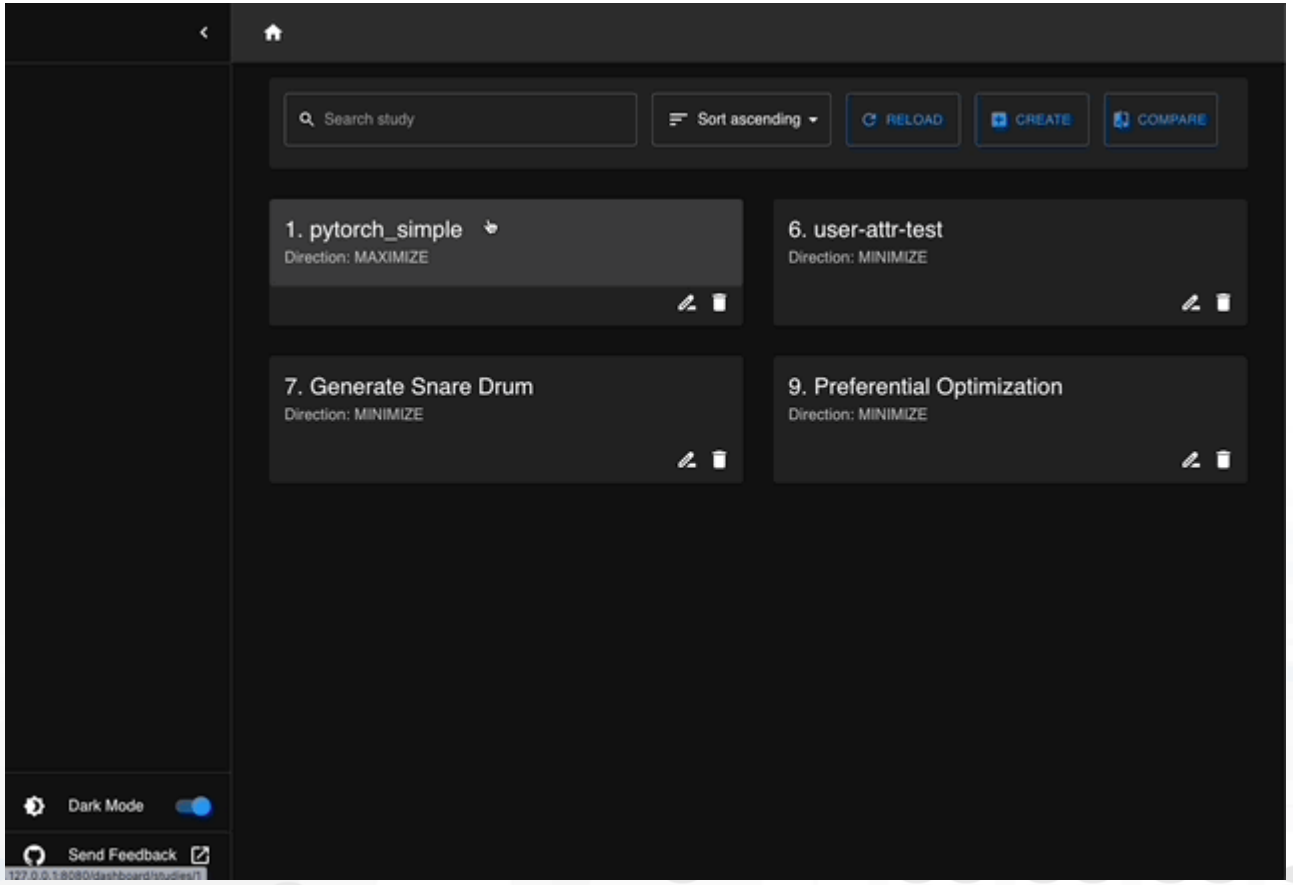
Executed at 2023.11.07 22:45:04 in 204ms

Out 14

1-10 1000 rows × 11 columns pd.DataFrame CSV

	number	value	datetime_start	datetime_complete	duration	params_classifier	params_rf_max_depth	params_rf_max_features	params_rf_n_estimators	params_svc_c	state
0	0	0.320000	2023-11-07 22:44:12.131607	2023-11-07 22:44:12.706522	0 days 00:00:00.574915	SVC	NaN	NaN	NaN	1.948677e-05	COMPLETE
1	1	0.966667	2023-11-07 22:44:12.707531	2023-11-07 22:44:13.134972	0 days 00:00:00.427441	RandomForest	8.0	None	161.0	NaN	COMPLETE
2	2	0.966667	2023-11-07 22:44:13.136063	2023-11-07 22:44:13.403994	0 days 00:00:00.267931	SVC	NaN	NaN	NaN	2.111439e+01	COMPLETE
3	3	0.966667	2023-11-07 22:44:13.404784	2023-11-07 22:44:14.513376	0 days 00:00:01.108592	RandomForest	6.0	log2	956.0	NaN	COMPLETE
4	4	0.320000	2023-11-07 22:44:14.514113	2023-11-07 22:44:14.527303	0 days 00:00:00.013190	SVC	NaN	NaN	NaN	4.648109e-07	COMPLETE
5	5	0.960000	2023-11-07 22:44:14.528039	2023-11-07 22:44:14.540901	0 days 00:00:00.012862	SVC	NaN	NaN	NaN	1.466889e+04	COMPLETE
6	6	0.960000	2023-11-07 22:44:14.541500	2023-11-07 22:44:14.554472	0 days 00:00:00.012972	SVC	NaN	NaN	NaN	1.396001e+06	COMPLETE
7	7	0.960000	2023-11-07 22:44:14.555534	2023-11-07 22:44:14.568693	0 days 00:00:00.013159	SVC	NaN	NaN	NaN	5.395809e+09	COMPLETE
8	8	0.960000	2023-11-07 22:44:14.569274	2023-11-07 22:44:14.582811	0 days 00:00:00.013537	SVC	NaN	NaN	NaN	6.889738e+02	COMPLETE
9	9	0.966667	2023-11-07 22:44:14.585592	2023-11-07 22:44:14.965305	0 days 00:00:00.379713	RandomForest	13.0	sqrt	445.0	NaN	COMPLETE

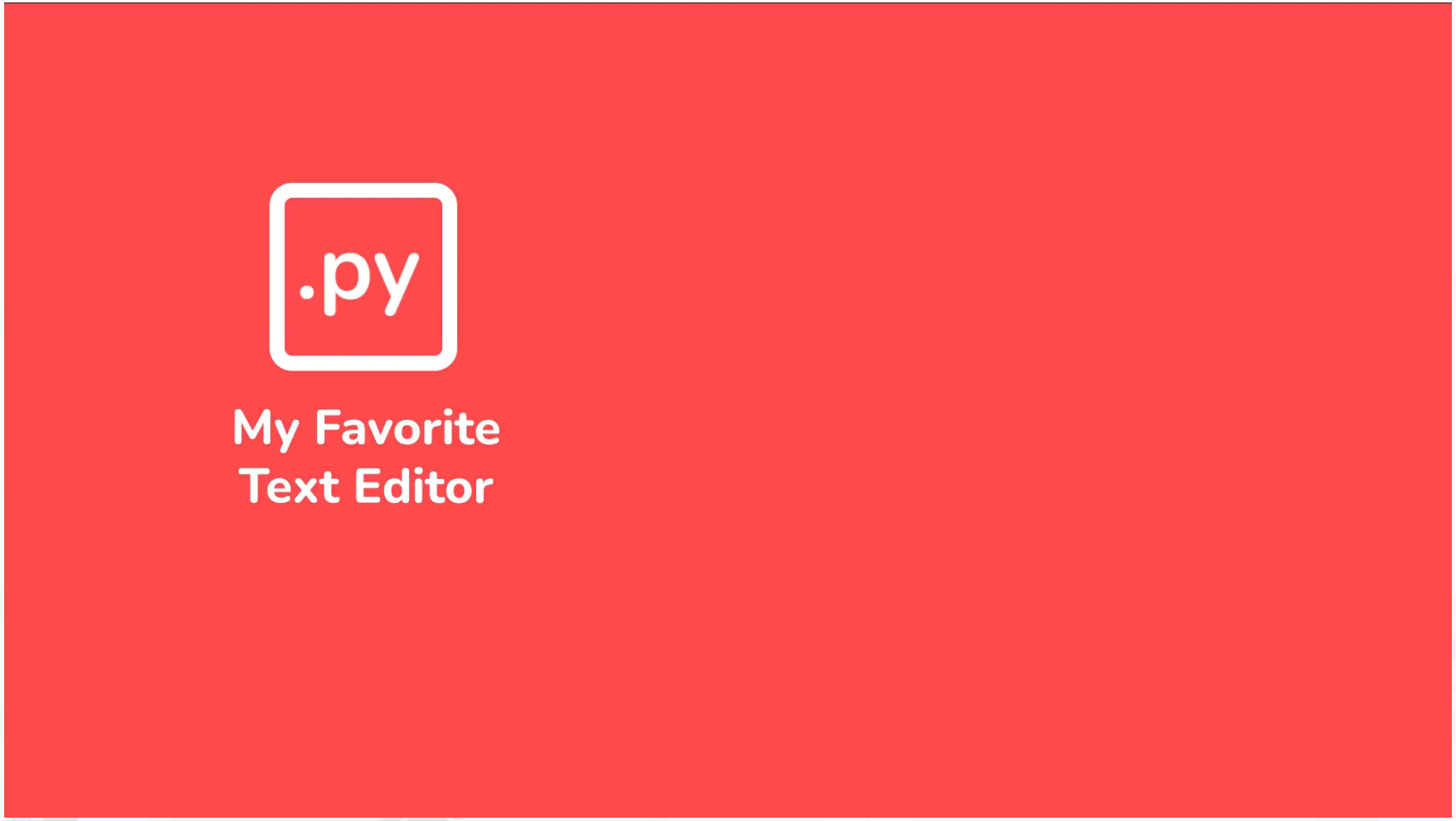
Optuna


Search study
Sort ascending
RELOAD
CREATE
COMPARE
1. pytorch_simple
Direction: MAXIMIZE
6. user-attr-test
Direction: MINIMIZE
7. Generate Snare Drum
Direction: MINIMIZE
9. Preferential Optimization
Direction: MINIMIZE
Dark Mode
Send Feedback

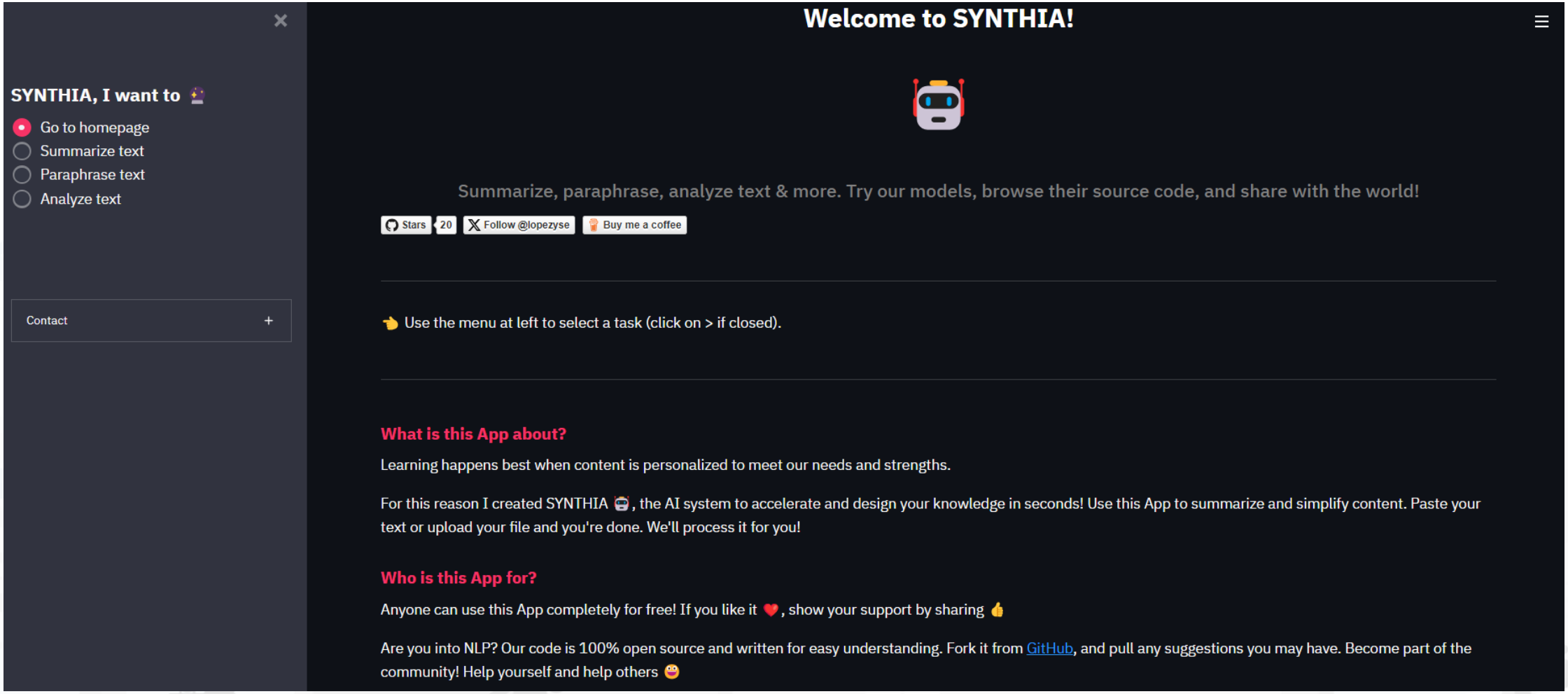
Streamlite

- There are several reasons why even if you are an AI researcher or machine learning engineer you will need to make a front-end for your application
- Streamlit is an open source framework designed to develop interfaces for ML projects
- Allows you to create a front-end in Python without being an expert
- Allows easy project deployment

Streamlit



Streamlite


SYNTHIA, I want to
Go to homepage
Summarize text
Paraphrase text
Analyze text
Contact
Welcome to SYNTHIA!
Summarize, paraphrase, analyze text & more. Try our models, browse their source code, and share with the world!
Stars 20
Follow @lopezyse
Buy me a coffee
Use the menu at left to select a task (click on > if closed).
What is this App about?
Learning happens best when content is personalized to meet our needs and strengths.
For this reason I created SYNTHIA, the AI system to accelerate and design your knowledge in seconds! Use this App to summarize and simplify content. Paste your text or upload your file and you're done. We'll process it for you!
Who is this App for?
Anyone can use this App completely for free! If you like it, show your support by sharing
Are you into NLP? Our code is 100% open source and written for easy understanding. Fork it from GitHub, and pull any suggestions you may have. Become part of the community! Help yourself and help others

Demo

- Stop talking, now I want to create a simple app with all these tools

End

End

Thank you and see you in lab and at the exam!