

APPLICATIONS DE TRAITEMENT D'IMAGES EN APPRENTISSAGE PROFOND



TSI

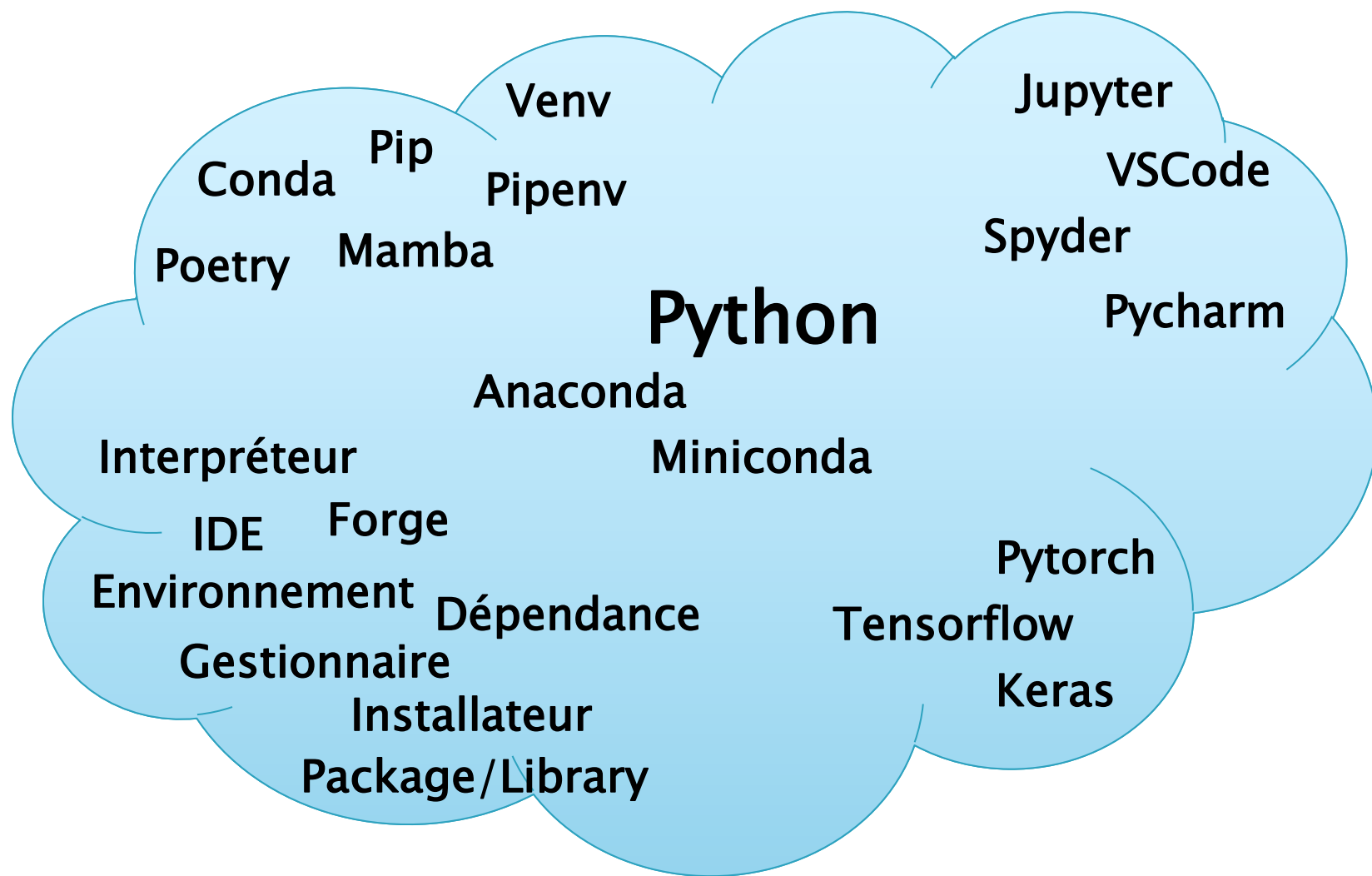
Marc Donias

Objectifs / Problématique

- ▶ Mise en œuvre d'architectures d'apprentissage profond
 - Mode « inférence »
 - Evaluation des performances
 - Absolue ou relative (comparative)
 - Au moyen de métriques
 - Sur des bases de données

- ▶ Applications de traitement des images
 - Classification (sémantique)
 - Détection
 - Classification (par instance)
 - +Localisation : étiquetage « grossier » (par parties rectangles)
 - Segmentation
 - Classification (par instance)
 - Etiquetage « fin » (par pixel)

Mots-clés





► Langage de programmation

- Interprété
- Multiparadigme : impérative structurée, fonctionnelle, orientée objet
- Multiplateformes : Windows, Unix (GNU/Linux, macOS), Android, iOS
- Quelques caractéristiques
 - ❑ Typage dynamique fort
 - ❑ Gestion automatique de la mémoire
 - ❑ Définition de blocs par indentation

► Historique

- Lancement en Février 1991 (version 0.9.0) par Guido van Rossum
- Branches incompatibles 2.x et 3.x
 - ❑ Version 2.0 en Octobre 2000 (→ 2.7 en Juillet 2010)
 - ❑ Version 3.0 en Décembre 2008 (→ 3.13 en Octobre 2024)

Installation de Python

- ▶ Python (<https://www.python.org/downloads>)
 - Juste un interpréteur du langage (pas une distribution)
- ▶ A travers une distribution
 - Combinaison
 - Langage(s)
 - Environnement(s) de développement
 - Outils
 - Bibliothèques
 - Populaires
 - Anaconda
 - Miniconda
- ▶ A travers un environnement de développement
 - VSCode

Distributions

- ▶ Anaconda (<https://www.anaconda.com/download>)
 - Distribution ouverte incluant les langages Python et R
 - Orientée science des données et IA (ML/DL)
 - IDE intégrés (Jupyter Notebook, Spyder, RStudio)
 - 250 Librairies
 - Forge (<https://anaconda.org/anaconda/repo>)
 - Outil propre pour la gestion de packages/bibliothèques, de dépendances et d'environnements
- ▶ Miniconda (<https://www.anaconda.com/download>)
 - Distribution ouverte et réduite nativement au langage Python, à Conda et à quelques packages

Gestionnaire de packages (1 / 2)

► Package

- Ensemble de fonctionnalités (classes, fonctions, modules) non présentes dans le langage Python
- « Synonyme » de bibliothèque ou de librairie

► Gestionnaire

- Accès (téléchargement) à des forges (dépôts de packages) avec spécification de la version (la dernière, précisément, supérieure à)
- Installation
- Gestion (plus ou moins robuste) des dépendances entre packages

► Pip (Forge PyPI, <https://pypi.org>)

- Gestionnaire « natif », évolutif et pas intégré par défaut (~package !)
- Gestion partielle des dépendances (souci avec les conflits)

Gestionnaire de packages (2 / 2)

► Conda

- Dépasse le cadre du langage Python
- Intégré à Anaconda
- Forges conda-forge (<https://anaconda.org/conda-forge>) et bioconda (<https://bioconda.github.io/>)
- Inclut un gestionnaire d'environnements

► Mamba

- Ré-implémentation multi-thread de Conda en C++
- Gestion plus rapide des dépendances (téléchargements en parallèle)

► Poetry (<https://python-poetry.org>)

- Limité au langage Python
- Résolution des dépendances garantissant la compatibilité
- Inclut un gestionnaire d'environnements

Gestionnaire d'environnements (1 / 2)

► Contexte

- Cas d'usage : mise à jour d'un package qui « détruit » d'autres développements
- Solution : environnement virtuel jouant le rôle d'isolant
 - Versions différentes de Python
 - Versions différentes (et multiples) de packages

► Virtualenv (package à installer)

- Création d'un répertoire local (copie de Python et des packages)
- Installation des packages avec Pip

► Conda

- Inclus dans Anaconda ou package à installer
- Non limité au langage Python

Gestionnaire d'environnements (2/2)

- ▶ **Pyenv (package à installer)**
 - Choix de la version de Python lancée par redirection (aucune copie)
- ▶ **Autres (packages à installer)**
 - Virtualenvwrapper (extension de virtualenv)
 - Pyenv-virtualenv et Pyenv-virtualenvwrapper (« combo »)
 - Pipenv
- ▶ **Venv (inclus dans Python à partir de 3.3)**
 - Version « réduite » et simple de Virtualenv
 - Mise à jour impossible
- ▶ **Pyvenv (inclus dans Python de 3.3 à 3.7)**

Environnements de développement

- ▶ Integrated Development Environment (IDE)
 - Outil multi-fonction de développement améliorant la productivité
 - Edition « intelligente » (surlignage, complétion, alertes)
 - Débogage/Test
 - Mise en production (packages, etc.)
- ▶ Spyder (léger et inclus dans Anaconda)
- ▶ PyCharm Community (personnalisable, débogueur visuel)
- ▶ VSCode (multi-langage et collaboratif)
- ▶ Jupyter Notebook (multi-langage et création d'un carnet/journal de bord)

Bibliothèques d'apprentissage profond

► Keras

- Bibliothèque de réseaux de neurones pour Python
- Facile d'usage et prototypage rapide (la plus populaire)
- Encapsule les frameworks TensorFlow, Theano, CNTK
- Performances limitées

► TensorFlow

- Bibliothèque IA (ML/DL) de Google intègre TensorBoard (visualisation)
- Supporte de nombreux CPU et GPU
- La plus utilisée en contexte industriel (production et performance)

► Pytorch

- Bibliothèque IA (ML/DL) développée par Meta AI (Meta)
- Plus versatile et davantage utilisée en contexte de recherche

Classification

► Définition

- Identification de la classe d'appartenance (catégorie) d'une image
- Caractéristiques
 - Cas binaire (deux classes) ou multi-classes
 - Score ou probabilité d'affectation

► Architectures

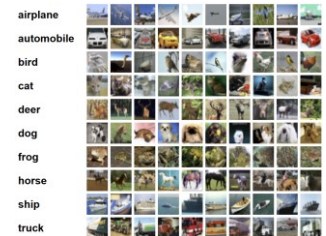
- AlexNet (2012)
- VGG-16 (2014)
- VGG-19 (2014)
- Res-Net50 (2015)
- GoogleNet (2014)
- Inception V3 (2015)

Base de données

- ▶ MNIST (<https://yann.lecun.com/exdb/mnist>)
 - Base standard de chiffres manuscrits (10 classes)
 - Images monochromes de faibles dimensions (28×28)
 - 60000 images (entraînement) + 10000 images (test)



- ▶ CIFAR-10 (<https://www.cs.toronto.edu/~kriz/cifar.html>)
 - 60000 images couleur 32×32
 - 10 classes de 6000 images
 - 50000 images (entraînement) + 10000 images (test)
 - Version CIFAR-100 (20 « super-classes » de 5 classes)



- ▶ ImageNET (<https://www.image-net.org/download.php>)
 - Base gigantesque (14M d'images HR + annotations)
 - Classification (sous-base ILSVRC)/Détection



<https://paperswithcode.com/datasets>

Métriques (1 / 5)

► Matrice de confusion

- Pour une classe (extension possible au cas multi-classes)
- Répartition des individus (nombre d'occurrences) en fonction d'un score (probabilité) « seuillé »

	Réalité 1	Réalité 0
Prédiction 1	Vrai Positif (VP)	Faux Positif (FP)
Prédiction 0	Faux Négatif (FN)	Vrai Négatif (VN)

- Désigne également la matrice des probabilités par normalisation

► Justesse (Accuracy)

- Part d'individus correctement prédits (positifs et négatifs)
- Evaluation de la pertinence globale à correctement prédire

$$\text{Accuracy} = \frac{VP + VN}{VP + FP + FN + VN}$$

Métriques (2 / 5)

► Précision

- Part d'individus positifs réels parmi les positifs prédits

$$\text{Précision} = \frac{VP}{VP + FP}$$

- Evaluation de la capacité à ne prédire que des individus positifs

► Rappel

- Part d'individus positifs prédits parmi les positifs réels
- Taux de vrais positifs (TVP)

$$\text{Rappel} = \frac{VP}{VP + FN}$$

- Evaluation de la capacité à ne pas « oublier » des individus positifs

Métriques (3 / 5)

► Probabilité de fausse alarme (PFA)

- Part d'individus positifs prédits réels parmi les négatifs réels
- Taux de faux positifs (FPR)

$$\text{FPR} = \frac{FP}{FP + VN}$$

- Evaluation de la capacité à prédire positif un individu qui est négatif

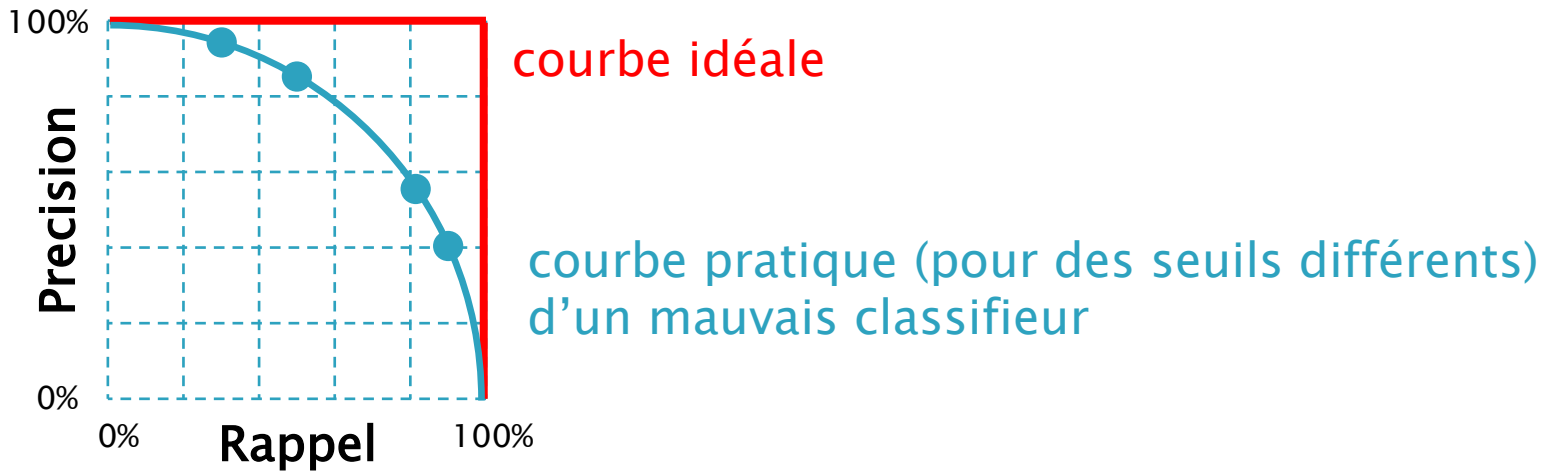
► Score F1 (F1 –score)

- Cas particulier des scores F-beta
- Moyenne harmonique (compromis) de la précision et du rappel

$$\text{F1-score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Rappel}}}$$

Métriques (4/5)

► Courbe de rappel de précision (Précision/Rappel ou PR)

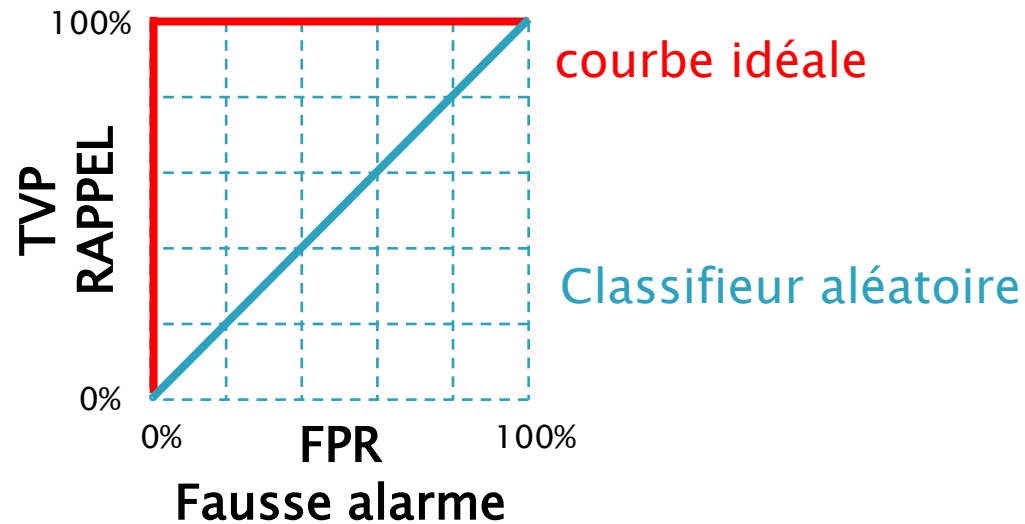


► AP

- AP_{50} , $AP50$, $AP@.5$, $AP@0.5$: précision (moyenne) pour un seuil de 0.5 ou 50% (point de la courbe PR)
- $AP_{[.5:.95]}$, $AP_{[.5:.05:.95]}$, $AP@[.5:.95]$, $AP@[.5:.05,.95]$: moyenne de précision (moyenne) pour différents seuils (aire sous la courbe PR)
- mAP ou AP : moyenne de la moyenne de précision (moyenne) pour différents seuils sur l'ensemble des classes

Métriques (5 / 5)

- ▶ Courbe ROC (TVP en fonction de FPR)



- ▶ AUC (surface sous la courbe ROC)

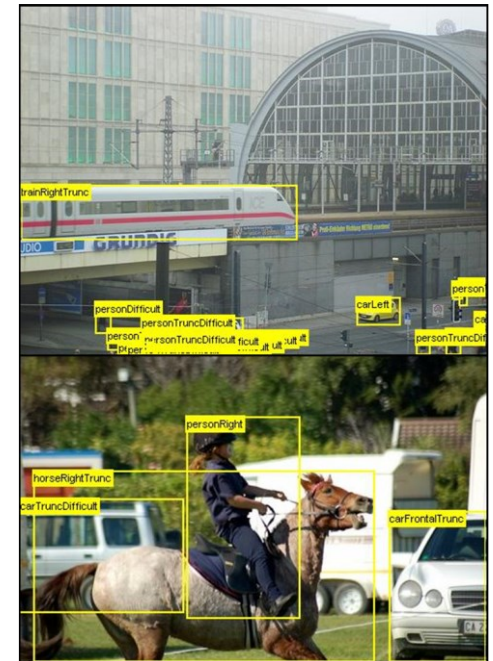
Détection

► Définition

- Identification de la classe d'appartenance (catégorie) d'objets détectés dans une image
- Caractéristiques (par objet détecté)
 - ❑ Localisation (partie rectangle ou boîte englobante)
 - ❑ Score ou probabilité d'affectation de classe

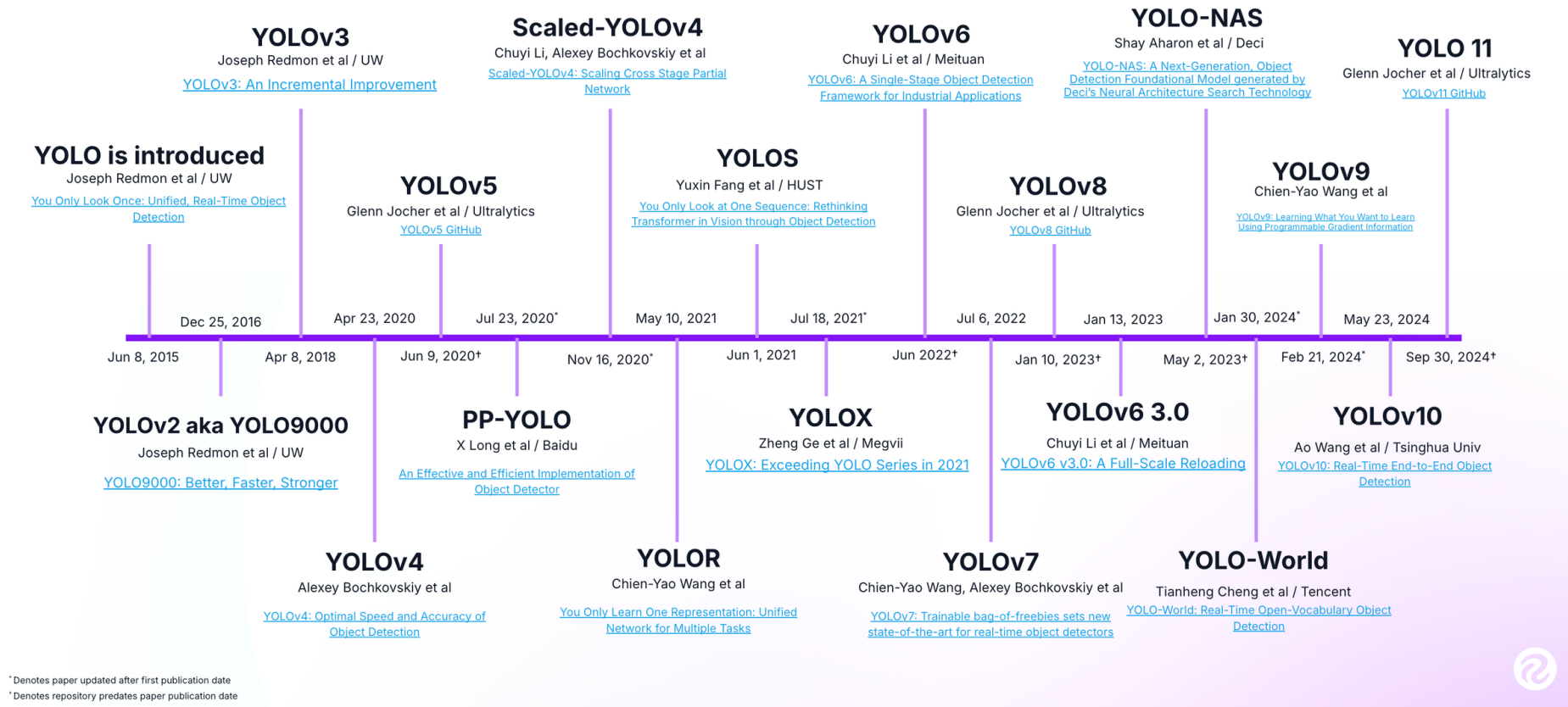
► Architectures

- En une étape
 - ❑ YOLO (multiples versions)
 - ❑ SSD, RetinaNet, CenterNet, EfficientDet
- En deux étapes
 - ❑ Construit sur une architecture de classification
 - ❑ R-CNN, SPP-Net, Fast R-CNN, Faster R-CNN, FPN, R-FCN, Mask R-CNN, detectoRS



@PASCAL VOC 12

Famille YOLO



@<https://blog.roboflow.com/guide-to-yolo-models>

Scores de boites englobantes

► IoU (Intersection over Union)

- Index ou coefficient de similarité de Jacquard

$$IoU = \frac{|A \cap B|}{|A \cup B|}$$

► Indice de Sorensen–Dice

- Indice de Sorensen ou coefficient de Dice

$$SD = \frac{2|A \cap B|}{|A| + |B|}$$

► GloU

$$GloU = IoU - \frac{|C - A \cup B|}{|C|}$$

