



2^eme année Master MIDVI

Année universitaire:2022/2023



FACULTÉ DES SCIENCES DHAR EL MAHRAZ
UNIVERSITÉ SIDI MOHAMED BEN ADELLAH

VIDEO ANALYTICS

Pr. MAHRAZ Mohamed Adnane
Laboratoire LISAC
adnane_1@yahoo.fr

EXEMPLE 1



vidéo surveillance

EXEMPLE 2

*Town Center (Body)
sequence*

EXEMPLE 2

Town Center (Head)
sequence

EXEMPLE 2

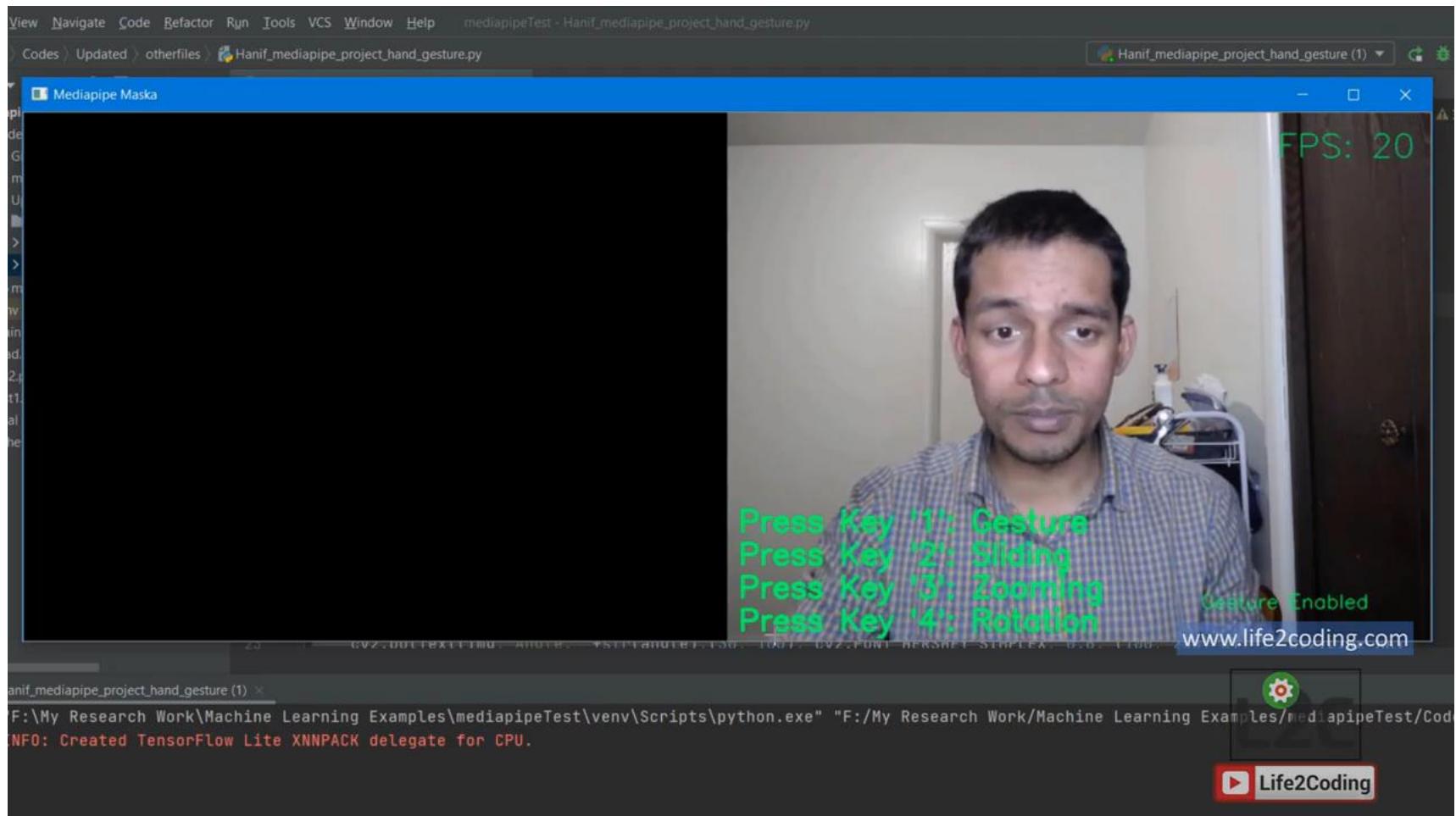
*LISA10 Urban
sequence*

EXEMPLE 3



Transformer un match de foot en big data

EXEMPLE 4



QU'EST-CE QUE L'ANALYSE VIDÉO ?

L'analyse vidéo consiste à générer automatiquement des descriptions de ce qui est en train de se produire dans la vidéo (sémantique de la vidéo). Ces descriptions ou métadonnées peuvent être utilisées pour faire des listes de personnes, de voitures et d'autres objets détectés dans le flux vidéo, ainsi que des descriptions de leur aspect ou de leurs mouvements. Ces informations peuvent alors être utilisées comme base pour agir, par exemple, pour décider s'il faut ou non avertir le personnel de sécurité, ou encore déclencher un enregistrement.

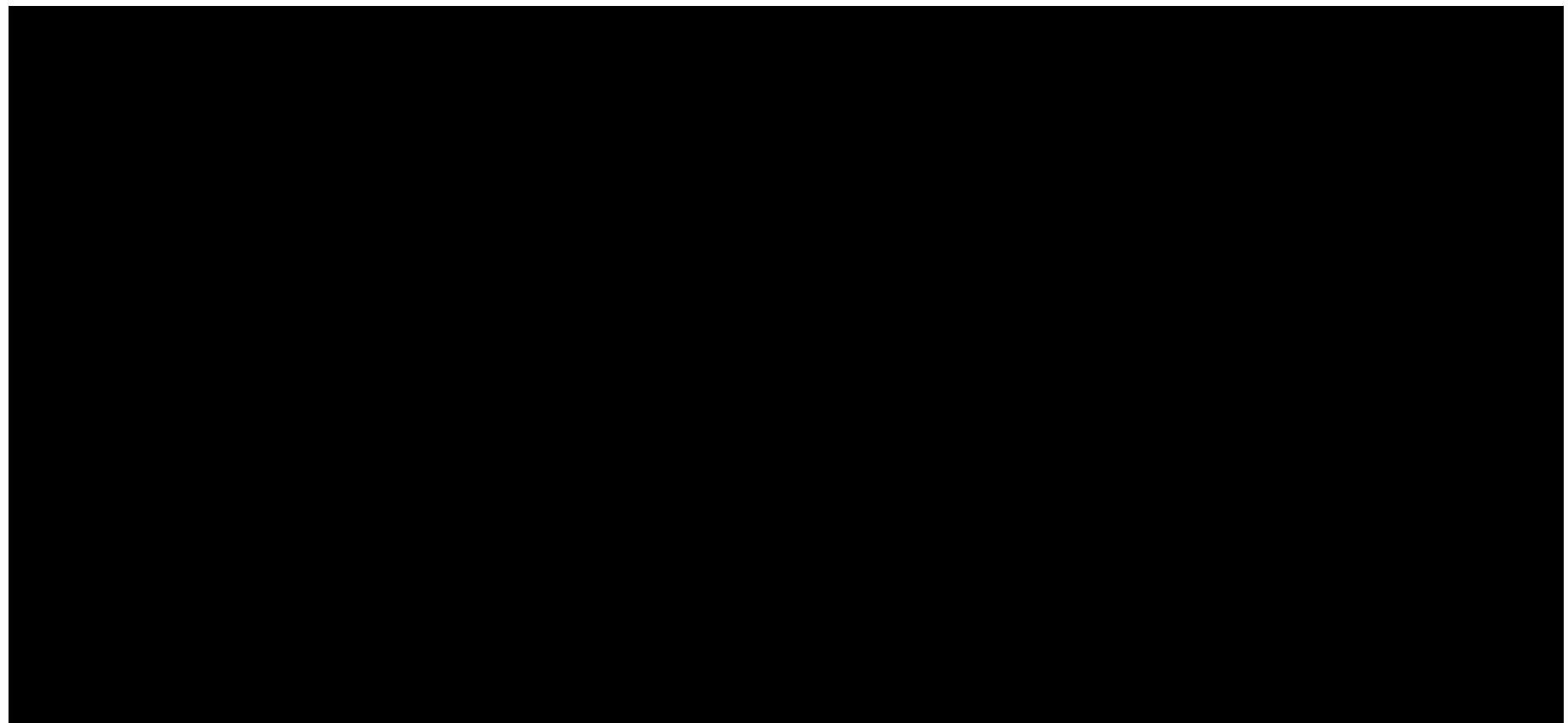
POURQUOI UTILISER L'ANALYSE VIDÉO ?

L'analyse vidéo présente de nombreux avantages, notamment :

- Économie de coûts - moins de vidéo est envoyée sur le réseau ce qui réduit sa charge et les besoins en stockage.
- Économie de temps - la surveillance et les recherches dans les vidéos enregistrées sont facilitées, ce qui permet aux opérateurs de gérer plus de caméras.
- Efficacité améliorée - la vidéosurveillance automatique des incidents relatifs à la sécurité aide à éviter les infractions au lieu de réagir après que l'incident a eu lieu.
- Création de valeur commerciale - le comptage des personnes par les caméras situées aux entrées des magasins, montre comment le système de surveillance peut permettre de mieux comprendre d'autres caractéristiques commerciales et apporter de nouvelles solutions.

QUELQUES APPLICATIONS

RECONNAISSANCE DES GESTES DE LA MAIN



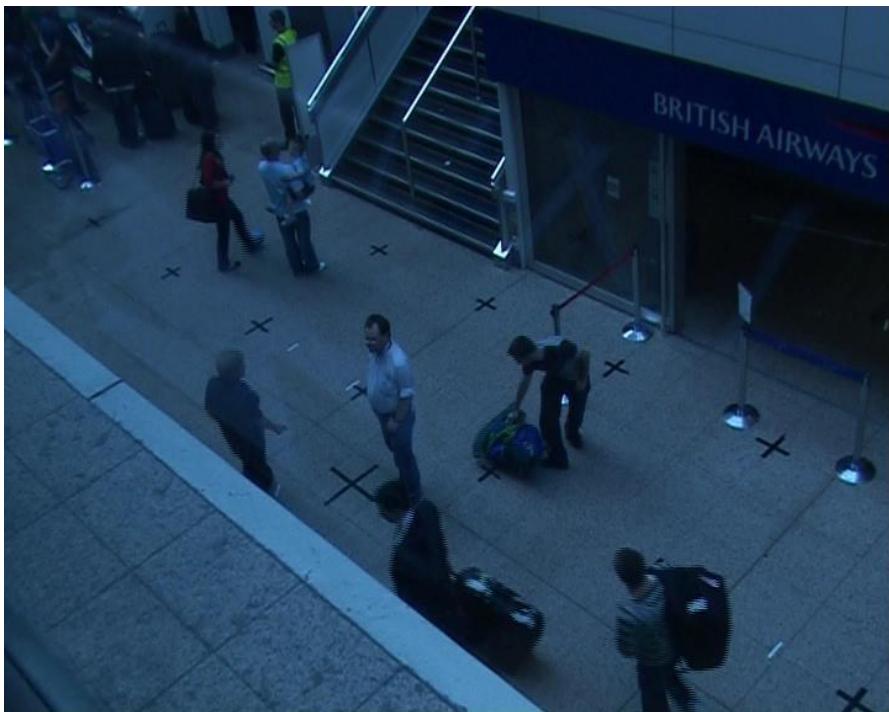
RECONNAISSANCE D'ACTIVITÉS HUMAINES



SCÉNARIOS MULTI-CAMÉRA / MULTI-VUE

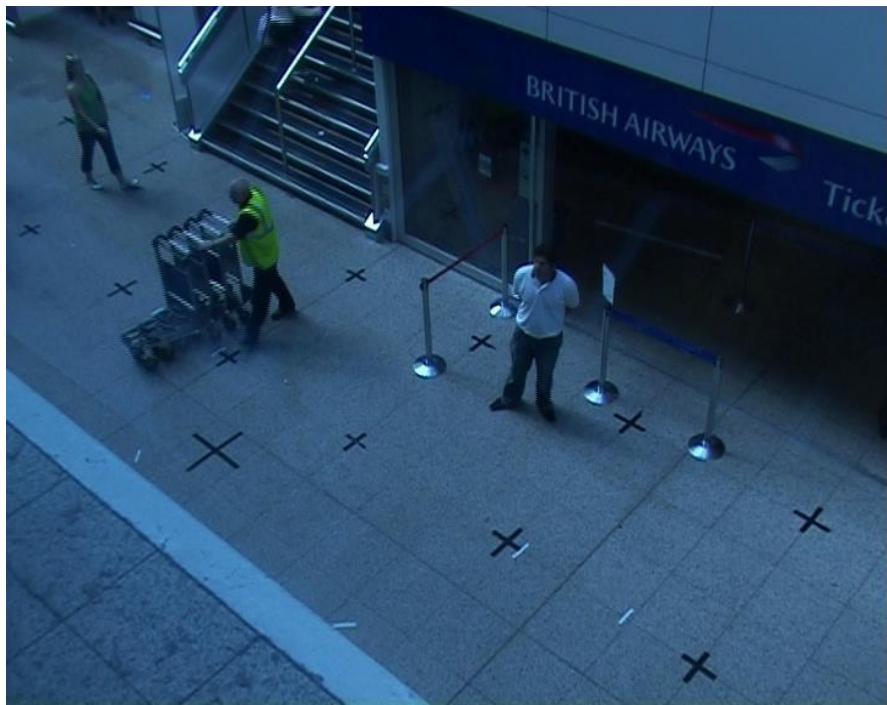


ACTIVITÉS DE LONGUE DURÉE (COMPORTEMENT)



Vol de bagage

ACTIVITÉS DE LONGUE DURÉE (COMPORTEMENT)



Bagage abandonné

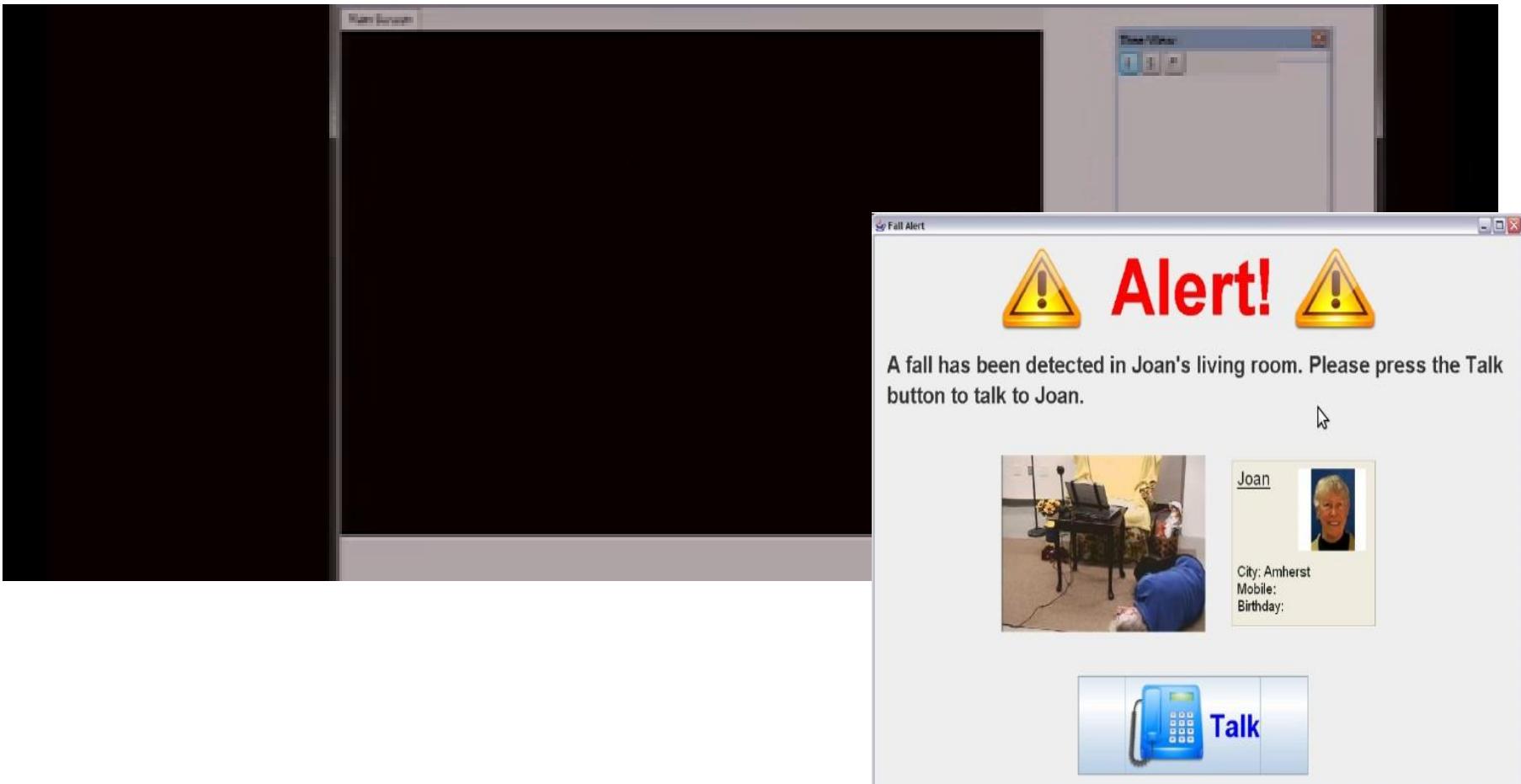


« Glander »

ANALYSE DE COMPORTEMENT DE FOULES



DÉTECTION DE CHUTES



ACQUISITION



Webcams



Téléphones

ACQUISITION



Caméras fixées



Caméras PTZ
(« Pan / Tilt / Zoom »)

ACQUISITION



Telerobot



Kompai



Pekee II

Caméras libres (robotique)!

ACQUISITION

Kinect



KINECT™
for XBOX 360.

QUELQUES PROBLÈME: DÉFIS

Exemple : lumière, couleurs

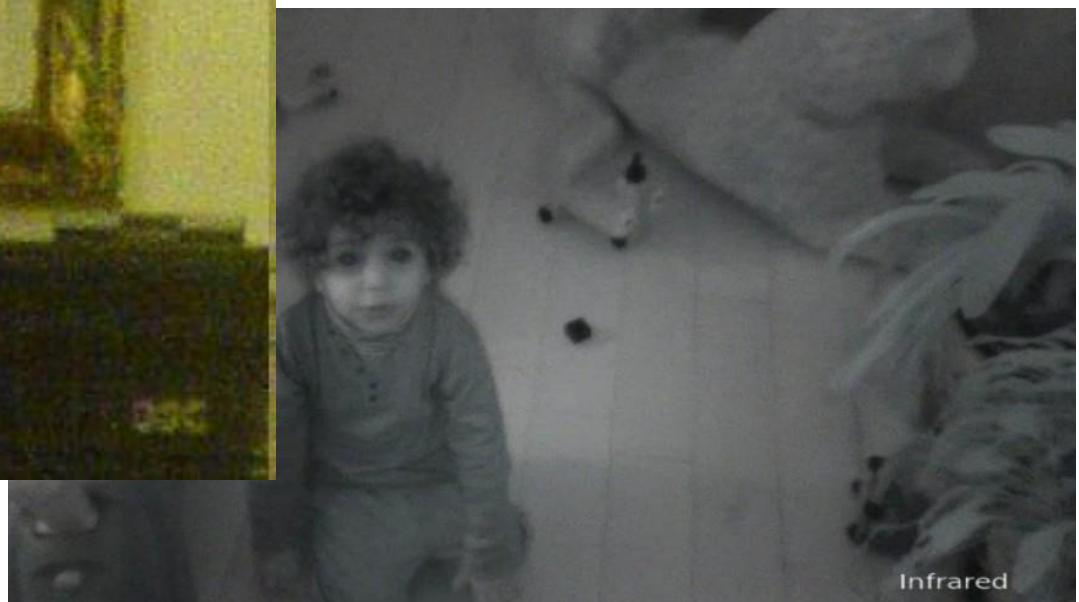


QUELQUES PROBLÈME: DÉFIS

Faible luminosité

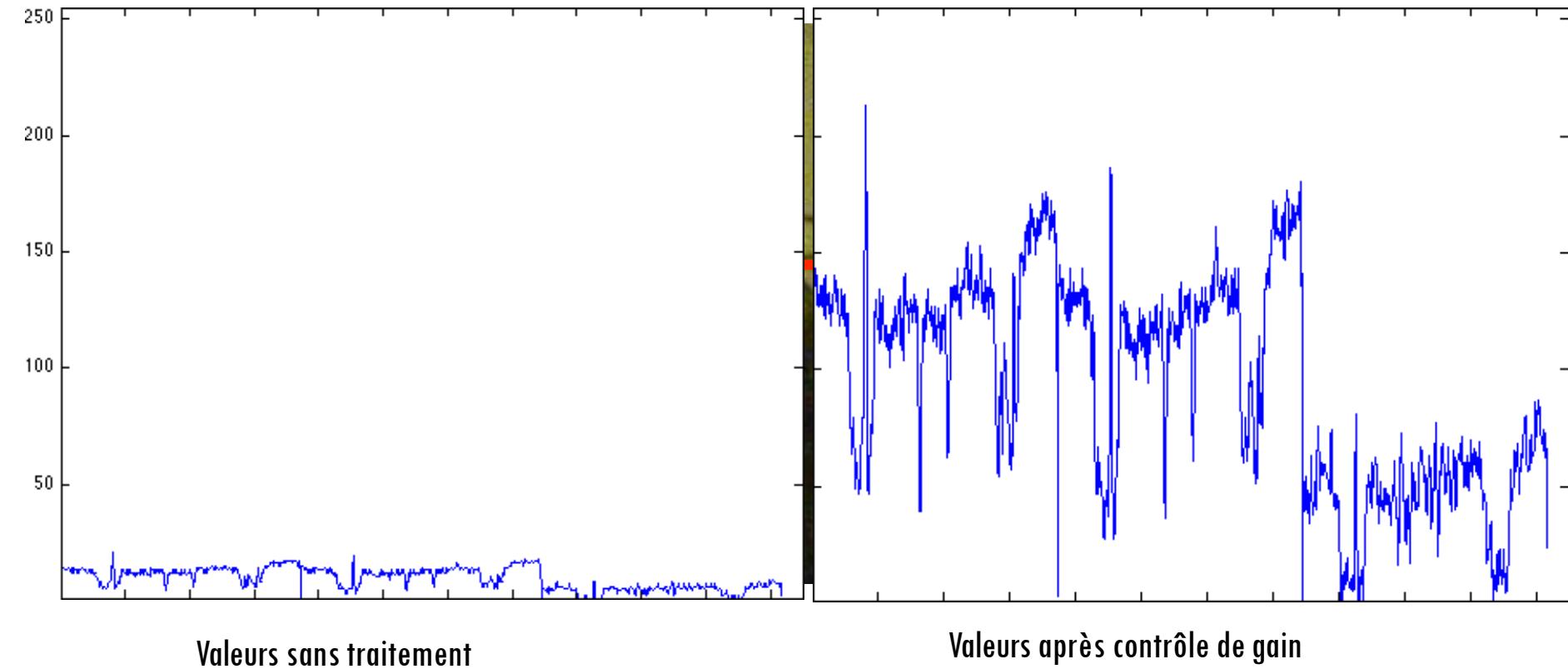


Solution : infra-rouge



QUELQUES PROBLÈME: DÉFIS

Exemple : lumière, couleurs



QUELQUES PROBLÈME: DÉFIS

Effets d'ombres



QUELQUES PROBLÈME: DÉFIS

Problème d'échantillonnage



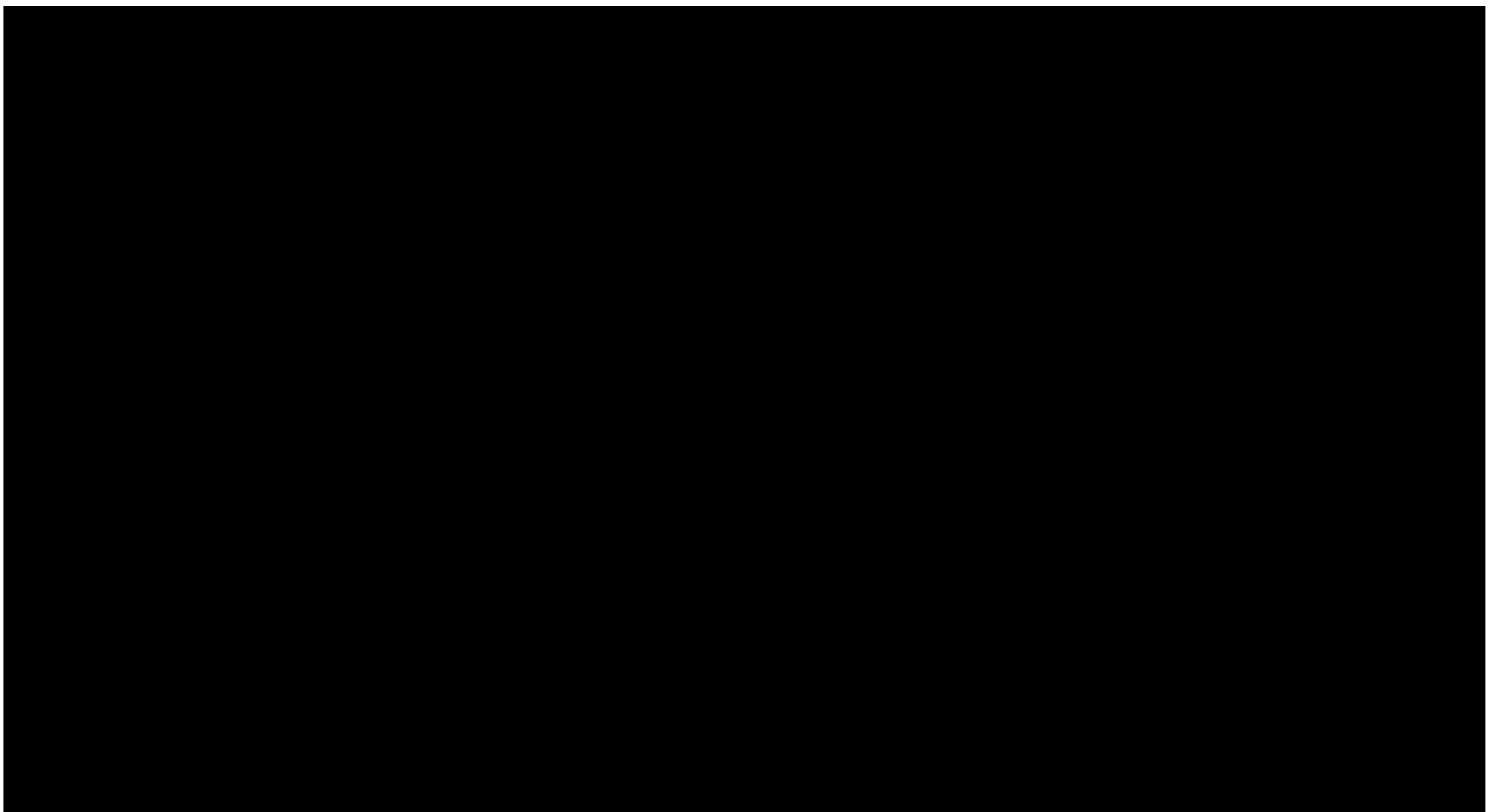
Apple iPhone 3GS



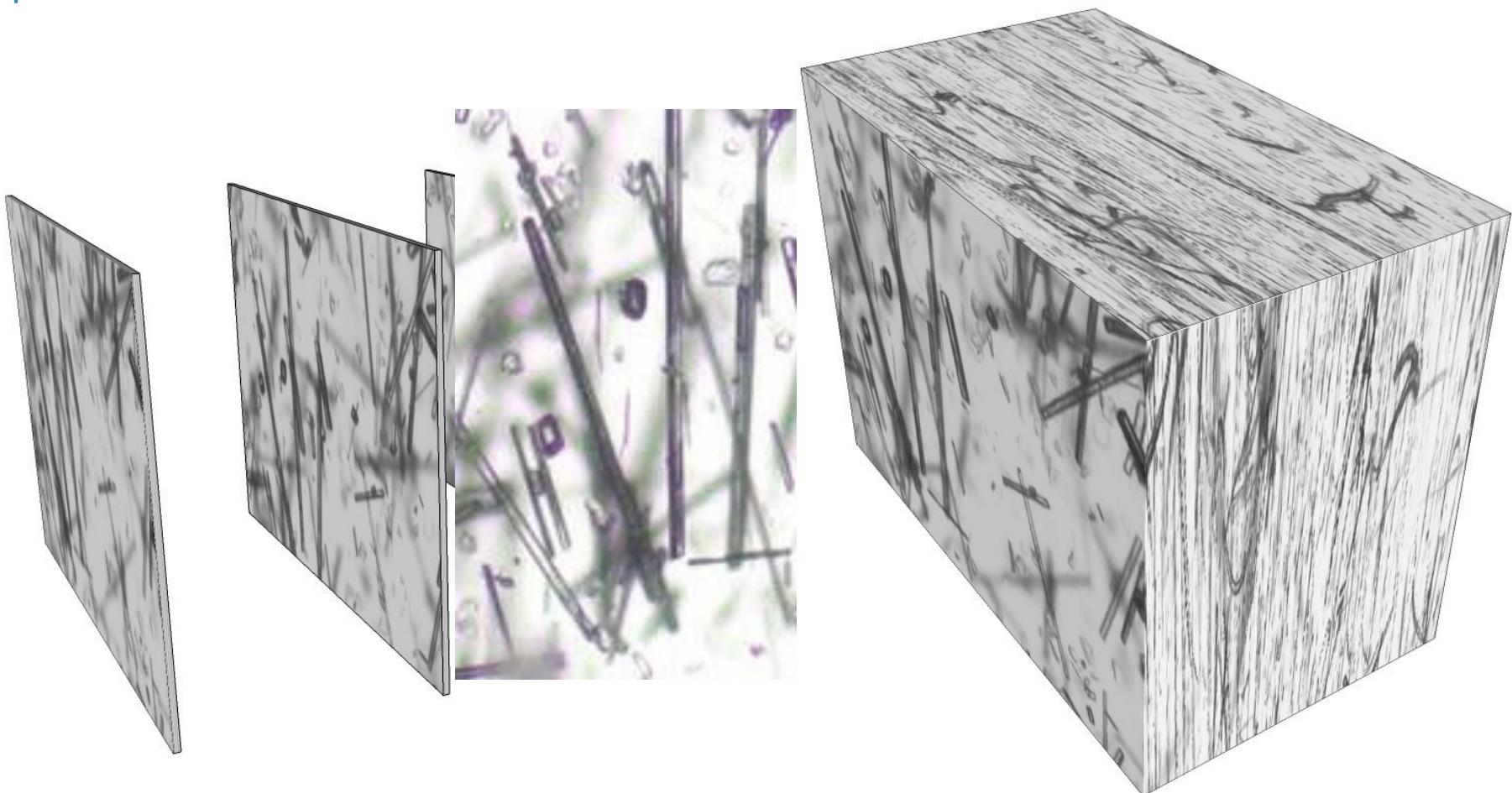
Appareil photo numérique de bas de gamme

QUELQUES PROBLÈME: DÉFIS

Vidéos entrelacées



C'EST QUOI UNE VIDÉO?



ATELIER N°1?

Lecture d'une vidéo enregistrée sur le disque dur.

Langage de programmation: Python



MODÉLISATION DE L'ARRIÈRE- PLAN: APPROCHES SIMPLES

POURQUOI MODÉLISER L'ARRIÈRE-PLAN?



Dans la plupart des cas, les objets ont plus d'intérêt que la scène
Minimiser le coût de traitement ainsi que les erreurs.

POURQUOI MODÉLISER L'ARRIÈRE-PLAN?

Etape primordiale et importante.

Estimer le modèle d'arrière-plan mène à la segmentation / détection fiable des objets de premier plan.

DOMAINE D'APPLICATION

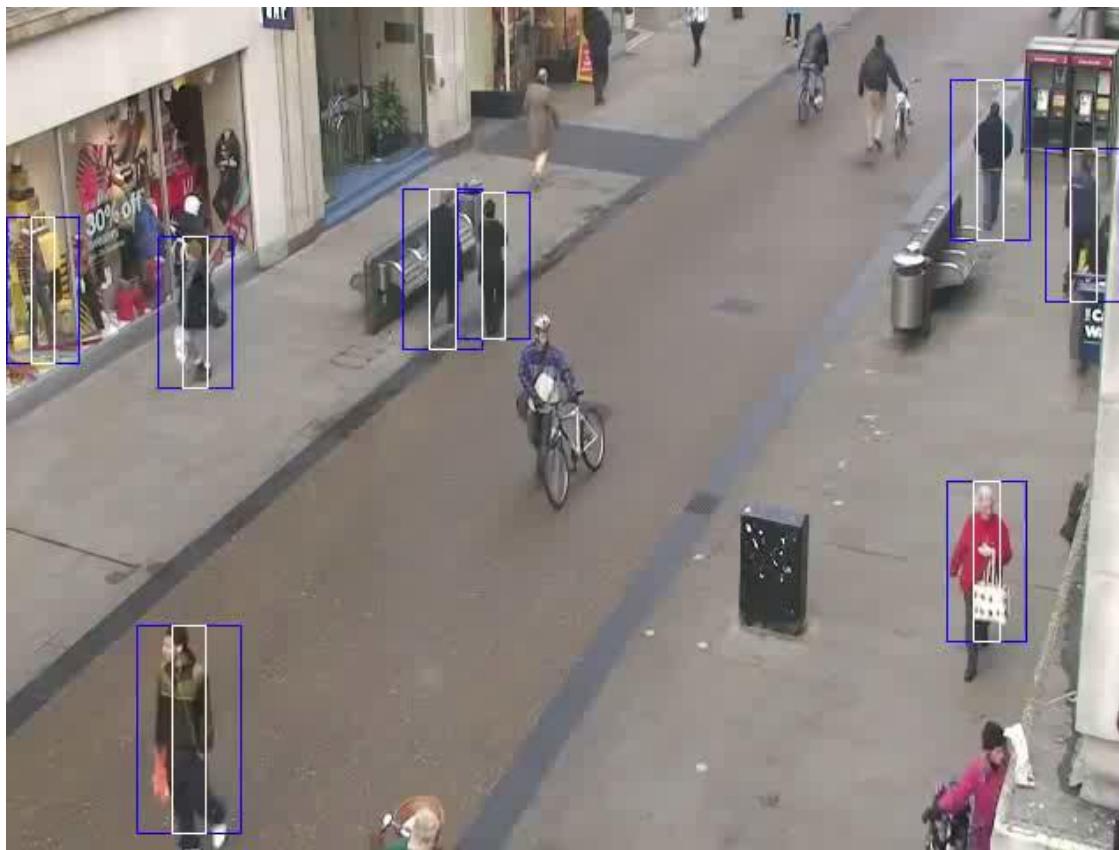
Surveillance du trafic à partir de caméras statiques
(Comptage des véhicules, détection et suivi des véhicules)

Reconnaissance d'action humaine (courir, marcher, sauter,
s'accroupir)

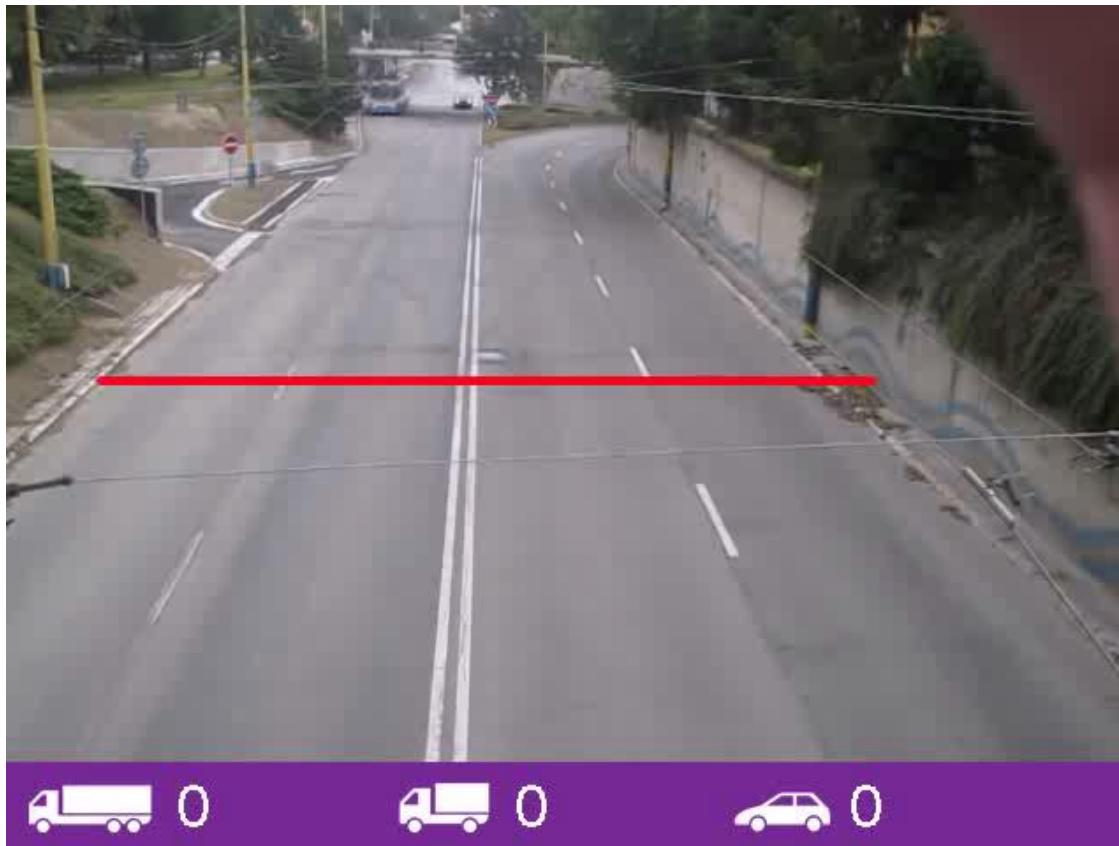
Interaction homme-machine ("interface humaine")

Suivi des objets

SUIVI DES PERSONNES



COMPTAGE DU TRAFIC



DÉTECTION ET RECONNAISSANCE FACIALE



FACTEURS À CONSIDÉRER

Changement d'éclairage

Lumière artificielle

Ombre

Emplacement (intérieur/extérieur)

ÉTUDE DE LOCALISATION - EXTÉRIEUR

Changements saisonniers

- Pluie, neige, etc.
- Soleil brillant, nuageux, brumeux, etc.

Cycle de jour (jour/nuit)

- Changement d'éclairage progressif

Phases du soleil / lune

ÉTUDE DE LOCALISATION - INTÉRIEUR

Condition d'éclairage

- Emplacement des sources
- Caractéristiques des sources

Lumière allumée / éteint

Ombre

EXIGENCES

L'image d'arrière-plan n'est pas fixe mais doit s'adapter à:

Changements d'illumination

- graduel
- soudain (comme les nuages)

Les changements de mouvement

- oscillations de la caméra
- objets de fond à haute fréquence (tels que les branches d'arbres, les vagues de la mer...)

Changements dans la géométrie d'arrière-plan

- voitures garées, etc.

APPROCHES

Type de modélisation

- Image
- Pixel

Procédure de mise à jour

- La moyenne mobile (Moving average)
- La moyenne glissante (Running Average)
- Médian

ALGORITHME - PROBLÈMES

Complexité

Vitesse

Mémoire

Précision

APPROCHES SIMPLES

Image à l'instant t: $I(x, y, t)$



Arrière-plan à l'instant t: $B(x, y, t)$



—

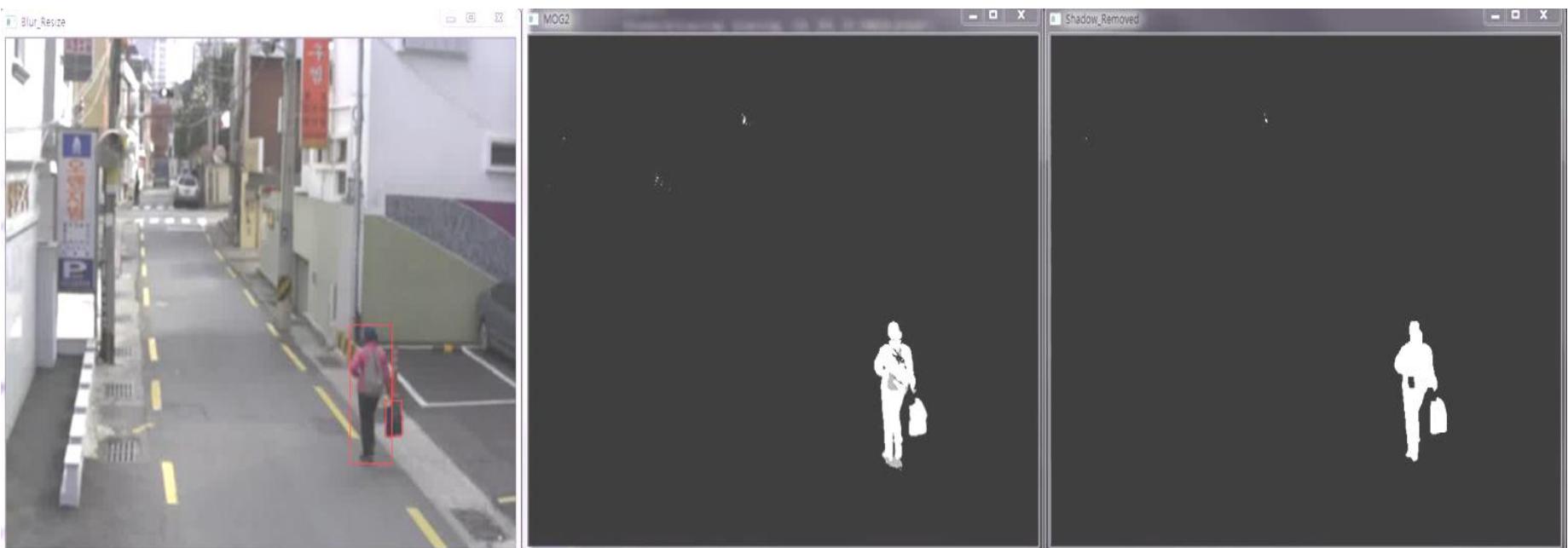
$| > Th$

Estimer l'arrière-plan à l'instant t

Soustraire l'arrière-plan estimé de l'image en cours.

Faire un seuillage en utilisant la différence absolue afin de trouver un masque du premier-plan

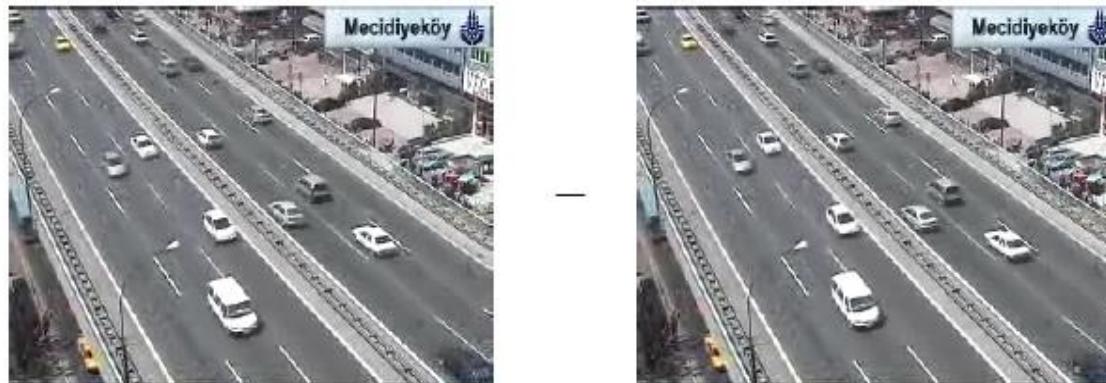
APPROCHES SIMPLES



DIFFÉRENCE D'IMAGES (FRAME DIFFERENCING)

L'équation de la soustraction de l'arrière-plan devient: $B(x, y, t) = I(x, y, t-1)$

$$|I(x, y, t) - I(x, y, t-1)| > Th$$



Tout dépend de la structure des objets, la vitesse, la cadence des images et le seuil global Th, cette approche peut ou ne pas être utile.

La précision de cette approche dépend de la vitesse du mouvement dans la scène. Des mouvements plus rapides peuvent nécessiter des seuils plus élevés.

DIFFÉRENCE D'IMAGES (FRAME DIFFERENCING)

Cette approche ne peut fonctionner que si tous les pixels de premier plan sont en mouvement et tous les pixels d'arrière-plan sont statiques.

$Th = 25$



$Th = 50$



$Th = 100$



$Th = 200$



DIFFÉRENCE D'IMAGES (FRAME DIFFERENCING)



FILTRE MOYEN (MEAN FILTER)

Dans ce cas l'arrière-plan est la moyenne des anciennes n images (frames)

$$B(x, y, t) = \frac{1}{n} \sum_{i=0}^{n-1} I(x, y, t - i)$$

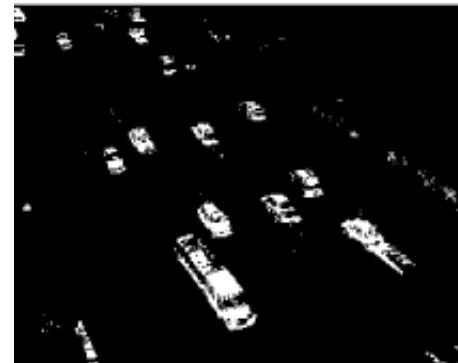
D'où $|I(x, y, t) - \frac{1}{n} \sum_{i=0}^{n-1} I(x, y, t - i)| > Th$

n est le nombre d'images précédentes prises pour faire la moyenne.

n dépend de la vitesse de la vidéo (nombre d'images par seconde dans la vidéo) et de la quantité de mouvement dans la vidéo.

Pour n=10

Arrière plan estimé



Masque du premier-plan

FILTRE MOYEN (MEAN FILTER)

Pour n=20

Arrière plan estimé



Masque du premier-plan



Pour n=50



FILTRE MÉDIAN:(MEDIAN FILTER)

supposant que l'arrière-plan est plus susceptible d'apparaître dans une scène, nous pouvons utiliser la médiane des n images précédentes comme modèle d'arrière-plan:

$$B(x, y, t) = \text{median}\{I(x, y, t - i)\}$$

$$|I(x, y, t) - \text{median}\{I(x, y, t - i)\}| > Th$$

alors que $i \in \{0, 1, \dots, n - 1\}$

FILTRE MÉDIAN:(MEDIAN FILTER)

N=10



N=20



N=50



Arrière-plan
estimé



Masque du
premier-plan



LA MOYENNE GLISSANTE (RUNNING AVERAGE)

Dans le cas où on cherche un modèle d'arrière-plan moyen, ceci peut être géré par une moyenne glissante:

$$B(x, y, t) = \frac{t-1}{t} B(x, y, t-1) + \frac{1}{t} I(x, y, t)$$

Plus généralement

$$B(x, y, t) = (1 - \alpha) B(x, y, t-1) + \alpha I(x, y, t)$$

α est le taux d'entraînement

Koller et al., *ICPR '94+ $\mu_t = M\mu_{t-1} + (1 - M)((1 - \alpha)\mu_{t-1} + \alpha I_t)$

Où M est le masque binaire du FG

COMPARAISON ENTRE LES DIFFÉRENTES MÉTHODES

Avantages

- Extrêmement facile pour l'implémenter
- Rapide en termes de temps d'exécution.
- Les arrière-plans estimés ne sont pas constants, ils changent au fur et à mesure.

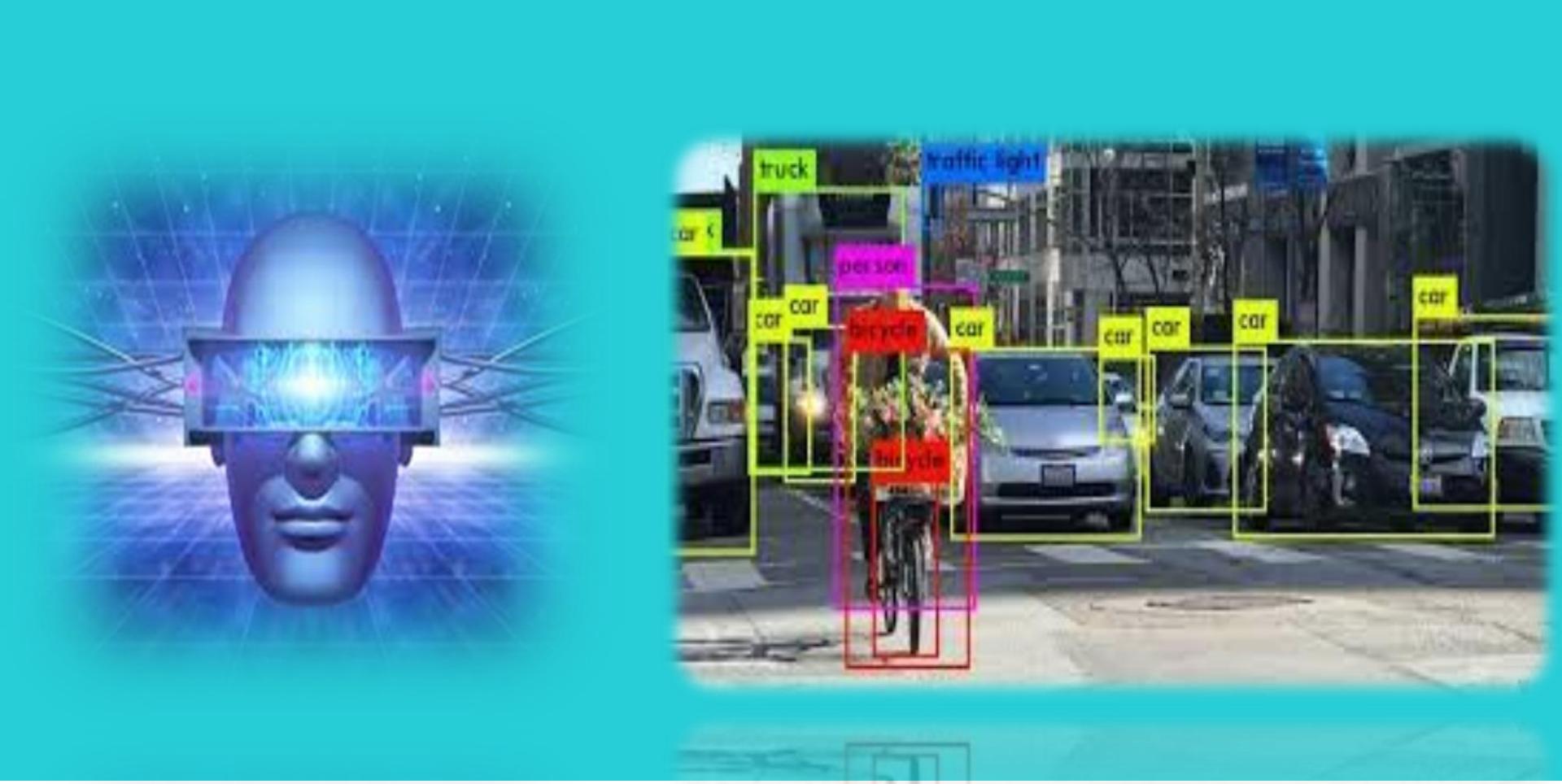
Inconvénients

- La précision de la différence d'images dépend de la vitesse des objets ainsi que la cadence de la vidéo.
- Les approches basées sur le filtre moyen et le filtre médian consomme beaucoup de mémoire
- Ces approches utilisent un seuillage globale Th , qui ne dépend pas du temps.

ATELIER2

En utilisant une vidéo enregistrée sur le disque dur ou capture à partir d'une web Cam. Réaliser une segmentation avant-plan/arrière-plan en utilisant:

- La différence d'images
- La dérivation temporelle
- La moyenne glissante (Running Average)
- Le filtre Médian
- Le filtre moyen



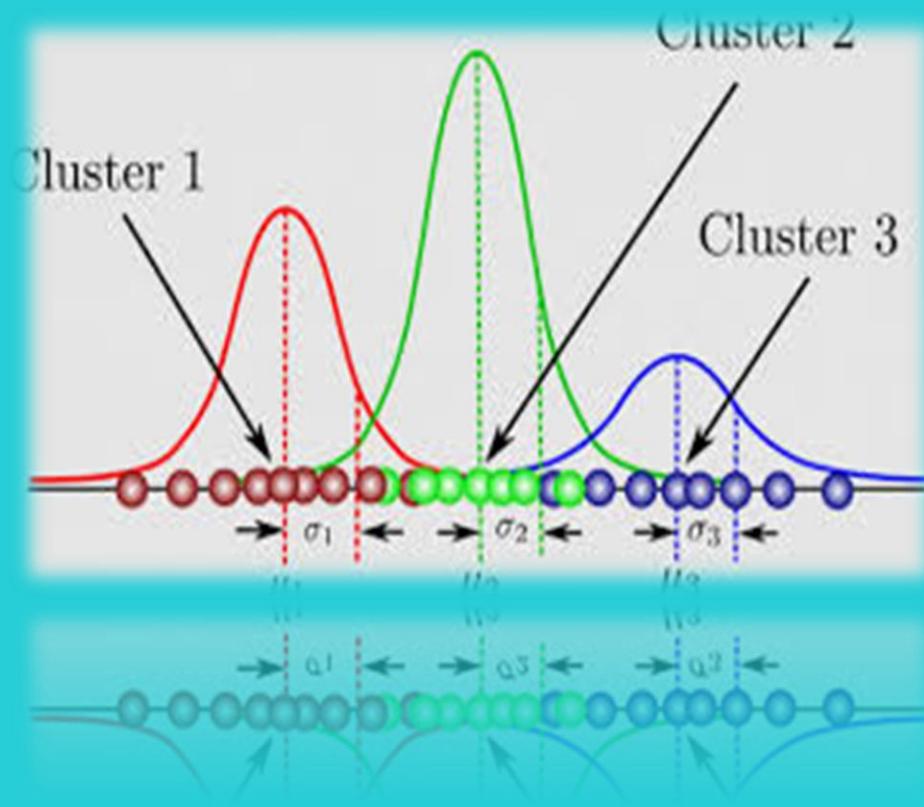
MÉTHODES AVANCÉES POUR LA DÉTECTION DES OBJETS

MACHINE LEARNING VS DEEP LEARNING

Les avancées actuellement en AI et en hardware, nous a permis de booster la recherche pour résoudre le problème de détection d'objet dans une vidéo.

Deux grandes approches existent dans la littérature:

- Approches basées sur Machine Learning (old school)
- Approches basées sur Deep Learning (new school)

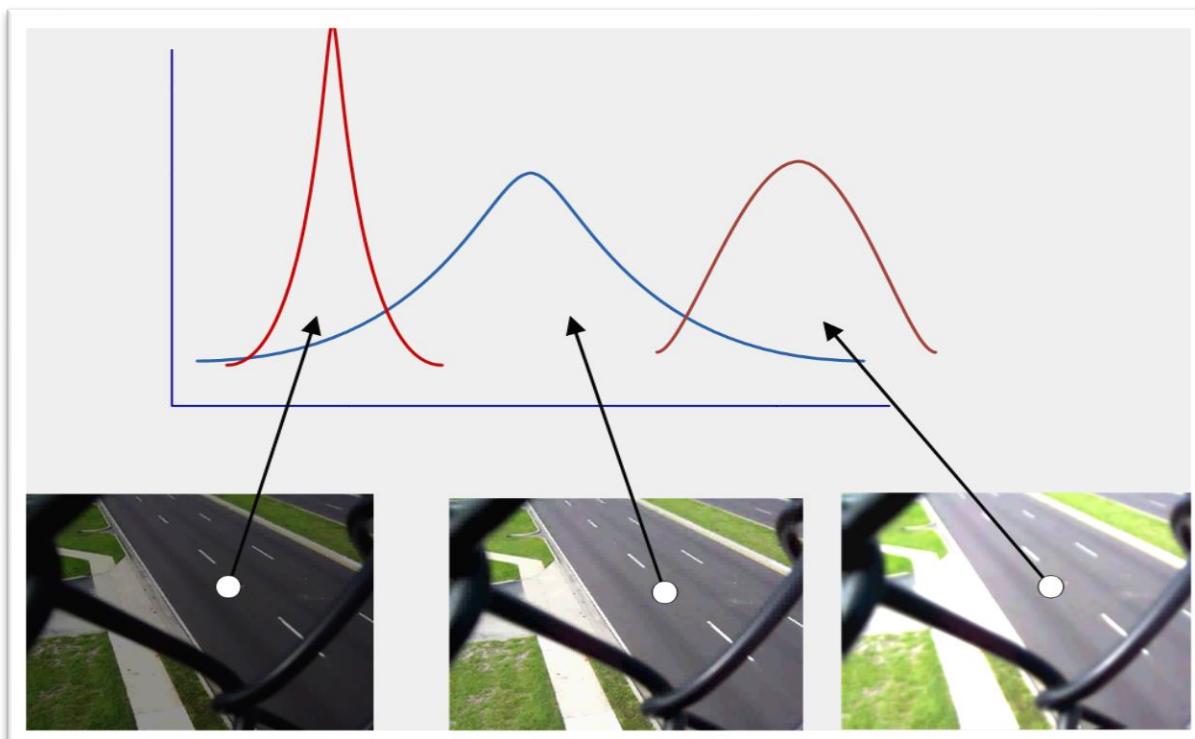


MÉLANGE DE GAUSSIENNES: CLUSTERING

ESTIMATION DE L'ARRIÈRE-PLAN: MÉLANGE DE GAUSSIENNES

Chaque pixel est modélisé avec un mélange de gaussiennes

Flexible pour gérer les variations en arrière-plan



ESTIMATION DE L'ARRIÈRE-PLAN: MÉLANGE DE GAUSSIENNES

Deux tâches effectuées en temps réel

- Apprendre le modèle de base
- La classification des pixels comme arrière-plan ou au premier plan

Apprendre le modèle de base

- Les paramètres de gaussiennes
 - Moyenne
 - Variance/covariance
 - Poids

Nombre de gaussiennes par pixel

ESTIMATION DE L'ARRIÈRE-PLAN: MÉLANGE DE GAUSSIENNES

Les valeurs d'un pixel particulier sont modélisées comme un mélange de gaussiennes adaptatives.

- Pourquoi le mélange?
 - Plusieurs surfaces apparaissent sur un pixel
- Pourquoi adaptatif?
 - Les conditions d'éclairage changent

À chaque itération, les gaussiennes sont évaluées en utilisant une simple heuristique pour déterminer celles qui sont les plus susceptibles de correspondre à l'arrière-plan.

Les pixels qui ne correspondent pas aux «gaussiennes d'arrière-plan » sont classés comme premier plan.

ESTIMATION DE L'ARRIÈRE-PLAN: MÉLANGE DE GAUSSIENNES

A chaque instant t , ce qui est connu à propos d'un pixel (x_0, y_0) , c'est son histoire

$$\{X_1, \dots, X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\}$$

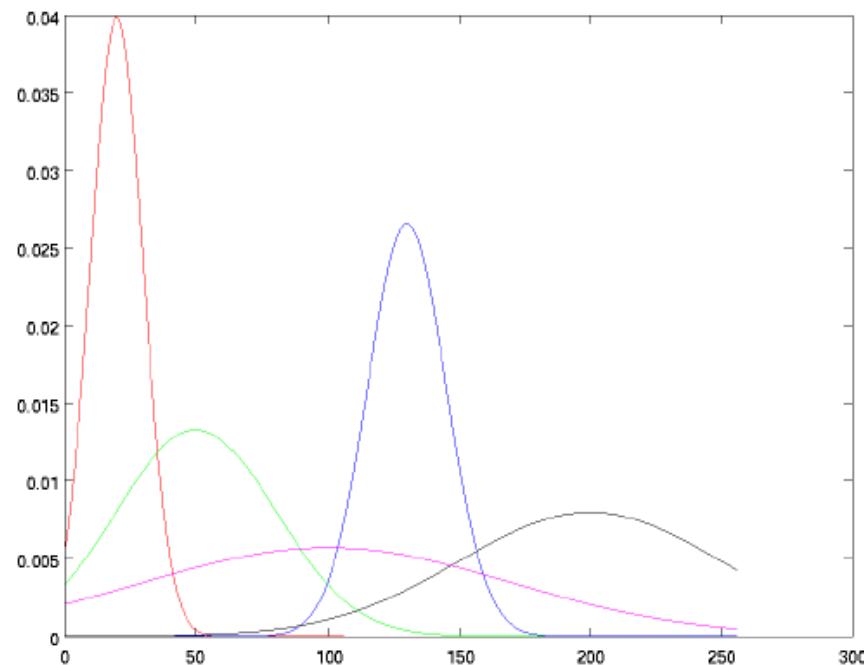
Cette histoire est modélisée par une distribution d'un mélange de K gaussiennes

$$P(X_t | \theta_t) = \sum_{i=1}^K \pi_i * N(X_t | \theta_{i,t}) = \sum_{i=1}^K \pi_{i,t} * \frac{1}{\sqrt{2\pi}^d \sqrt{\det(\Sigma_{i,t})}} e^{-\frac{1}{2}(X_t - \mu_{i,t})^T \Sigma_{i,t}^{-1} (X_t - \mu_{i,t})}$$

$$\theta_t = \{(\mu_{i,t})_{i=1..K}, (\Sigma_{i,t})_{i=1..K}, (\pi_{i,t})_{i=1..K}\}$$

ESTIMATION DE L'ARRIÈRE-PLAN: MÉLANGE DE GAUSSIENNES

Si on suppose qu'on a une succession des images niveau de gris et que $K = 5$, l'histoire d'un pixel ressemble à cette figure



ADAPTATION DU MODÈLE

Une approximation en ligne K-means est utilisée pour mettre à jour les gaussiennes

Si une nouvelle valeur de pixel, x_{t+1} , peut être associée à l'une des gaussiennes existantes (à moins de $2,5\sqrt{\det(\Sigma)}$), les gaussiennes $\mu_{i,t+1}$ et $\Sigma_{i,t+1}$ sont mises à jour comme suit:

$$\mu_{i,t+1} = (1 - \rho)\mu_{i,t} + \rho x_{t+1}$$

$$\Sigma_{i,t+1} = (1 - \rho)\Sigma_{i,t} + \rho(x_{t+1} - \mu_{i,t+1})^2$$

Avec $\rho = \alpha N(x_{t+1} | \theta_{i,t}) + \Sigma_{i,t+1}$ et α est un taux d'apprentissage.

Les poids antérieurs de tous les gaussiennes sont ajustés comme suit:

$$\pi_{i,t+1} = (1 - \alpha)\pi_{i,t} + M_{i,t+1}$$

Avec $M_{i,t+1} = 1$ pour les pixels associé à l'une des gaussiennes, sinon $M_{i,t+1} = 0$

ESTIMATION DE L'ARRIÈRE-PLAN

Heuristique : les gaussiennes avec les poids les plus élevés et une variance minimale correspondent à l'arrière-plan.

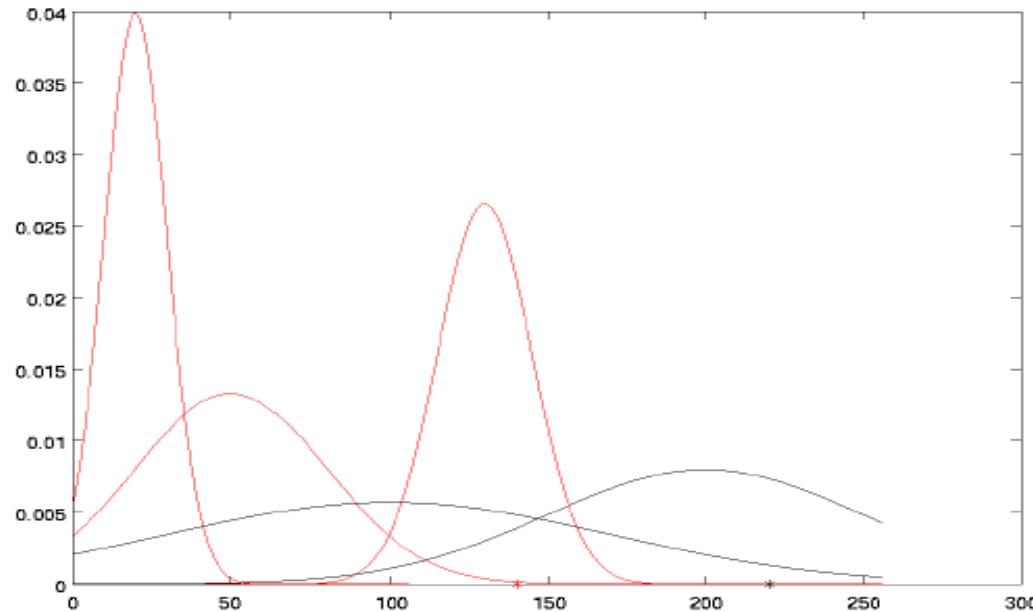
Les gaussiennes sont ordonnées par la valeur de $\pi / \det(\Sigma)$ (un poids élevé et une petite variance donneront une valeur élevée).

Par conséquent, on va choisir les B premières distributions comme modèle d'arrière-plan

$$B = \arg \min_b (\sum_{i=1}^b \pi_i > T)$$

Où T est la partie minimale de l'image qui devrait être en arrière-plan

ESTIMATION DE L'ARRIÈRE-PLAN

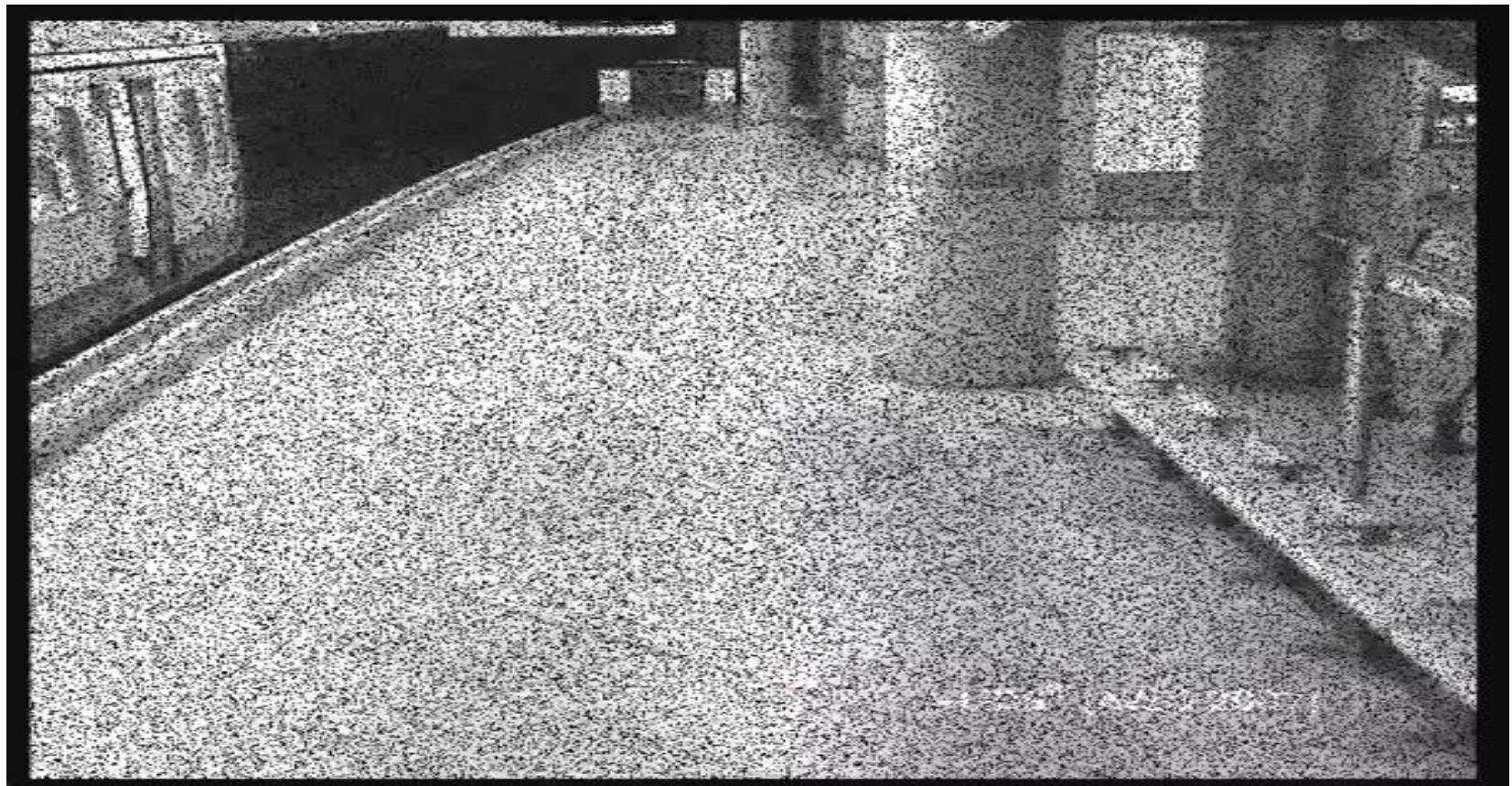


Après l'estimation du modèle d'arrière-plan, les distributions rouges deviennent le modèle d'arrière-plan et les distributions noires sont considérées comme étant de premier plan.

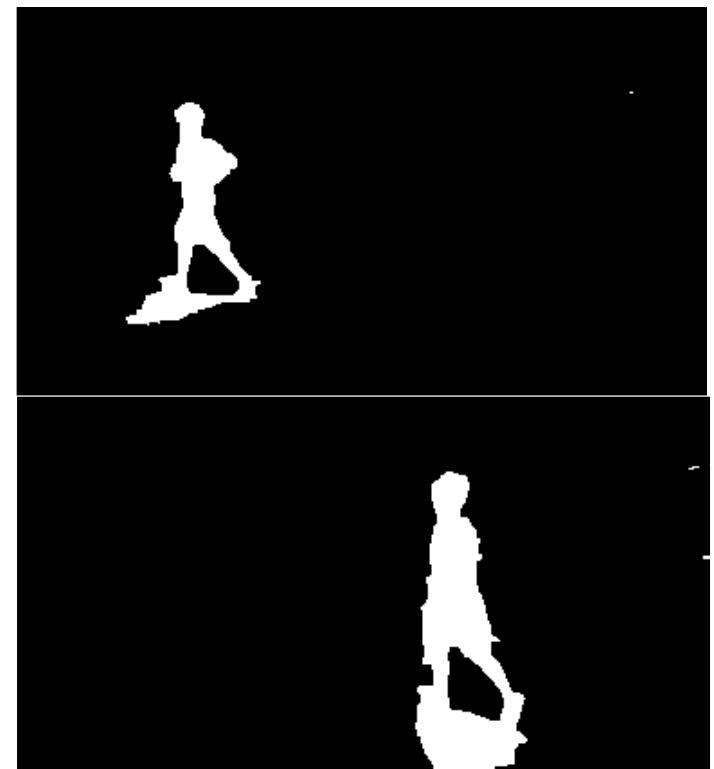
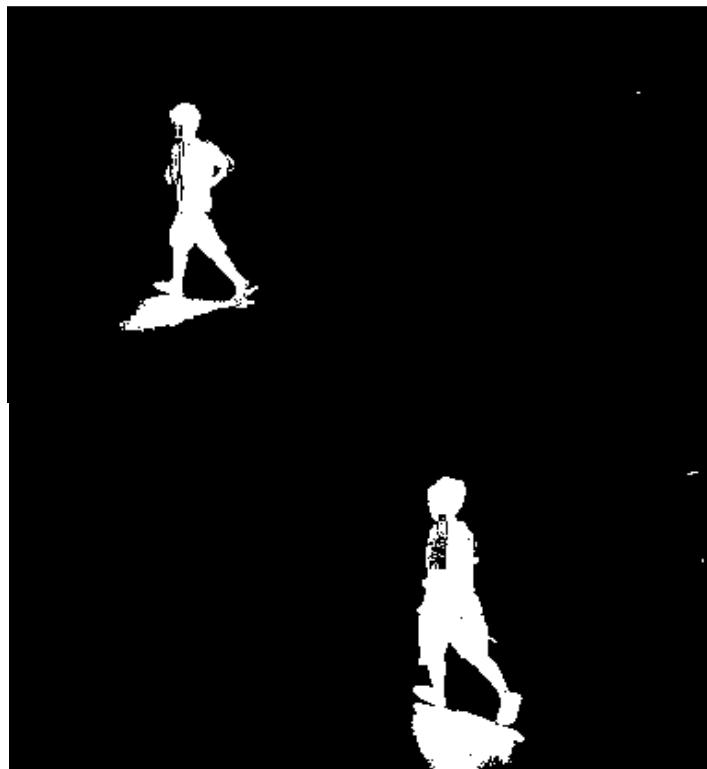
Exemple de la soustraction de l'arrière-plan en utilisant GMM



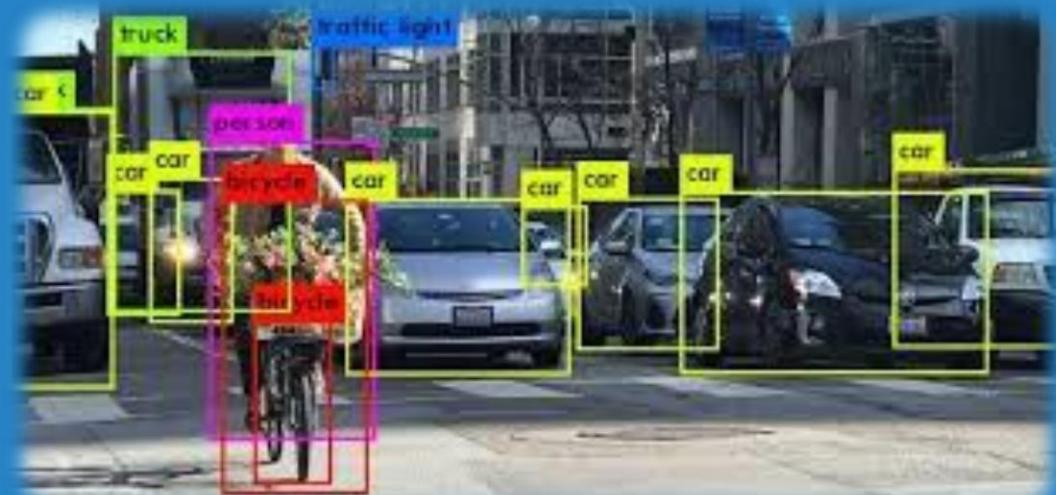
EXEMPLE DE LA SOUSTRACTION DE L'ARRIÈRE-PLAN EN UTILISANT GMM



Exemple de la soustraction de l'arrière-plan en utilisant GMM



On utilise la fermeture morphologique comme post traitement pour se débarrasser du bruit résultant



MODÈLES DEEP LEARNING POUR LA DÉTECTION D'OBJETS

MODÈLES DEEP LEARNING POUR LA DÉTECTION D'OBJETS

Le problème de détection d'objets dans une vidéo est de plus en plus résolu, grâce aux avancées de la vision par ordinateur et de Deep Learning.

Les modèles entraînés sur de grands datasets, accessibles au public, simplifient davantage cette tâche.

Les chercheurs se concentrent plus sur d'autres sujets plus prometteurs, tels que la narration visuelle, la segmentation et le suivi d'objets, etc.

QU'EST-CE QUE LA DÉTECTION D'OBJET ?

La détection d'objets est le domaine de la vision par ordinateur qui traite de la localisation et de la classification des objets contenus dans une image ou une vidéo.

Pour le dire simplement : la détection d'objets revient à dessiner des cadres de délimitation (bounding boxes) autour des objets détectés qui nous permettent de les localiser dans une scène donnée (ou comment ils se déplacent à travers elle).

DÉTECTION D'OBJETS VS CLASSIFICATION D'IMAGES

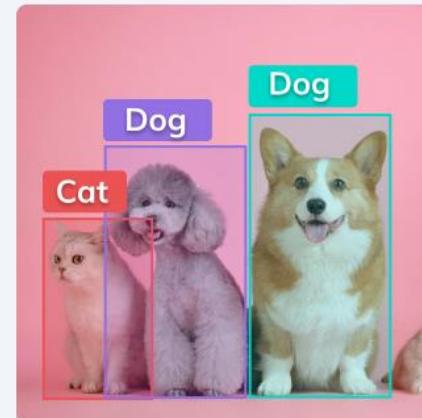
La classification d'image envoie une image entière via un classifieur (tel qu'un réseau neuronal profond). Les classifieurs prennent en considération l'image entière mais ne vous disent pas où l'objet apparaît dans l'image.

La détection d'objet est légèrement plus avancée, car elle crée une boîte englobante autour de l'objet classifié.

Classification



Detection



DÉTECTION D'OBJET VS SEGMENTATION D'IMAGE

La segmentation d'image est le processus de définition des pixels d'une classe d'objets qui se trouvent dans une image.

La segmentation sémantique des images marquera tous les pixels appartenant à cet objet, mais ne définira pas les limites de chaque objet.

La détection d'objet ne segmentera pas l'objet, mais définira clairement l'emplacement de chaque instance d'objet individuel avec une boîte.

La combinaison de la segmentation sémantique avec la détection d'objets conduit à la segmentation d'instance, qui détecte d'abord les instances d'objet, puis segmente chacune dans les boîtes détectées (appelées dans ce cas régions d'intérêt)

DÉTECTION D'OBJET VS SEGMENTATION D'IMAGE

Object Detection + Semantic Segmentation
= Instance Segmentation



Object detection



Semantic Segmentation



Instance Segmentation

TYPES ET MODES DE DÉTECTION D'OBJETS

Avant le progrès de Deep Learning, la quasi-totalité des méthodes pour la détection d'objets était effectuée à l'aide de techniques d'apprentissage automatique classiques. Les plus courants comprenaient la technique de détection d'objets viola-jones, les transformées de caractéristiques invariantes à l'échelle (SIFT) et l'histogramme des gradients orientés (HOG).

Ceux-ci détecteraient un certain nombre de caractéristiques communes à travers l'image et classeraient leurs clusters à l'aide d'une régression logistique, d'histogrammes de couleurs ou de Random Forest.

LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS

Les algorithmes les plus importants de détection d'objets en un étage (one-stage)

- YOLO (2016)
- SSD (2016)
- RetinaNet (2017)
- YOLOv3 (2018)
- YOLOv4 (2020)
- YOLOR (2021)

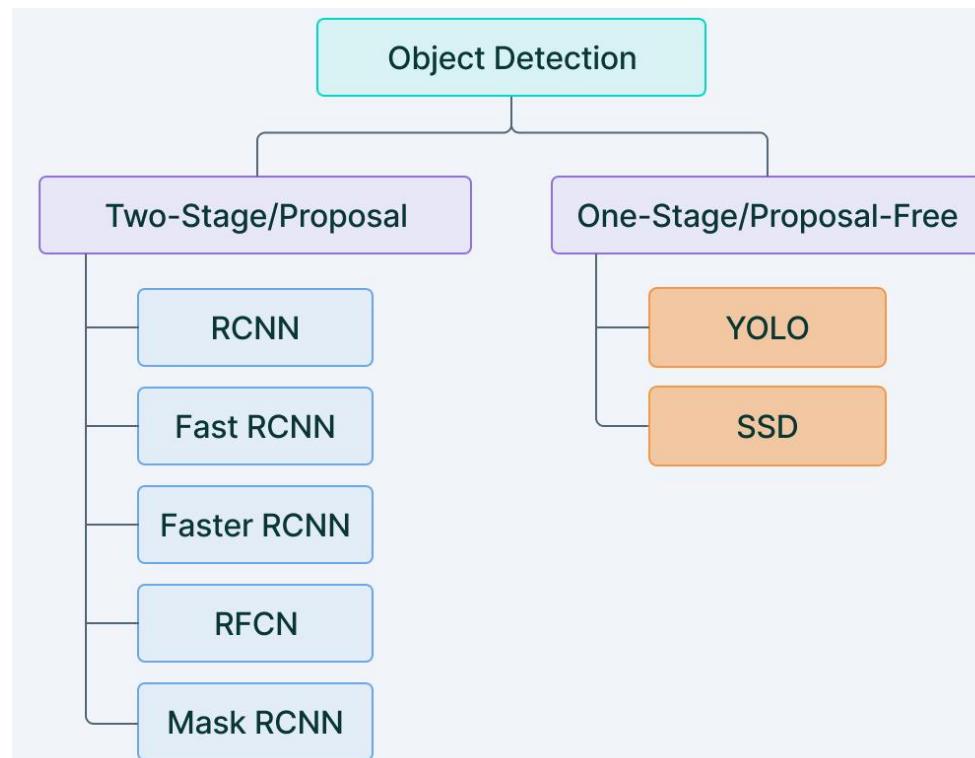
Les algorithmes les plus importants de détection d'objets en deux étages (two-stage)

- RCNN et SPPNet (2014)
- Fast RCNN et Faster RCNN (2015)
- Mask R-CNN (2017)
- Pyramid Networks/FPN (2017)
- G-RCNN (2021)

COMMENT FONCTIONNE LA DÉTECTION D'OBJETS

La détection d'objets est généralement classée en 2 catégories :

- DéTECTEURS d'objets à un étage.
- DéTECTEURS d'objets à deux étages.



DÉTECTEURS D'OBJETS À UN ÉTAGE

Un détecteur à un étage supprime le processus d'extraction RoI et classe et régresse directement les boîtes d'ancrage candidates. Exemples : famille YOLO (YOLOv2, YOLOv3, YOLOv4 et YOLOv5) CornerNet, CenterNet et autres.

ARCHITECTURE YOLO

YOLO est une architecture de détection d'objets simplement appelée YOU ONLY LOOK ONCE. Cela implique l'utilisation d'un seul réseau neuronal entraîné de bout en bout (e2e) pour prendre une image en entrée et prédire directement les boîtes englobantes et les étiquettes de classe pour chaque boîte englobante. YOLO est un détecteur typique à un étage.

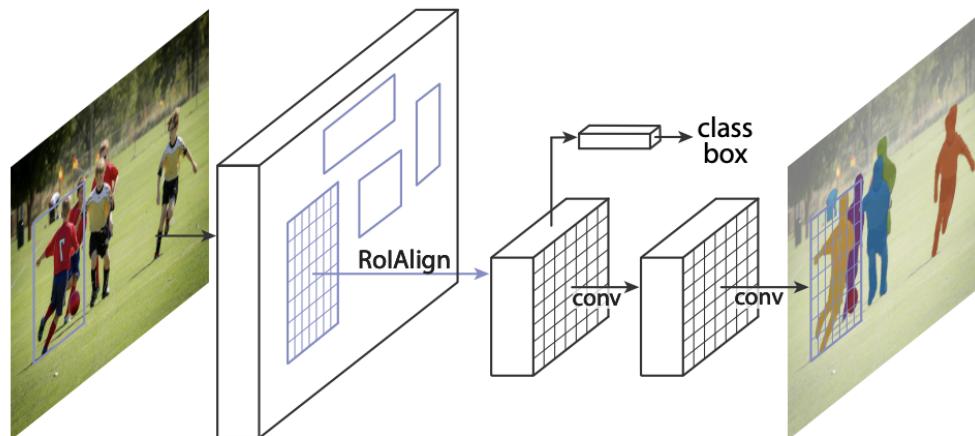
DÉTECTEURS D'OBJETS À DEUX ÉTAGES

Les détecteurs à deux étages divisent la tâche de détection d'objet en deux étapes : extraire les RoI (région d'intérêt), puis classer et régresser les RoI.

Des exemples d'architectures de détection d'objets orientées en 2 étapes incluent R-CNN, Fast-RCNN, Faster-RCNN, Mask-RCNN et autres.

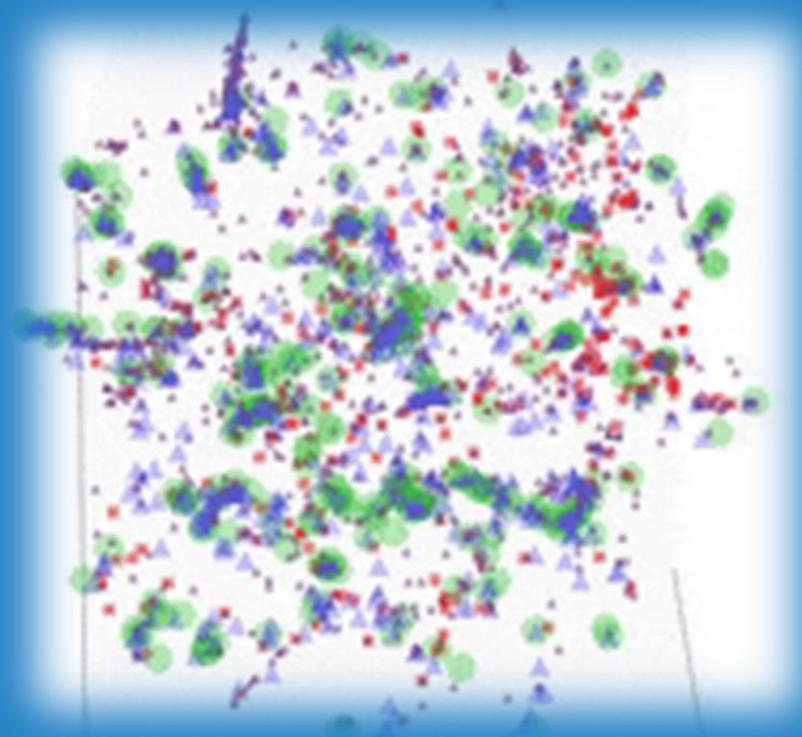
MASK R-CNN

Le masque R-CNN est une technique typique de segmentation d'instance d'objet pour la détection d'objets. Cette architecture est une extension de Faster R-CNN en ajoutant une branche pour prédire les masques de segmentation sur chaque RoI, en parallèle avec la branche existante pour la classification et la régression de la boîte englobante. La branche de masque est un petit FCN appliquée à chaque RoI, prédisant un masque de segmentation d'une manière pixel à pixel.



ATELIER 3

En utilisant une vidéo enregistrée sur le disque dur ou capture à partir d'une web Cam. Réaliser une segmentation avant-plan/arrière-plan en utilisant la méthode MOG, MOG2, GMG.

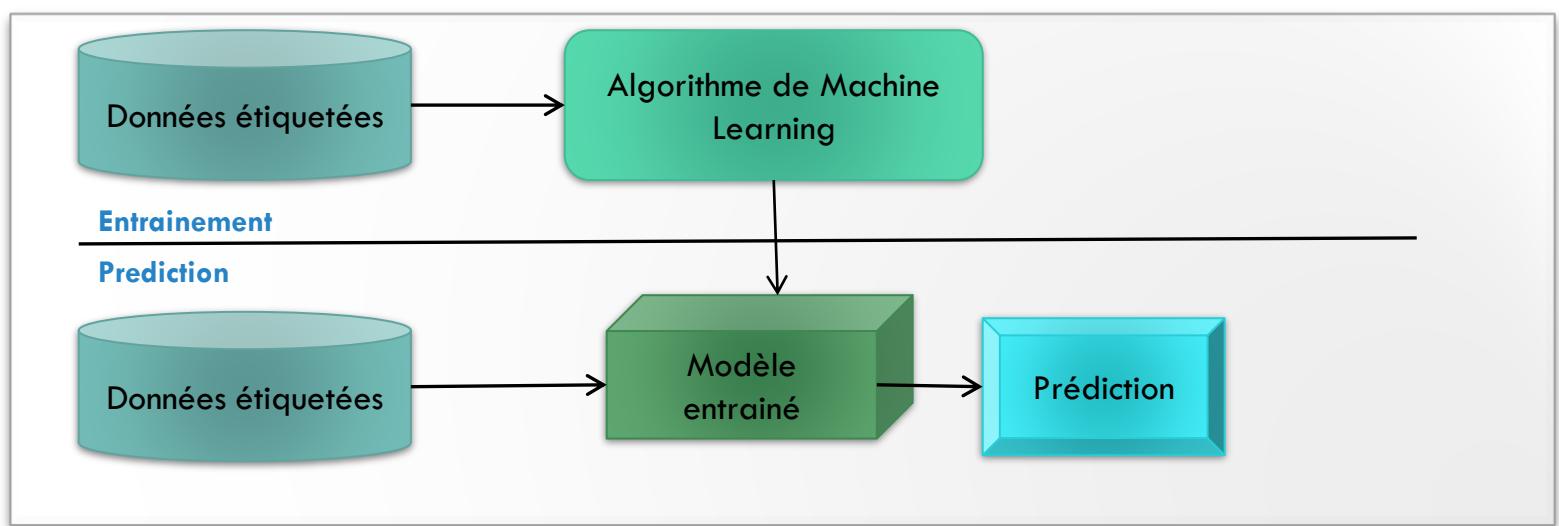


EXTRACTION MANUELLE DES CARACTÉRISTIQUES DANS UNE SÉQUENCE VIDÉO

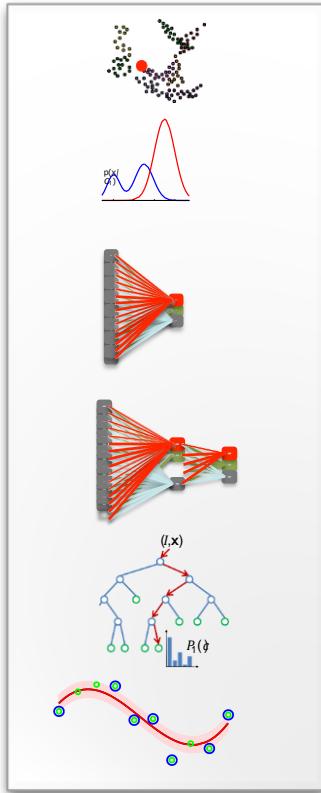
Pr. MAHRAZ Med Adnane

LES BASES DE MACHINE LEARNING

Machine learning est un domaine d'informatique qui donne aux ordinateurs la possibilité d'apprendre sans être explicitement programmé



ALGORITHMES DE MACHINE LEARNING UTILISÉS EN VISION



KNN (k plus proches voisins)

Classificateurs génératifs linéaires (Bayésiens)

Classificateurs discriminatifs linéaires

Réseaux de neurones (profonds)

Arbres de décision

SVM (Support Vector Machines)

EXTRACTION MANUELLE DES CARACTÉRISTIQUES DANS UNE VIDÉO

Spatiales

- SIFT
- SURF
- Haar features
- HOG
- LBP
- Harris
- ...

Spatio-temporelles

- HOG-TOP
- LBP-TOP
- HOG-HOF
- Harris3D
- ...

VECTEURS DE CARACTÉRISTIQUES

Invariance aux changements d'échelle

Invariance aux rotations

Invariance aux changements d'illumination.



DESCRIPTEUR SIFT

Scale-invariant feature
transform

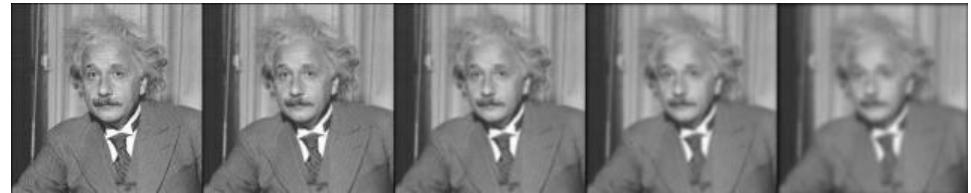
SIFT : ESPACE D'ÉCHELLES

Pyramide d'images convoluées par filtre gaussien $G(x, y, \sigma)$ avec un paramètre σ croissant (selon la suite géométrique de paramètre k) :

$$k^n = 2^p$$

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

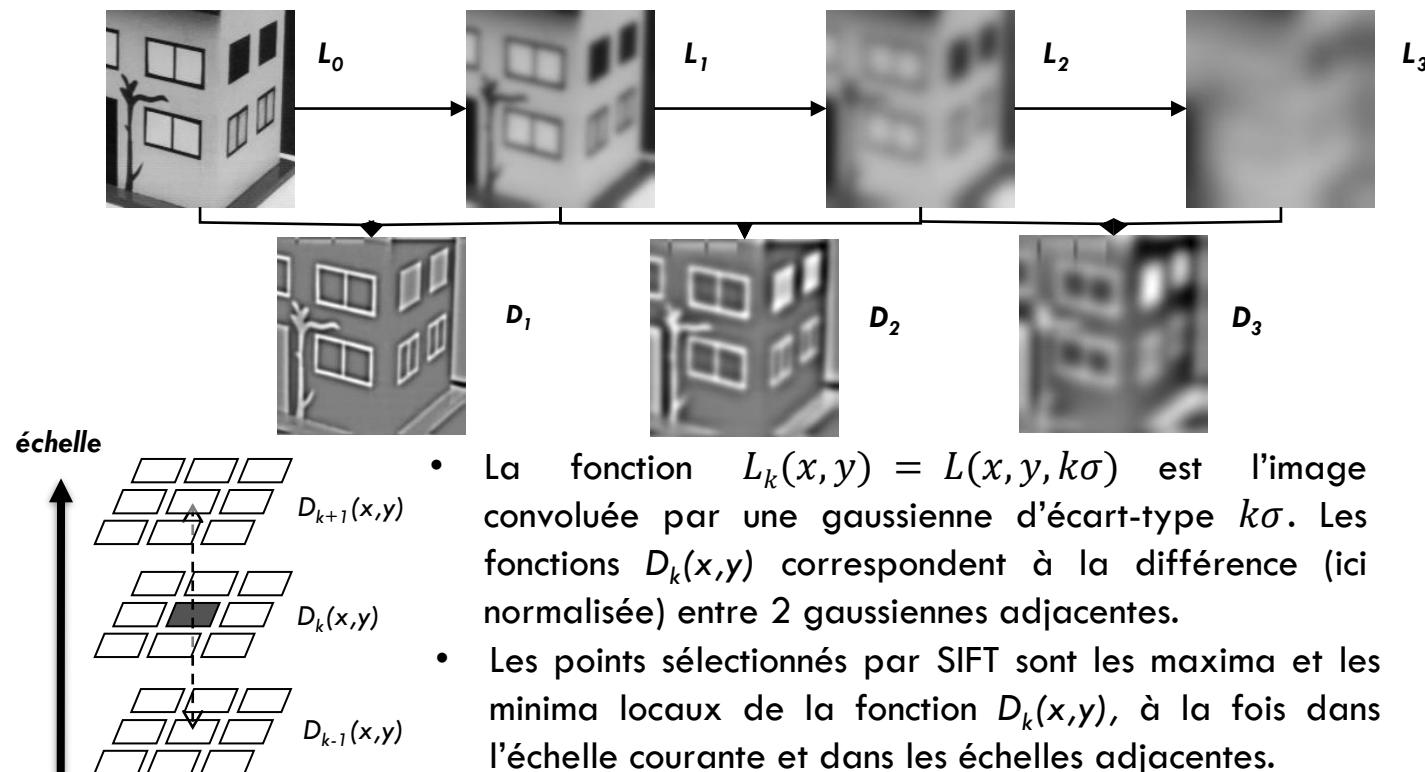
$$L(x, y, k^p \sigma)$$
$$p \in [0, 4]$$



$$D(x, y, \sigma) = L(x, y, k^p \sigma) - L(x, y, k^{p-1} \sigma)$$
$$p \in [1, 4]$$



DÉTECTEUR SIFT : EXTREMA DANS L'ESPACE D'ÉCHELLE



SIFT : CALCUL DE L'ORIENTATION

Histogramme des orientations dans un voisinage $V(x_k, y_k)$ du point clé à l'échelle σ où il est détecté, i.e.

Construction de l'histogramme :

Chaque pixel (x, y) de $V(x_k, y_k)$

- vote pour orientation
- avec poids

$$L(x, y) = G(x, y, \sigma) * I(x, y)$$

$$\tan^{-1} \left(\frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)} \right) \quad (\text{Orientation du gradient})$$

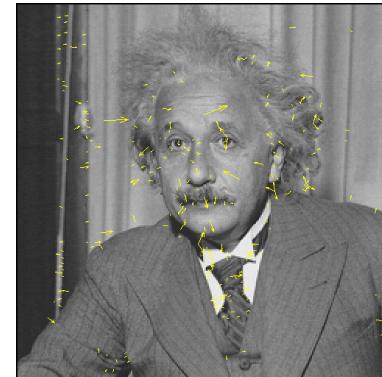
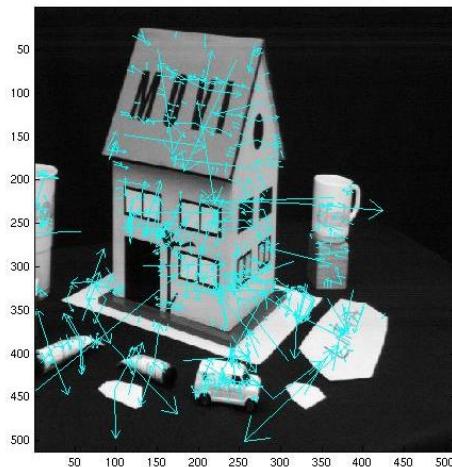
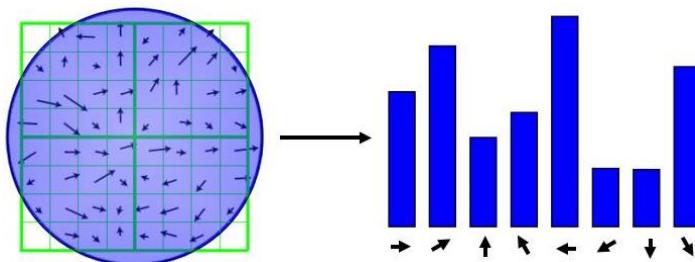
$$\sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \times \frac{2}{9\pi\sigma^2} \exp \left\{ -\frac{(x - x_k)^2 + (y - y_k)^2}{(9/2)\sigma^2} \right\}$$

les pixels distants ou de
faible gradient ne
'comptent' pas vraiment

(Norme du gradient)

(Pondération gaussienne 1.5σ)

SIFT : CALCUL DE L'ORIENTATION



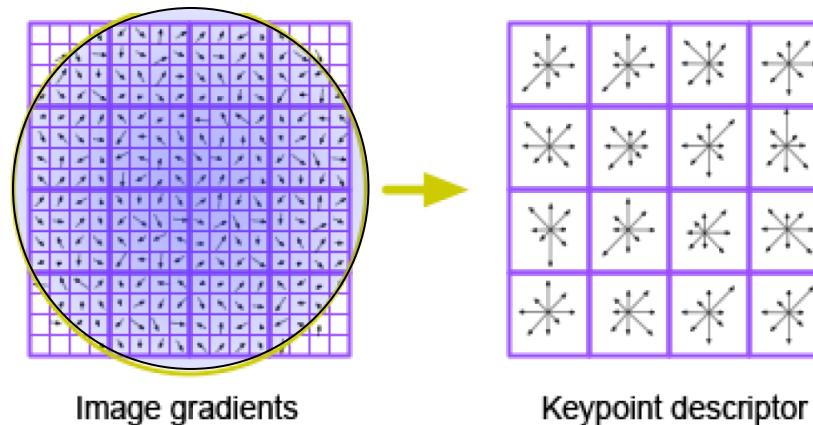
Les points d'intérêts SIFT : la direction de la flèche représente la direction θ et sa longueur à l'échelle σ associée.

SIFT : CALCUL DU DESCRIPTEUR

Voisinage 16×16 pixels autour du point clé, divisé en sous-blocs 4×4

Calcul des histogrammes (sur 8 bins, chaque échantillon étant pondéré par la norme du gradient et la fonction gaussienne) d'orientation sur chacun des sous-voisinages

- Descripteur contient 16 histogrammes de 8 bins \rightarrow dimension = 128



Rq : Normalisation du vecteur (norme = 1) + seuillage des composantes de valeur inférieure à 0.2 + renormalisation



DESCRIPTEUR LBP

Local Binary Pattern

LOCAL BINARY PATTERN

Local Binary Pattern(LBP) a été initialement conçu pour les textures

Il est simple et très efficace.

LBP a été décrit pour la première fois en 1994 et s'est depuis avéré être une vecteur de caractéristiques puissant pour la classification des textures.

LOCAL BINARY PATTERN ET LES OPÉRATEURS DE CONTRASTE

Exemple de calcul du LBP dans un voisinage 3x3 :

exemple seuillage poids

6	5	2
7	6	1
9	8	7

1	0	0
1	1	1
1	1	1

1	2	4
128	32	16
64	32	16

Propriétés importantes:

- LBP est invariant à tout changement de niveau de gris monotone
- simplicité de calcul

Pattern = 11110001

LBP = $1 + 16 + 32 + 64 + 128 = 241$

LOCAL BINARY PATTERN: EXEMPLE

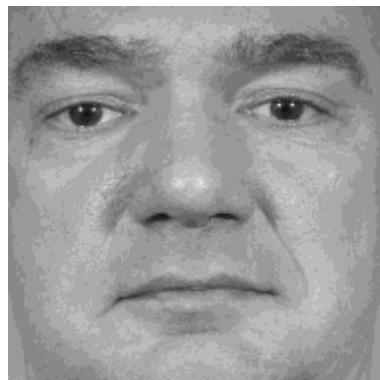
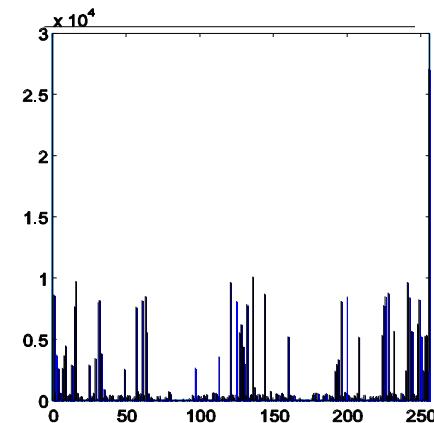


Image d'entrée

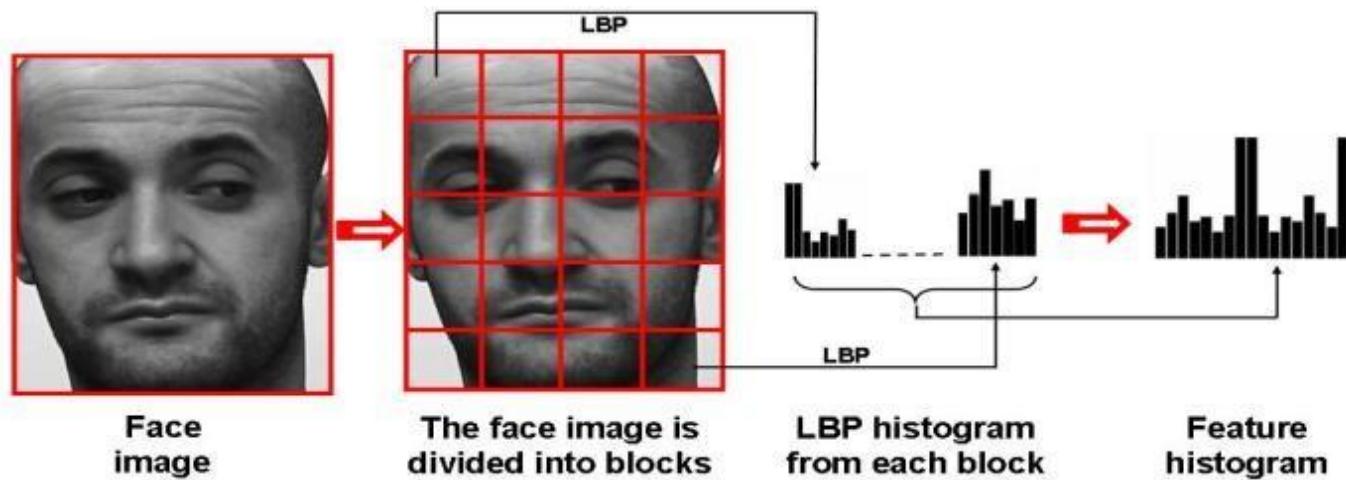


Image LBP

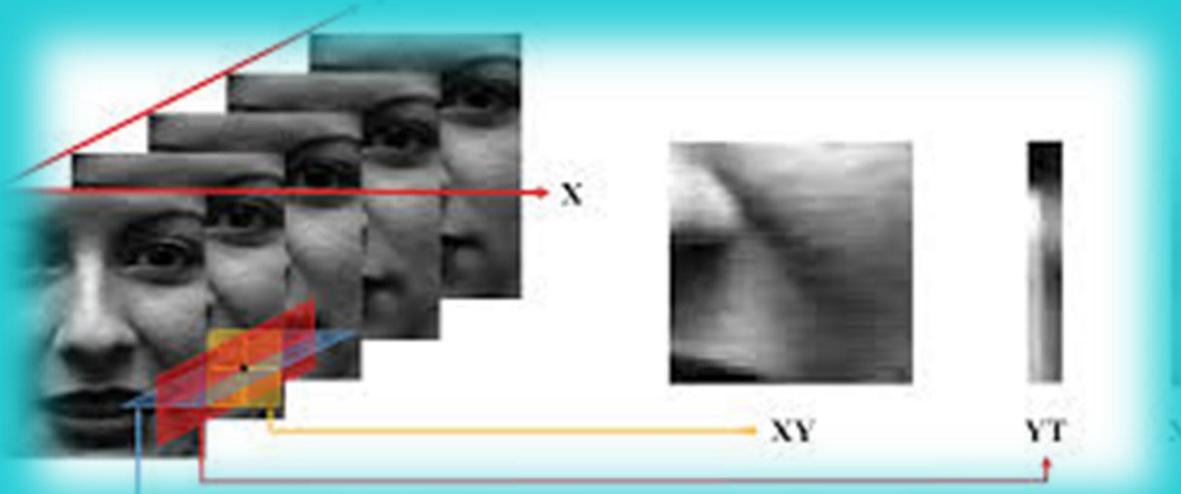


Histogramme LBP

DESCRIPTION DU VISAGE PAR LBP



Ahonen T, Hadid A & Pietikäinen M (2006) Face description with local binary patterns: application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(12):2037-2041



DESCRIPTEUR LBP-TOP: EXTENSION DE LBP

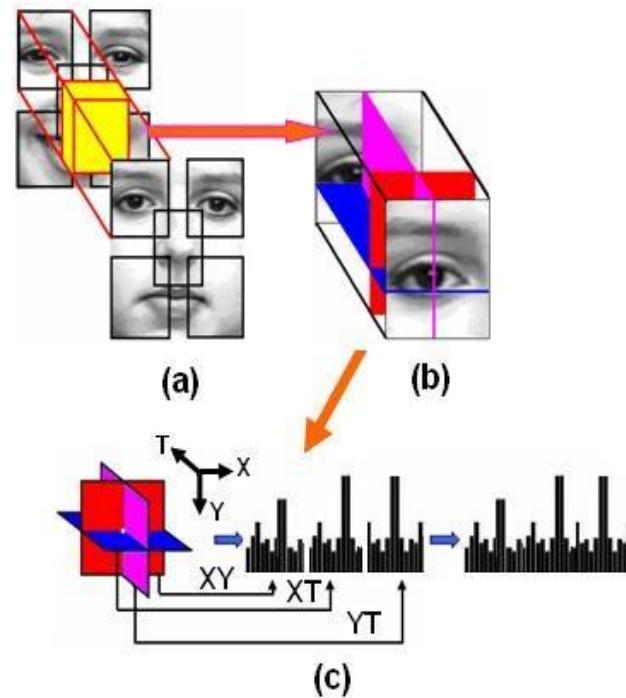
Local Binary Pattern-Three
Orthogonal Planes

LBP THREE ORTHOGONAL PLANES (LBP-TOP)

Extension de LBP dans les trois plans orthogonaux (TOP)

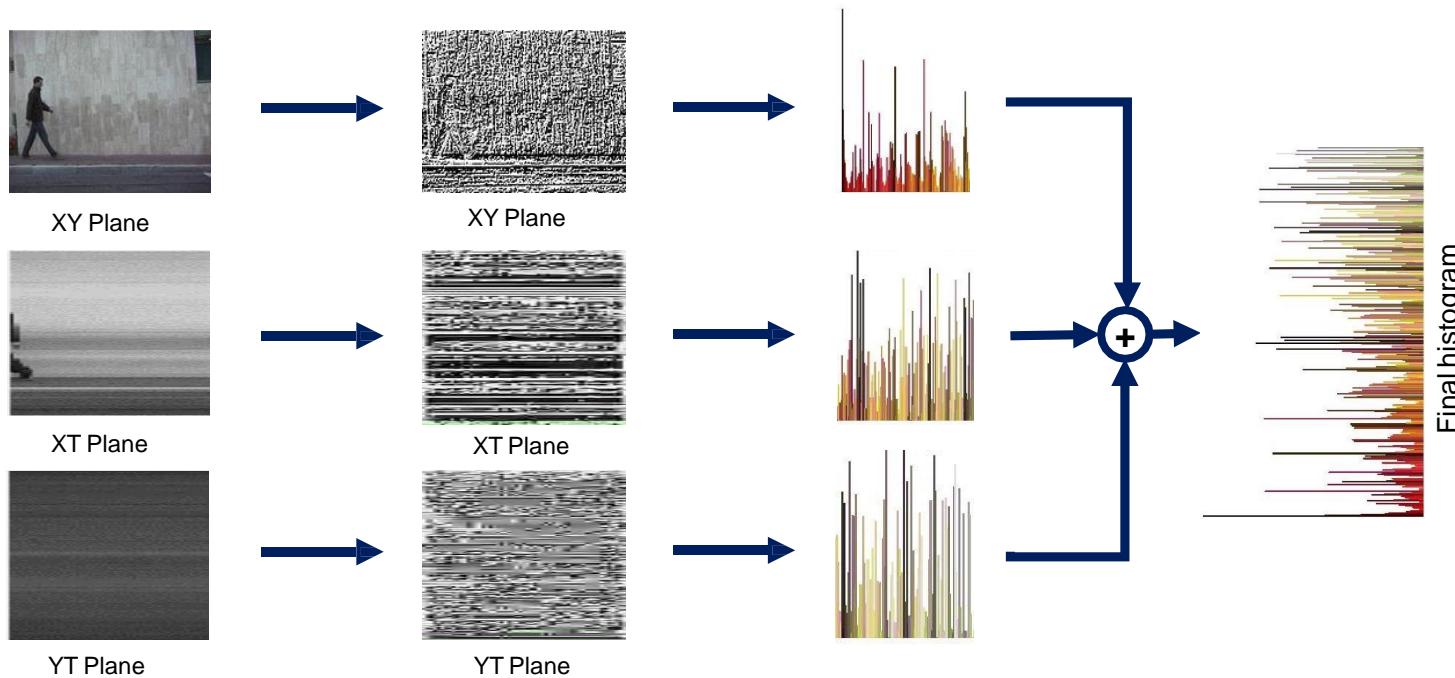
Encode la forme et le mouvement sur trois plans orthogonaux (XY, XT et YT)

Calculer l'occurrence de différents histogrammes de plan pour former l'histogramme final ($H = h_{XY} \cdot h_{XT} \cdot h_{YT}$)

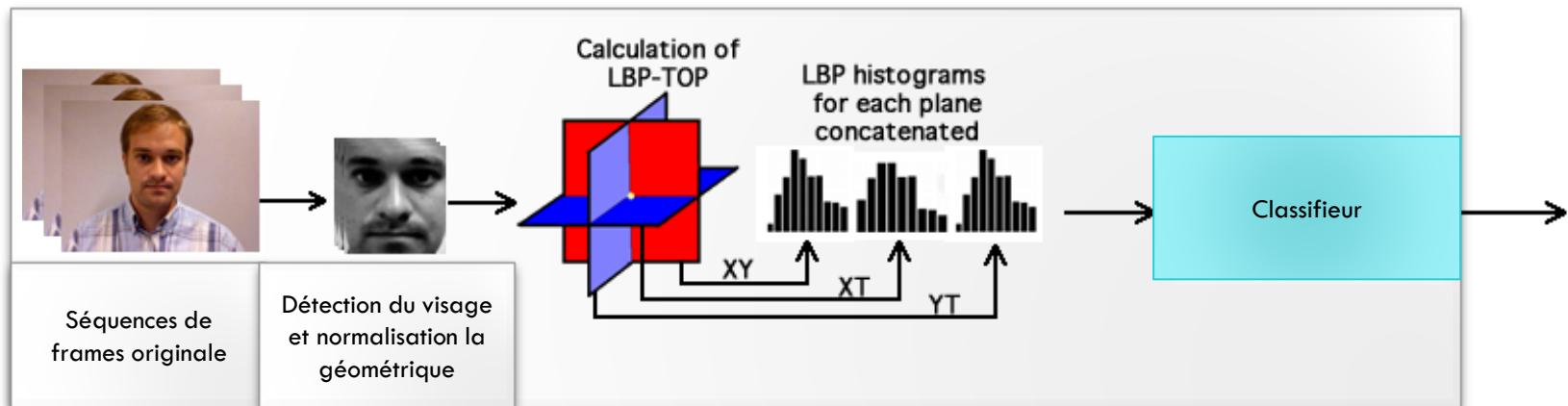


LBP - TOP $P_{XY}P_{XT}P_{YT}$ 12

LBP THREE ORTHOGONAL PLANES (LBP-TOP)



ANALYSE DE LA TEXTURE FACIALE: SPATIALE ET TEMPORELLE



CLASSIFICATION D'ÂGE À BASE D'UNE VARIANTE DE LBP

Concaténation des signes de LBP et les histogramme de magnitude CLBP_S_M

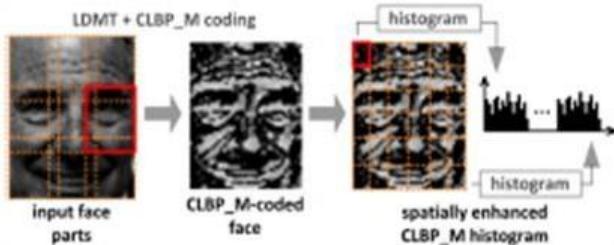


Figure. LBP magnitude histogram.

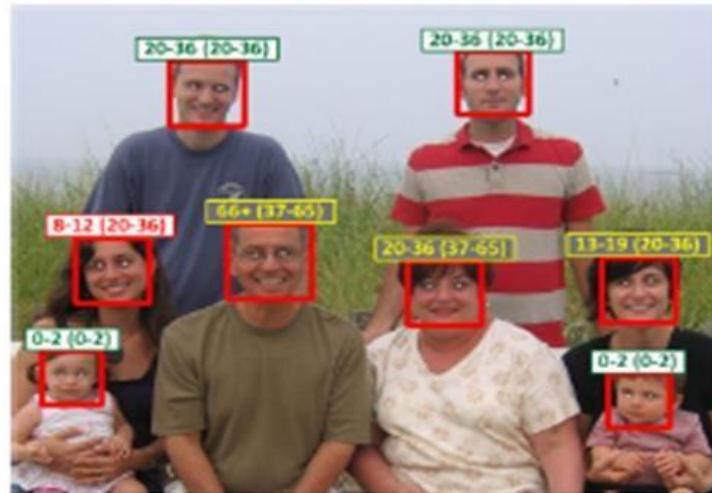


Figure. Examples of estimated age categories (ground truth in parentheses).

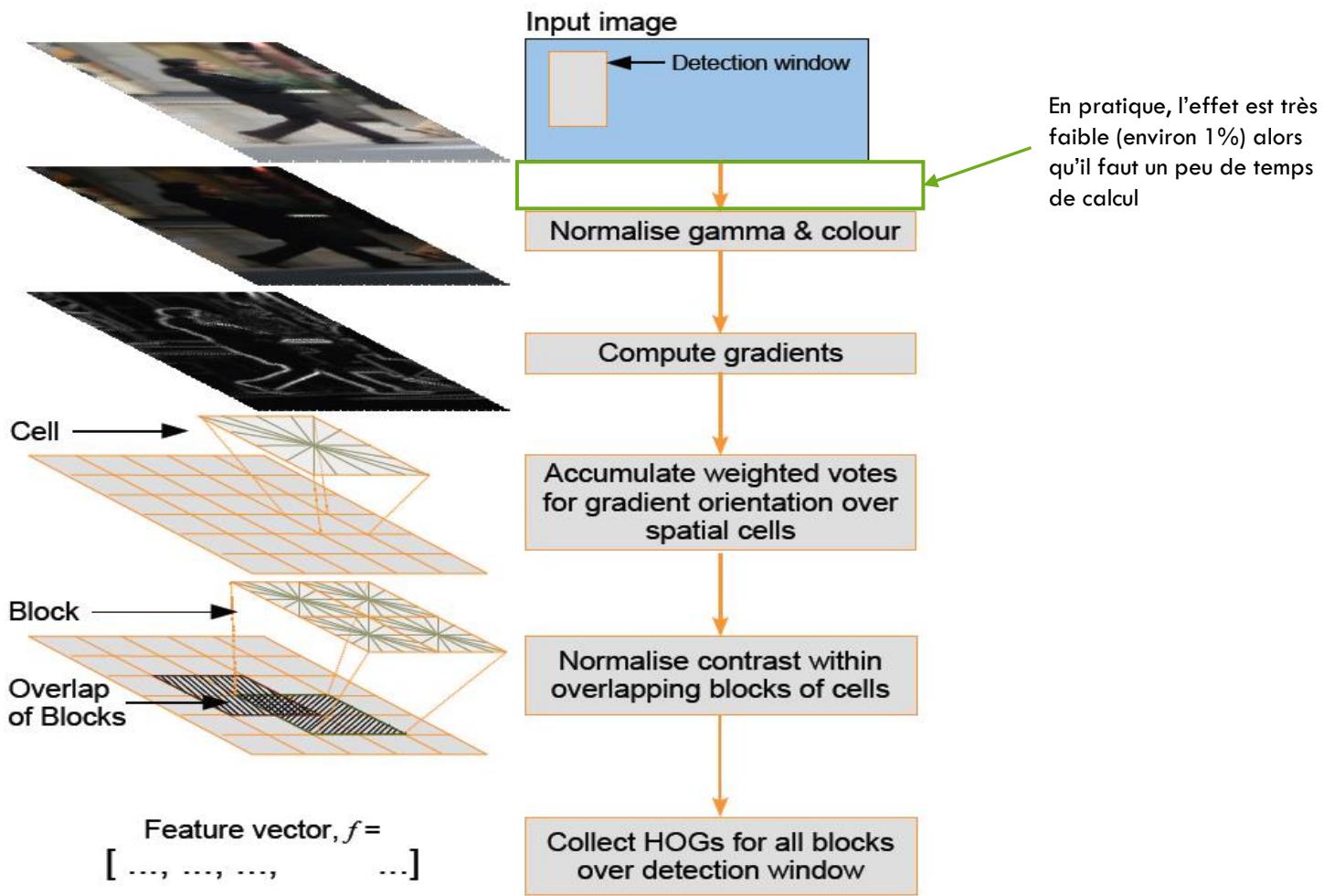
Ylioinas J, Hadid A & Pietikäinen M (2012) Age classification in unconstrained conditions using LBP variants. Proc. 21st International Conference on Pattern Recognition (ICPR 2012), Tsukuba, Japan.



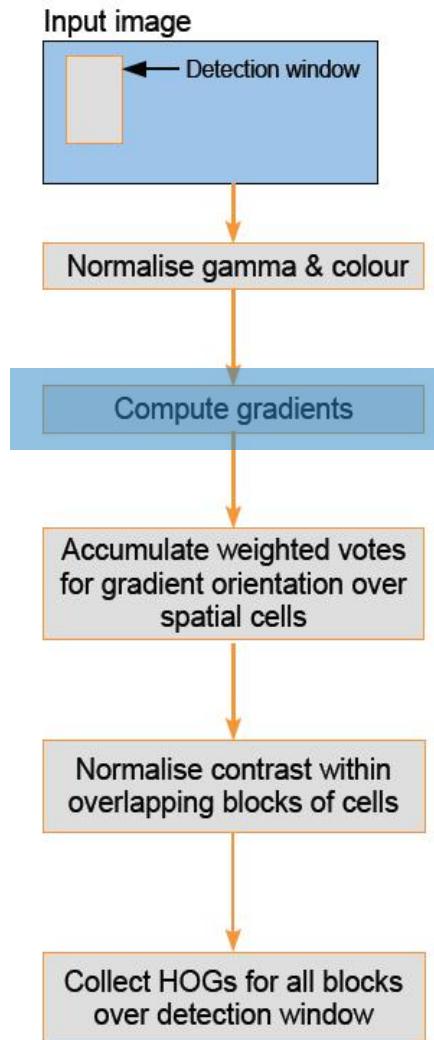
DESCRIPTEUR HOG

Histogram Of Gradient

HISTOGRAM OF GRADIENT : HOG

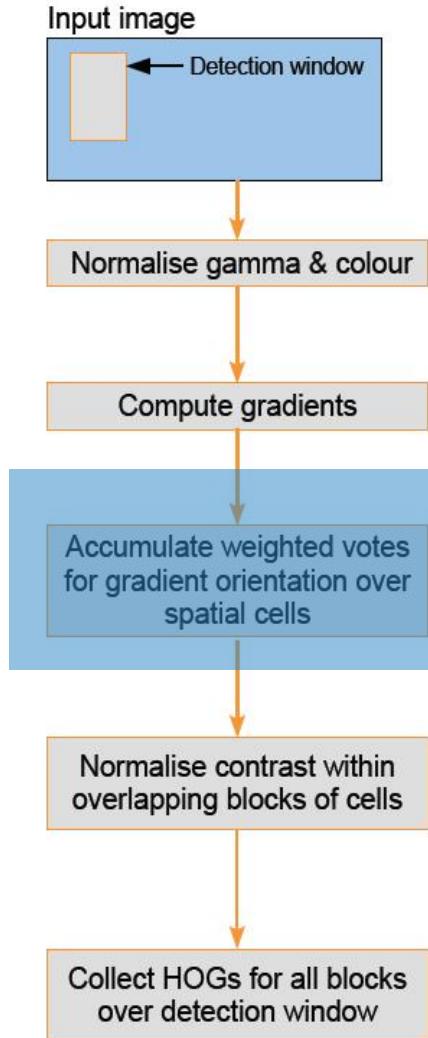


HOG: CALCUL DE GRADIENTS

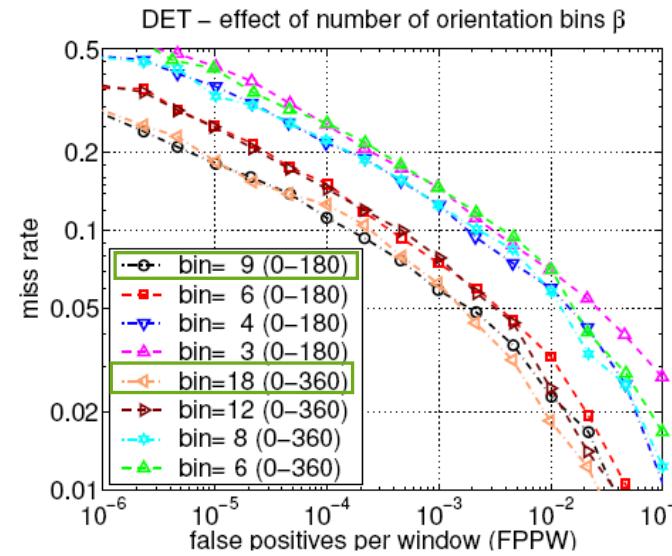


Type du masque	Centré 1D	Non centré 1D	Cubique-corrigé 1D	Diagonale 2x2	Sobel 3x3
Opérateur	$[-1, 0, 1]$	$[-1, 1]$	$[1, -8, 0, 8, -1]$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

HOG: ACCUMULER DES VOTES DE POIDS



- Combien de Bins doivent figurer dans l'histogramme?
- Devrions-nous utiliser des gradients orientés ou non orientés?
- Comment sélectionner les poids?
- Quelle taille de bloc devrions-nous utiliser?



HOG: VECTEUR DE CARACTÉRISTIQUES POUR UN BLOCK



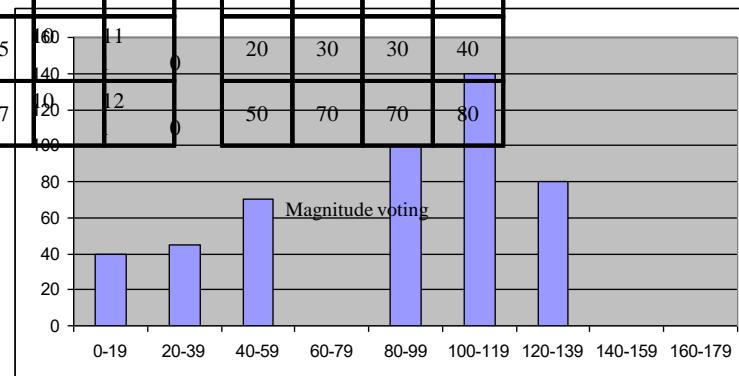
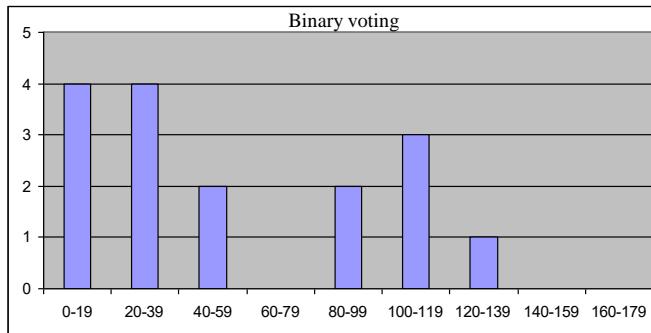
$$f = (h_1^1, \dots, h_9^1, h_1^2, \dots, h_9^2, h_1^3, \dots, h_9^3, h_1^4, \dots, h_9^4)$$

Angle

0	15	25	25
10	15	25	30
45	95	60	11
47	97	40	12

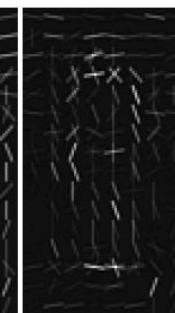
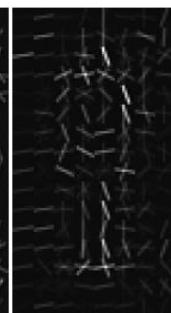
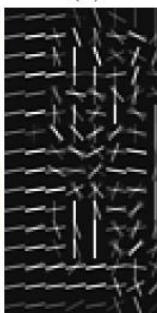
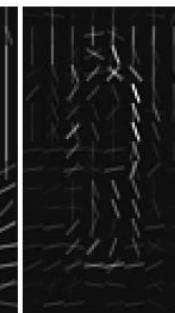
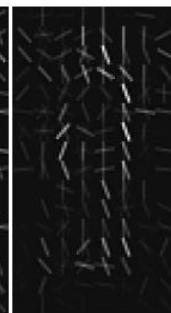
Magnitude

5	20	20	10
5	10	10	5
20	30	30	40
50	70	70	80

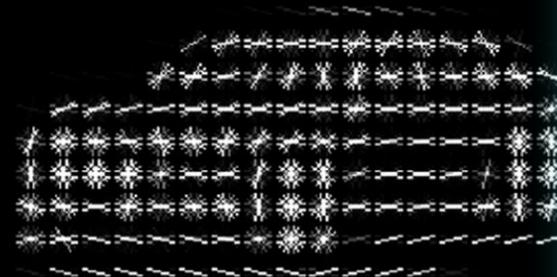


Le vecteur de caractéristiques s'étend pendant que la fenêtre se déplace

HOG: EXEMPLE



Dans chaque triplet: (1) l'image d'entrée, (2) le vecteur de caractéristiques R-HOG correspondant (seule l'orientation dominante de chaque cellule est affichée), (3) les orientations de dominantes sélectionnées par le SVM (obtenues en multipliant le vecteur de caractéristiques par les poids correspondants du SVM linéaire).



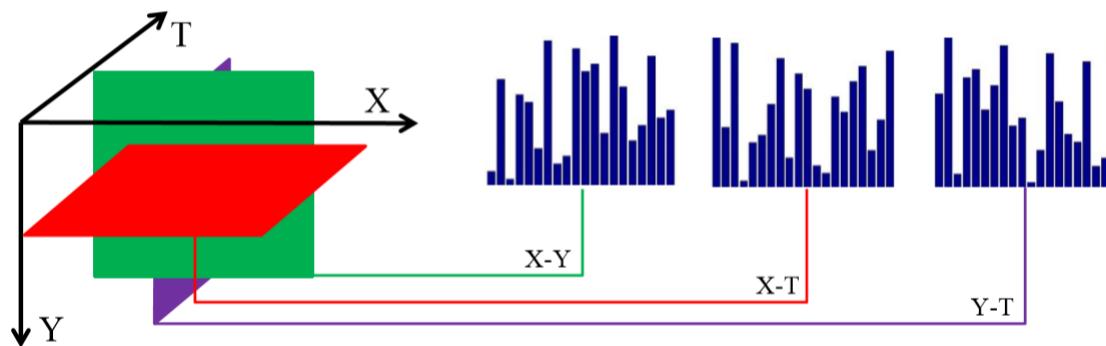
DESCRIPTEUR HOG-TOP: EXTENSION DE HOG

Histogram Of Gradient-
Three Orthogonal Planes

HOG THREE ORTHOGONAL PLANES (HOG-TOP)

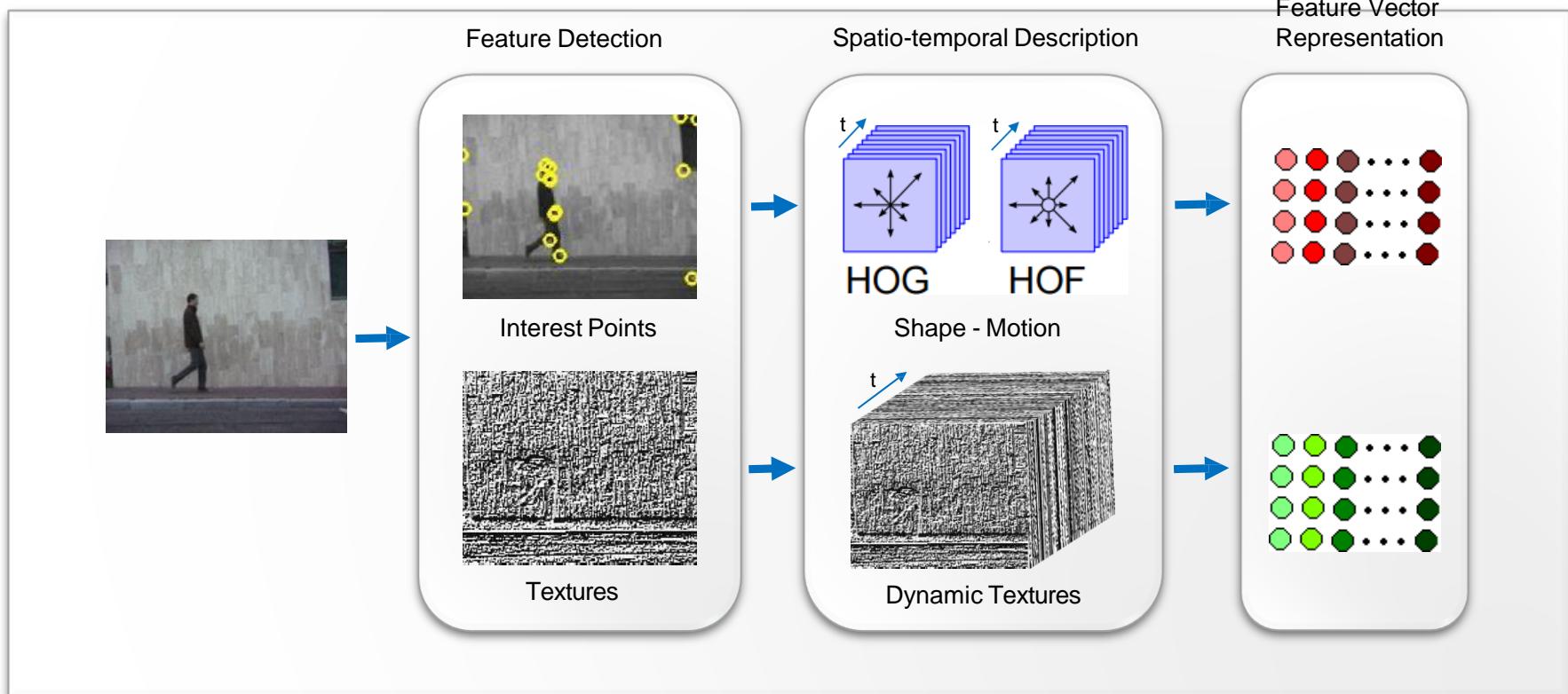


La texture dans les trois plans XY, XT et YT



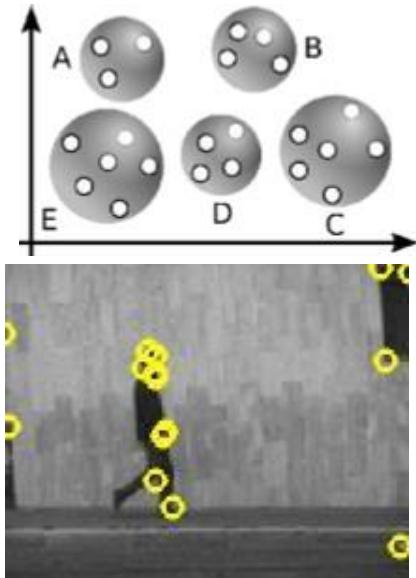
HOG dans les trois plans XY, XT et YT

DÉTECTION ET PRÉSENTATION DU VECTEUR DE CARACTÉRISTIQUES

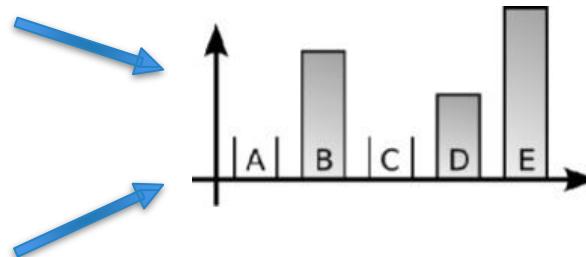


REPRÉSENTATION « BAG-OF-WORDS »: BAG OF SPACE-TIME FEATURES + SVM

Les vecteurs de caractéristiques d'apprentissage sont regroupés avec k-means



Chaque vecteur de caractéristiques est assigné à son centre le plus proche (visual word)



Une séquence vidéo complète est représentée sous forme d'un histogramme d'occurrence « visual words »

Classification avec un classifieur multi-classe et non linéaire (SVM)



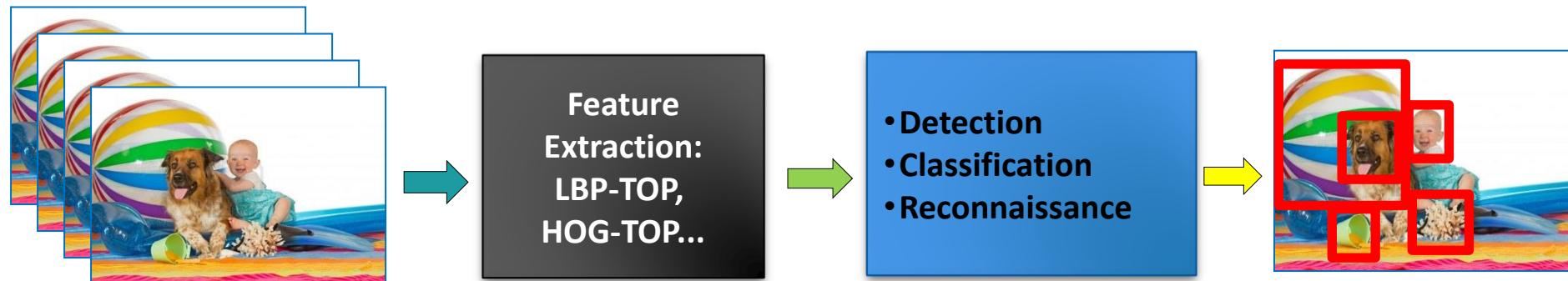
DEEP LEARNING ET EXTRACTION AUTOMATIQUE DE CARACTÉRISTIQUES

CNN + RNN

MACHINE LEARNING VS DEEP LEARNING

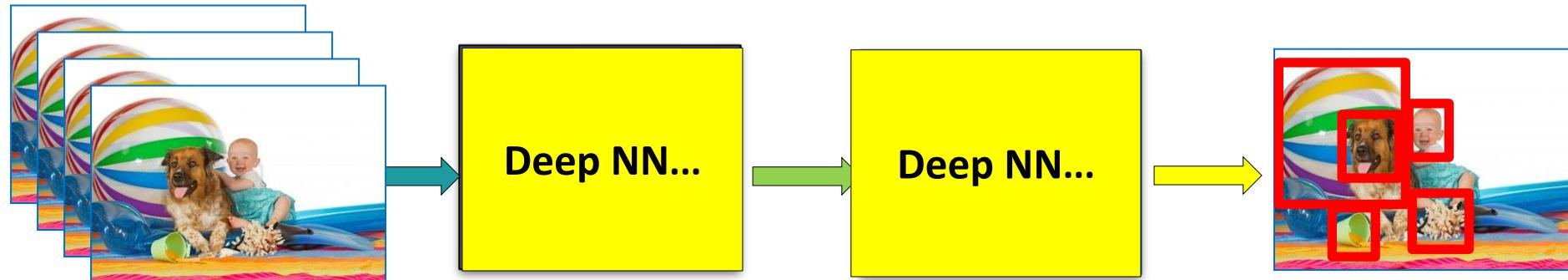
Extraire manuellement le vecteur caractéristiques: SURF, SIFT, HOG-TOP, LBP-TOP, Harris 3D...

Il faut entraîner un classifieur (SVM, RN,....)



DEEP LEARNING

- Construire automatiquement le vecteur de caractéristiques (Features) en fonction de la base d'apprentissage.
- Combiner l'extraction des caractéristiques et la classification.



RÉSEAUX DE NEURONES CONVOLUTIFS (CNN)

Convolutional Neural Networks est une extension du perceptron multicouche traditionnel MLP, basé sur 3 idées:

Connectivité locale

Poids partagés

Invariance à la translation

Voir l'article de LeCun (1998) sur la reconnaissance de texte:

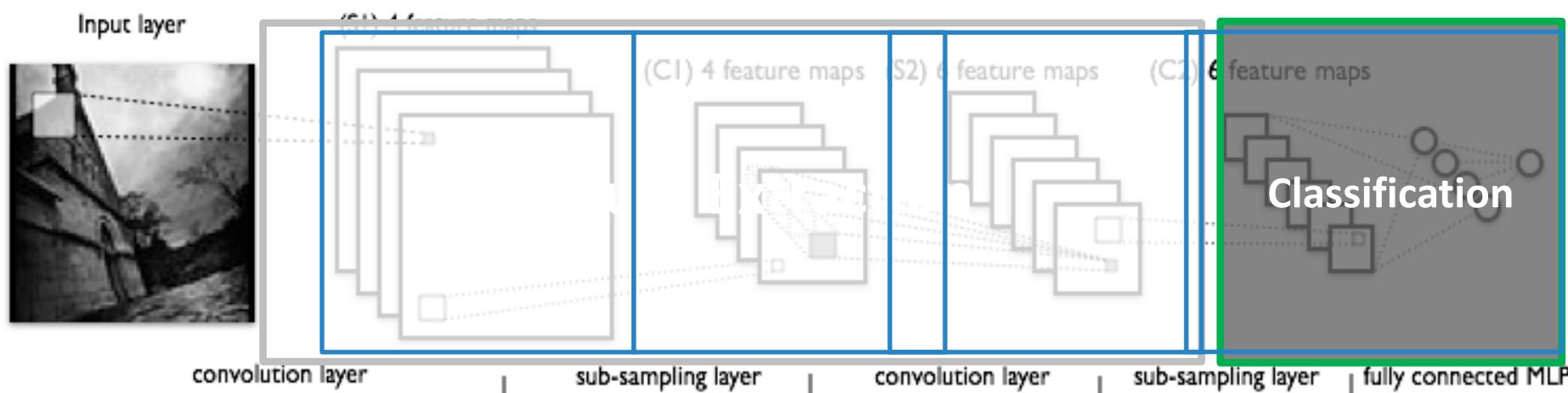
<http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf>

RÉSEAUX DE NEURONES CONVOLUTIFS (CNN)

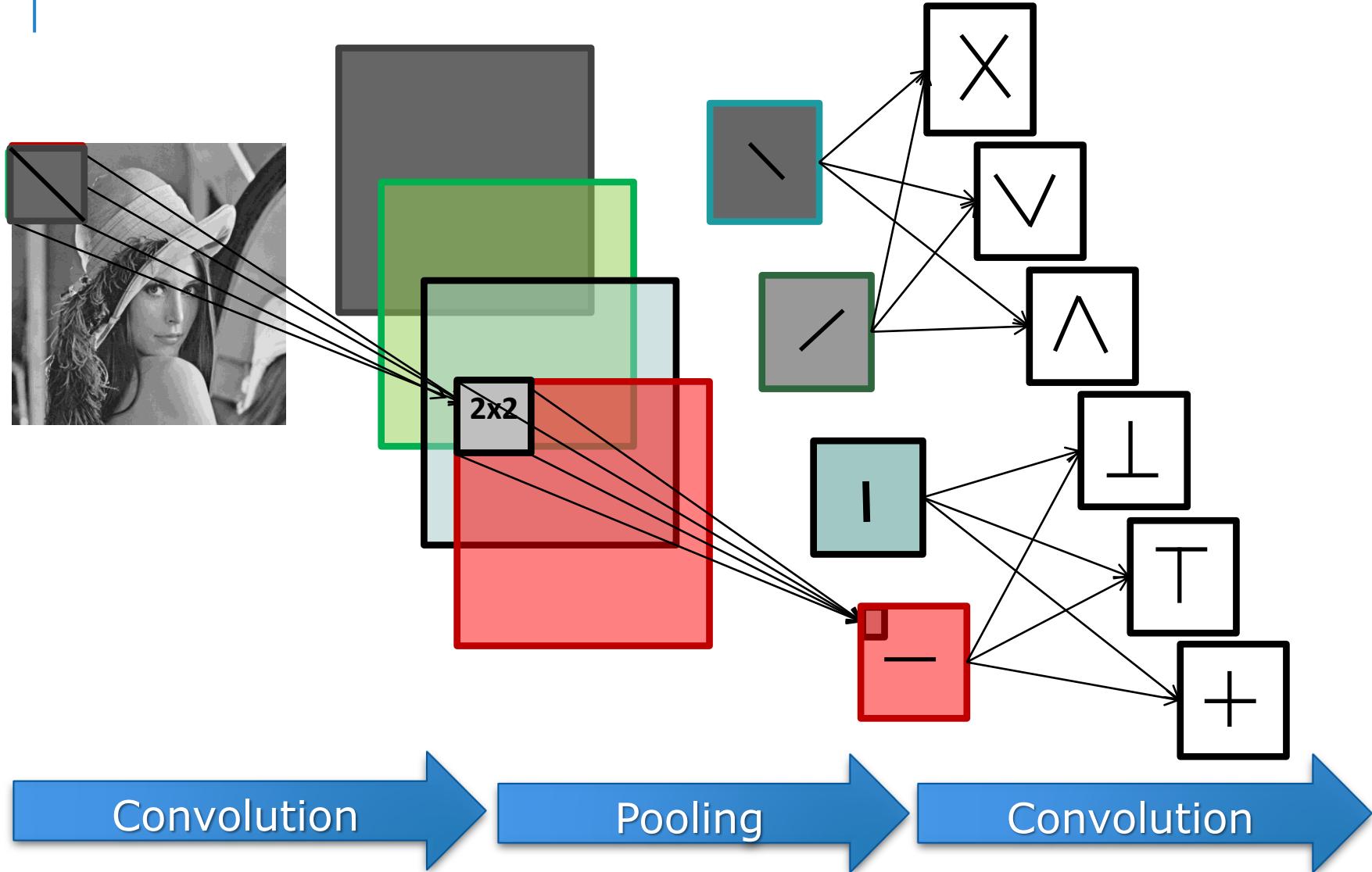
CNN - architecture NN multicouche

- Couche convective + non linéaire
- Couche de sous-échantillonnage
- Couche convolutionnelle + non-linéaire
- Couches entièrement connectées

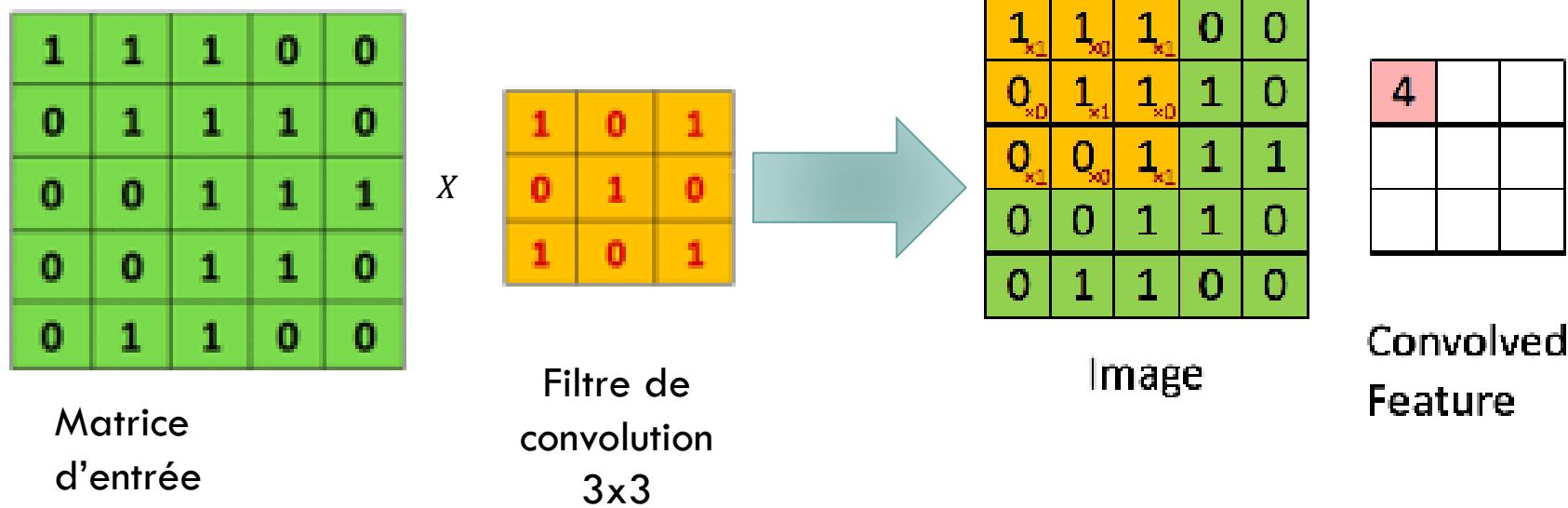
Supervisé



RÉSEAUX DE NEURONES CONVOLUTIFS

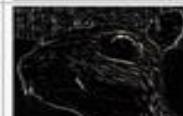


RÉSEAUX DE NEURONES CONVOLUTIFS (CNN)



RÉSEAUX DE NEURONES CONVOLUTIFS (CNN)

Effet des convolutions,
exemple sur des images :

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

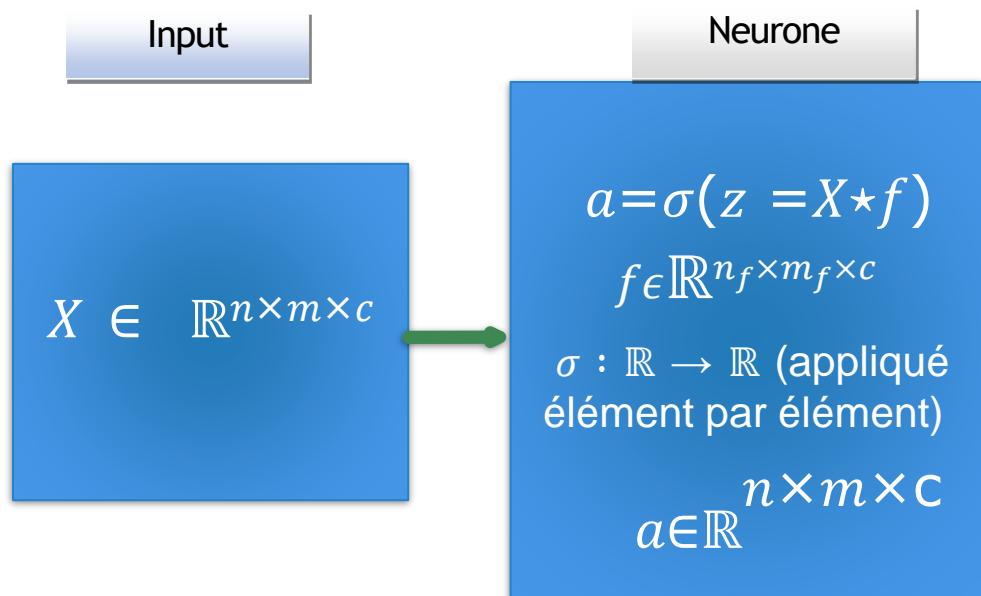
RÉSEAUX DE NEURONES CONVOLUTIFS (CNN)

Un neurone de convolution :

On ne choisit pas les filtres, on les apprend !!! Ce sont des paramètres entraînables du réseau

c s'appelle **nombre de canaux** :

- $c = 1$ pour une image en niveaux de gris.
- $c = 3$ pour une image RGB (une channel par couleur).



RÉSEAUX DE NEURONES CONVOLUTIFS (CNN)

Le Pooling : opération utilisée pour réduire la dimension

Recherche de détails plus « grossiers », de plus grandes « structures » dans l'image

Max-pooling de taille l : on prend l'élément maximal de chaque sous-tableau de taille l

Sum-pooling de taille l : on fait la somme de tous les éléments de chaque sous tableau de taille l

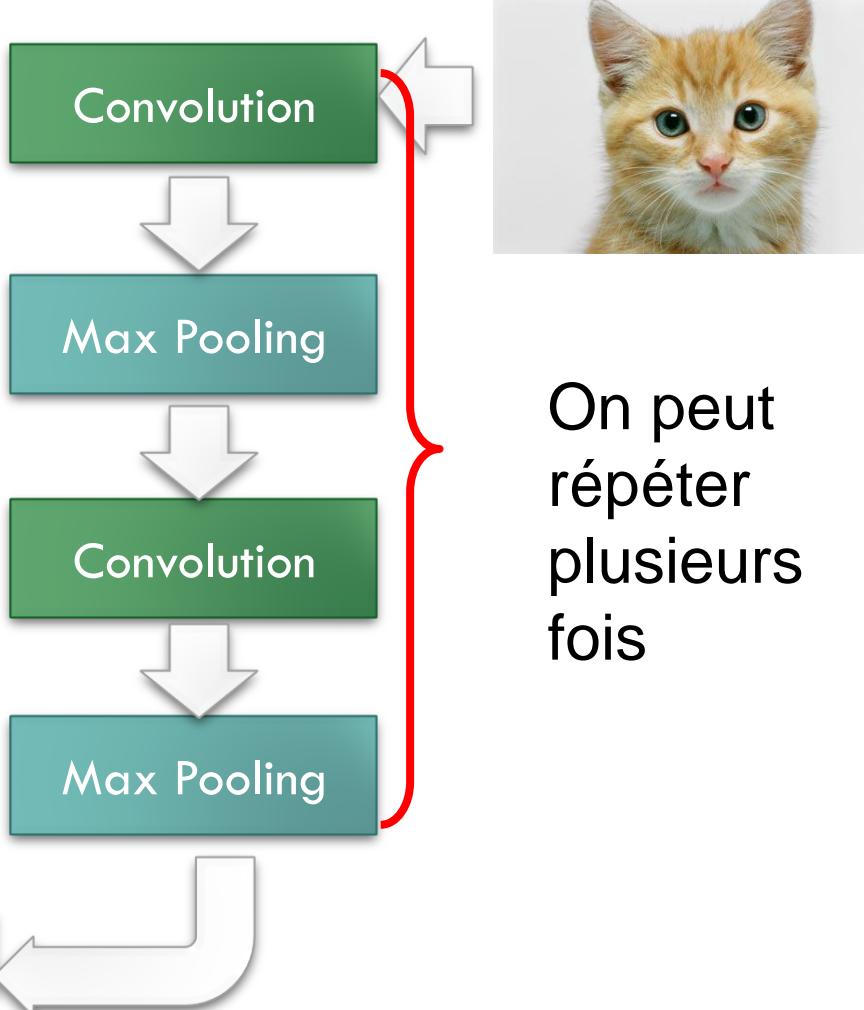
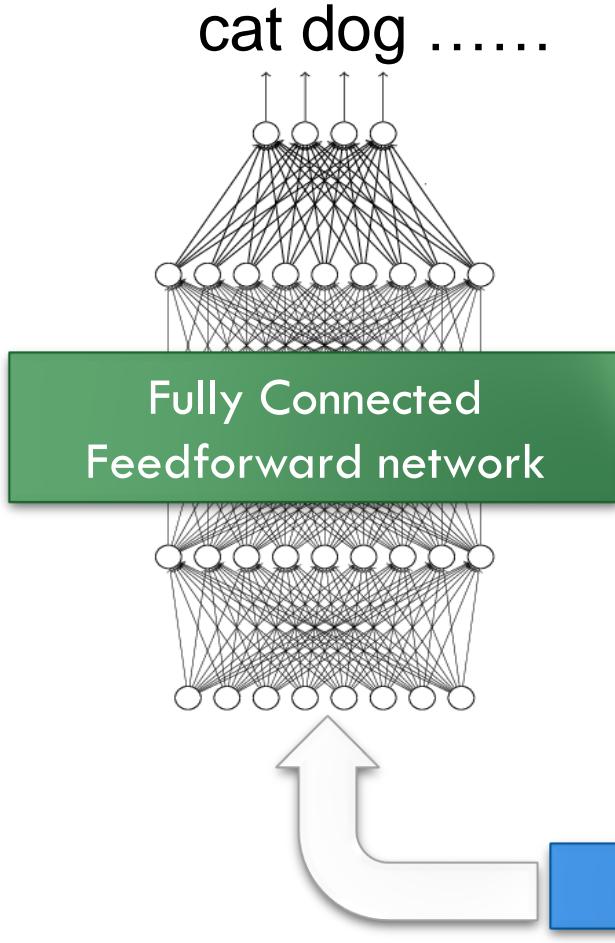
Feature Map

6	4	8	5
5	4	5	8
3	6	7	7
7	9	7	2

max pool
2x2 filters
and stride 2

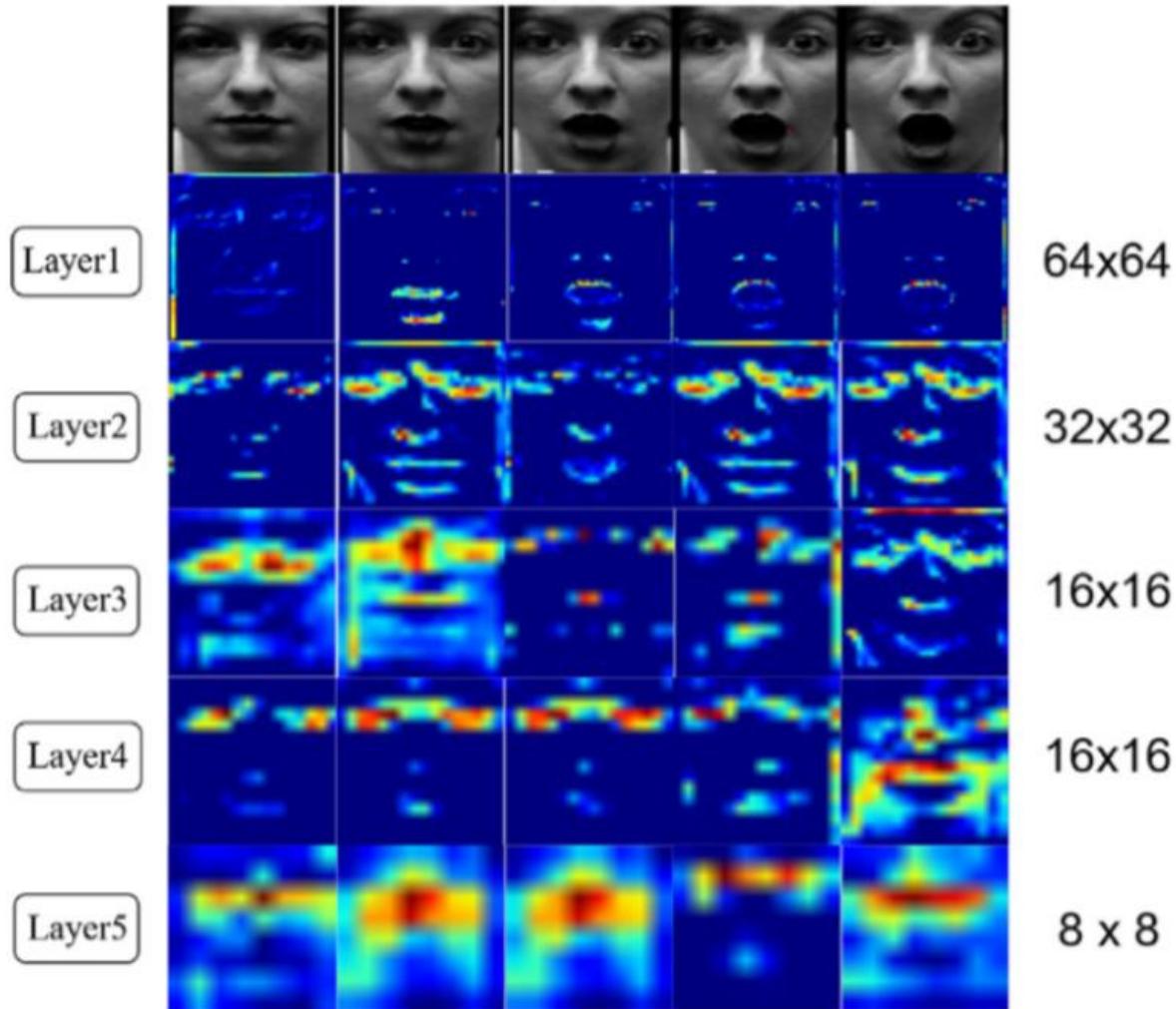
Max-Pooling

RÉSEAUX DE NEURONES CONVOLUTIFS (CNN)



RÉSEAUX DE NEURONES CONVOLUTIFS (CNN)

Les filtres apprennent des caractéristiques globales



RÉSEAUX DE NEURONES CONVOLUTIFS (CNN)

LeNet

- 1998- Nombre de paramètres: 60 mille

AlexNet

- 2012- Nombre de paramètres: 60 millions

VGG Net

- 2014 - Nombre de paramètres: 138 millions

GoogleNet

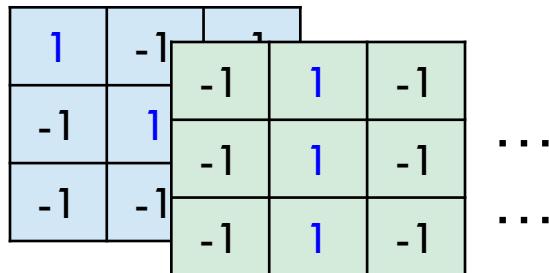
- 2014 - Nombre de paramètres: 4 millions

ResNet

- 2015

ARCHITECTURE CNN EN KERAS

```
model2.add( Convolution2D( 25, 3, 3,  
    input_shape=(28, 28, 1)) )
```



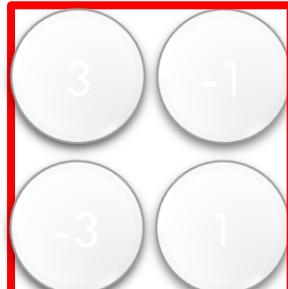
Il y a 25 filtres de taille 3x3.

Input_shape = (28 , 28 , 1)

28 x 28 pixels

1: Niveau de gris, 3: RGB

```
model2.add(MaxPooling2D( (2, 2) ))
```



input
↓

Convolution



Max Pooling

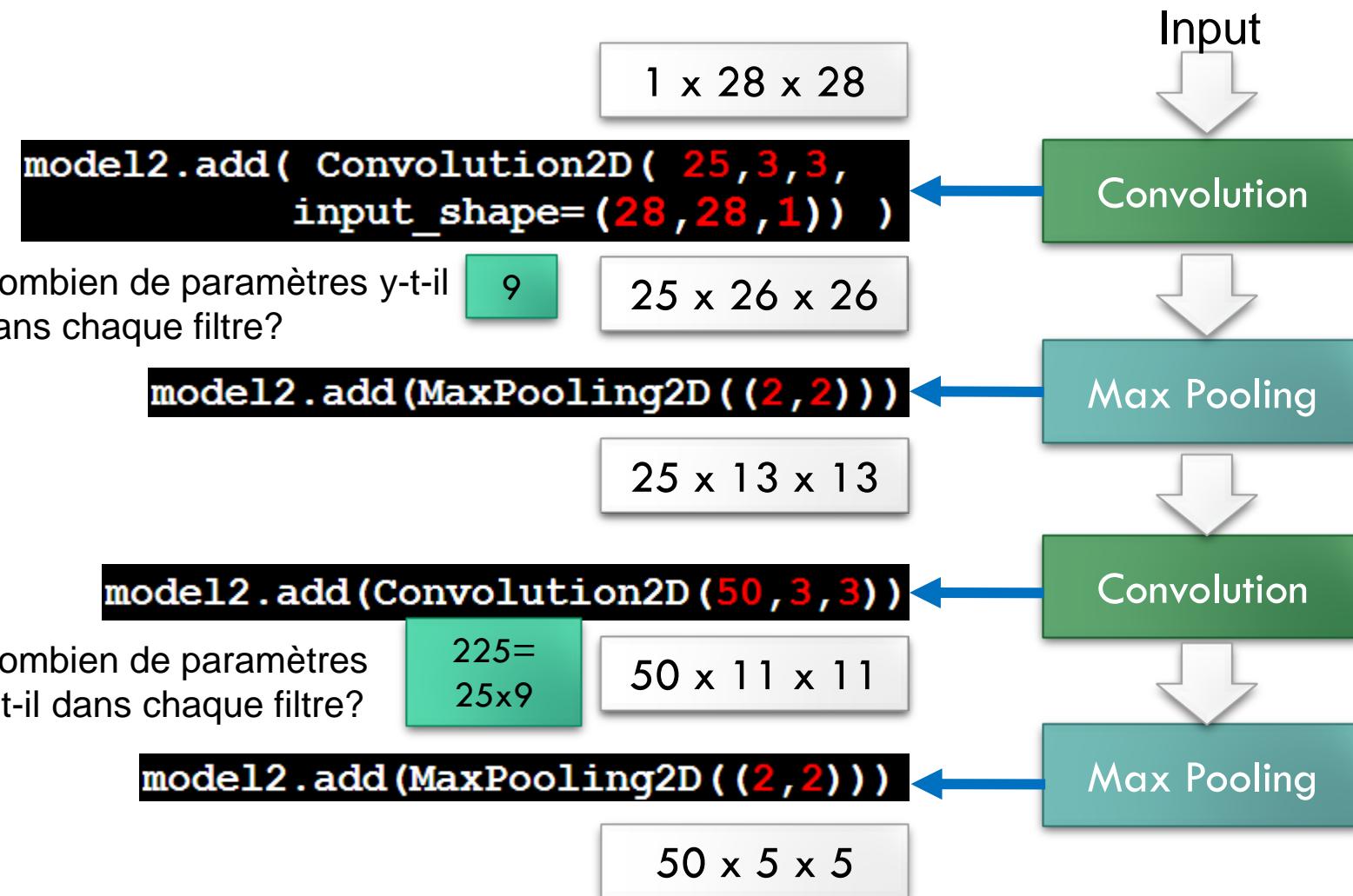


Convolution



Max Pooling

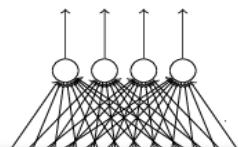
ARCHITECTURE CNN EN KERAS



Les trois couches existent aussi en version 1D et 3D

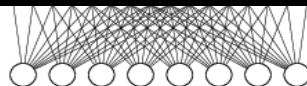
ARCHITECTURE CNN EN KERAS

Output



Fully connected
feedforward network

```
model2.add(Dense(output_dim=100))  
model2.add(Activation('relu'))  
model2.add(Dense(output_dim=10))  
model2.add(Activation('softmax'))
```



1250

Flattened

```
model2.add(Flatten())
```

Input

$1 \times 28 \times 28$

Convolution

$25 \times 26 \times 26$

Max Pooling

$25 \times 13 \times 13$

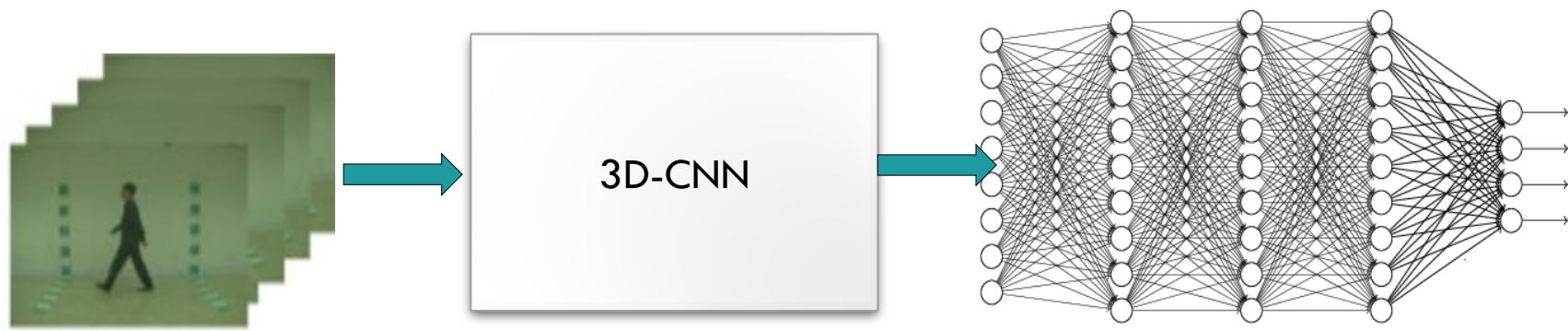
Convolution

$50 \times 11 \times 11$

Max Pooling

$50 \times 5 \times 5$

CNN 3D

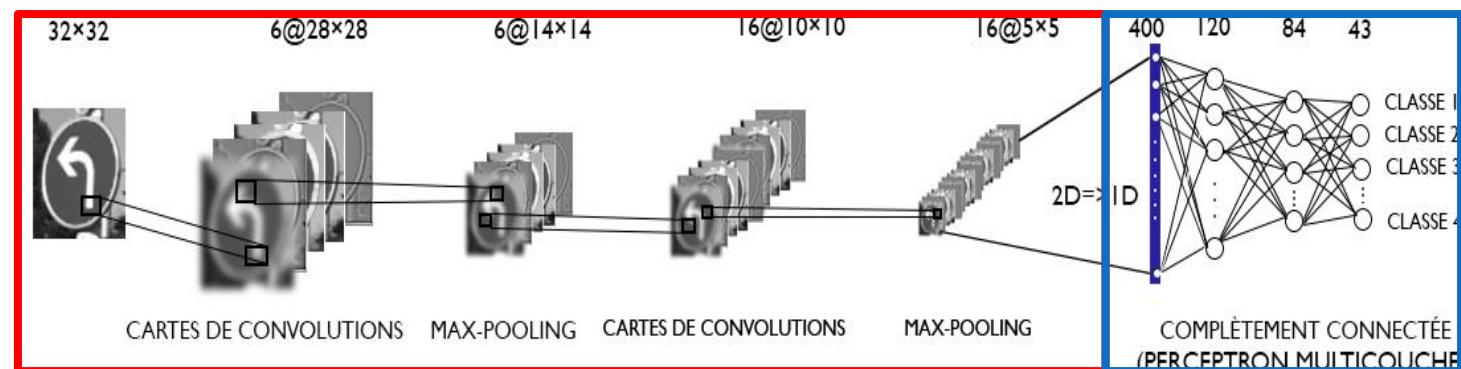


TRANSFER LEARNING

Idée : conserver l'extraction des caractéristiques apprise sur d'autres problématiques

Revient à conserver des couches de convolution apprises sur un problème similaire

On ne change que la (ou éventuellement les) dernière(s) couche(s) d'identification



LES RÉSEAUX RÉCURRENTS (RNN)

Problème à traiter : analyse de séquences (textes, enregistrement audio/vidéo)

- Input de tailles différentes
- Ordre dans les données

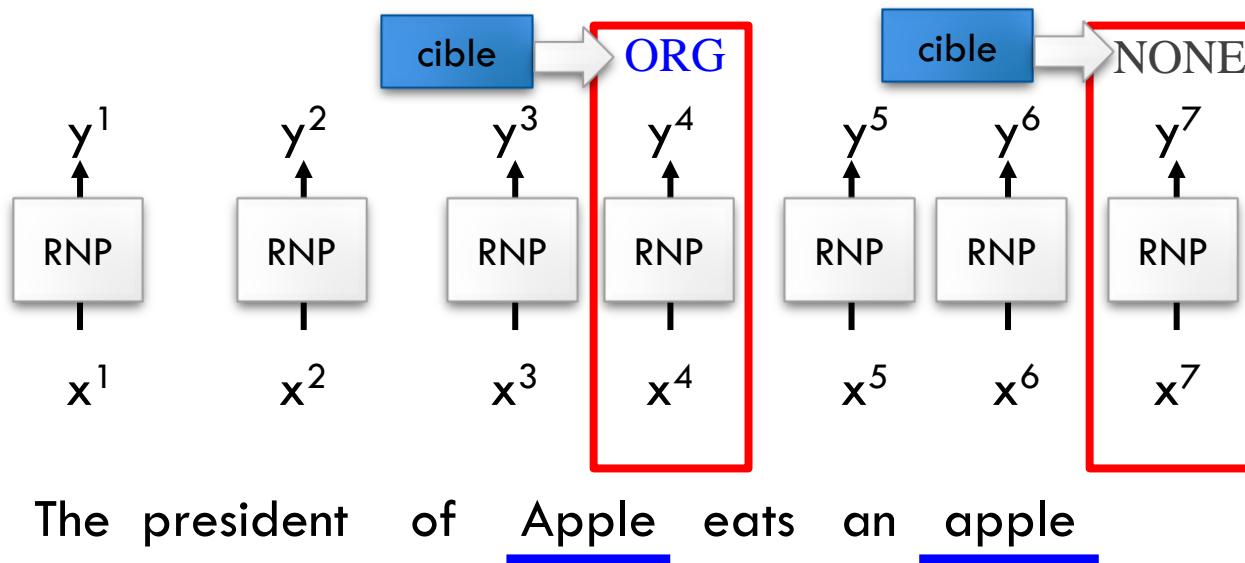
Recurrent Neural Network (+ LSTM) :

- Les neurones récurrents prennent en input :
 - La nouvelle information (le nouveau mot, la nouvelle image...)
 - Sa propre sortie précédente (ce que le neurone avait calculé avant)

POURQUOI MÉMORISER

Par exemple, interprétation de noms d'entités

- Personnes, lieux, organisations, etc. dans une phrase

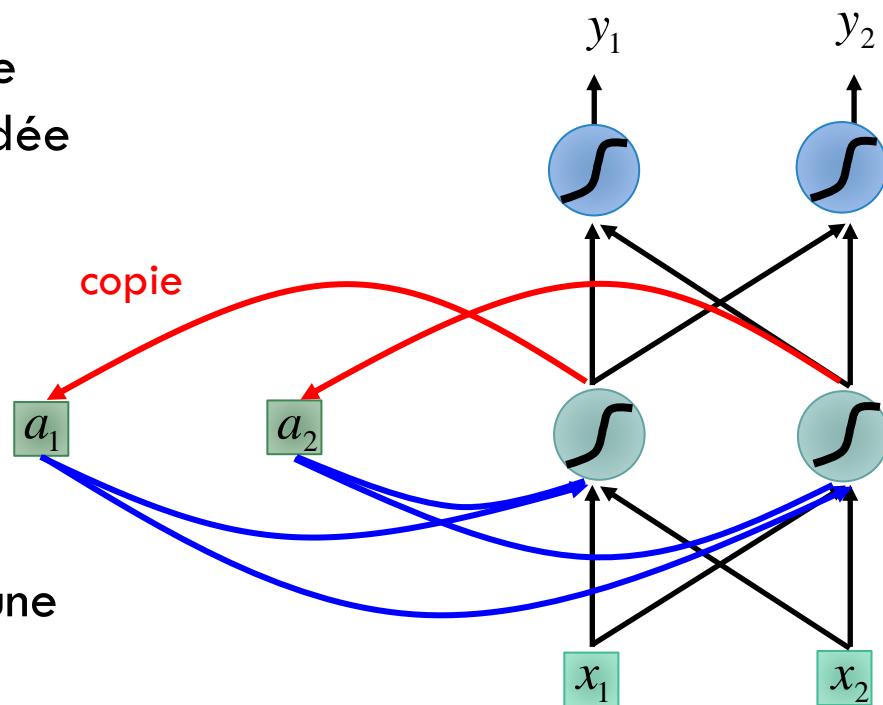


Besoin d'une mémoire de contexte!

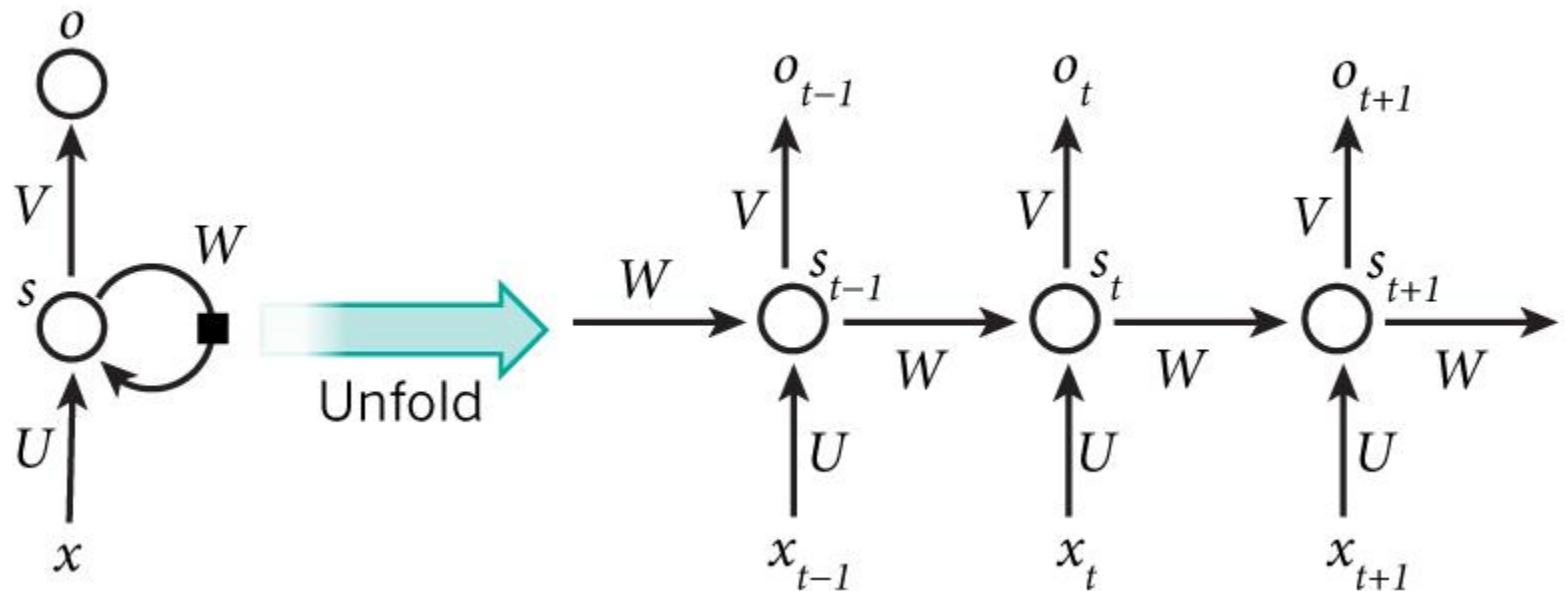
RÉSEAUX DE NEURONES RÉCURRENT (RNN)

La sortie de la couche
cachée est sauvegardée
entre itérations

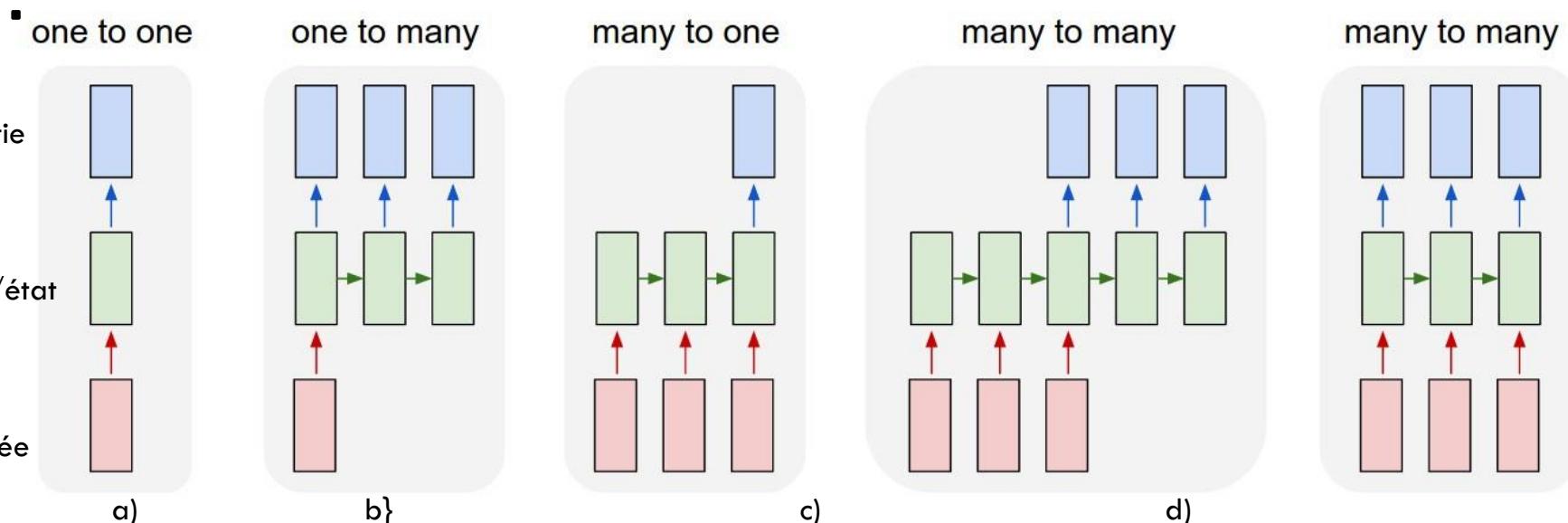
La mémoire devient une
entrée additionnelle



RÉSEAUX DE NEURONES RÉCURRENTS

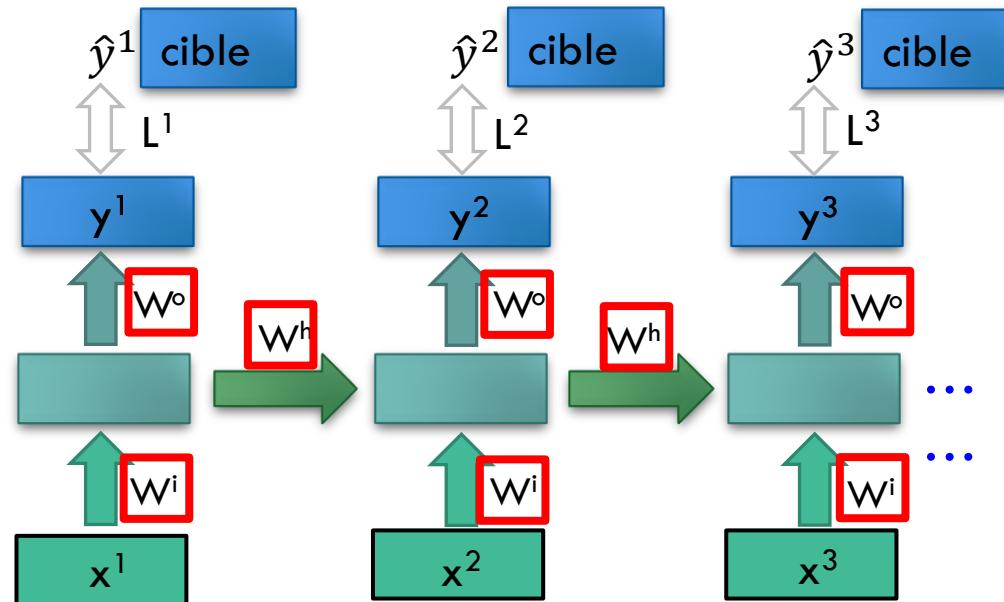


PERMETTENT D'OPÉRER SUR DES SÉQUENCES



Rectangle = vecteur; flèche = fonction. **(a)** Entrée fixe à sortie fixe (ex. classification d'image); **(b)** séquence en sortie (ex. annotation d'image, annotation en sortie); **(c)** séquence en entrée (ex. question en entrée, oui/non en sortie); **(d)** séquence à séquence (ex. traduction). **(e)** séquence à séquence synchrone (e.g. annotation de vidéo). Noter l'absence de contrainte de longueur due à couche verte récurrente.

COMMENT FAIRE L'APPRENTISSAGE ?

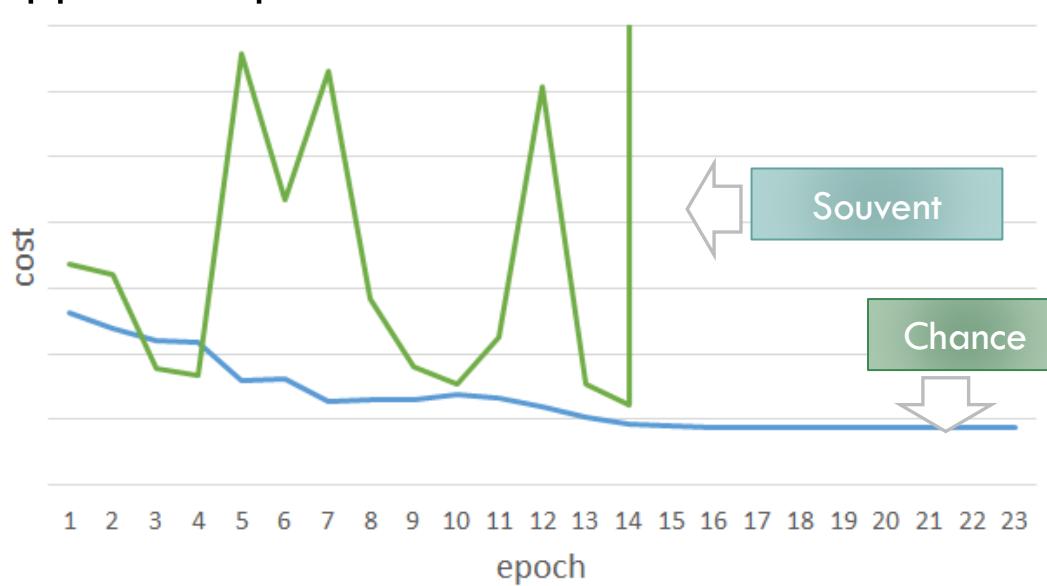


Trouver les valeurs des paramètres qui minimisent la fonction coût

Backpropagation through time (BPTT)

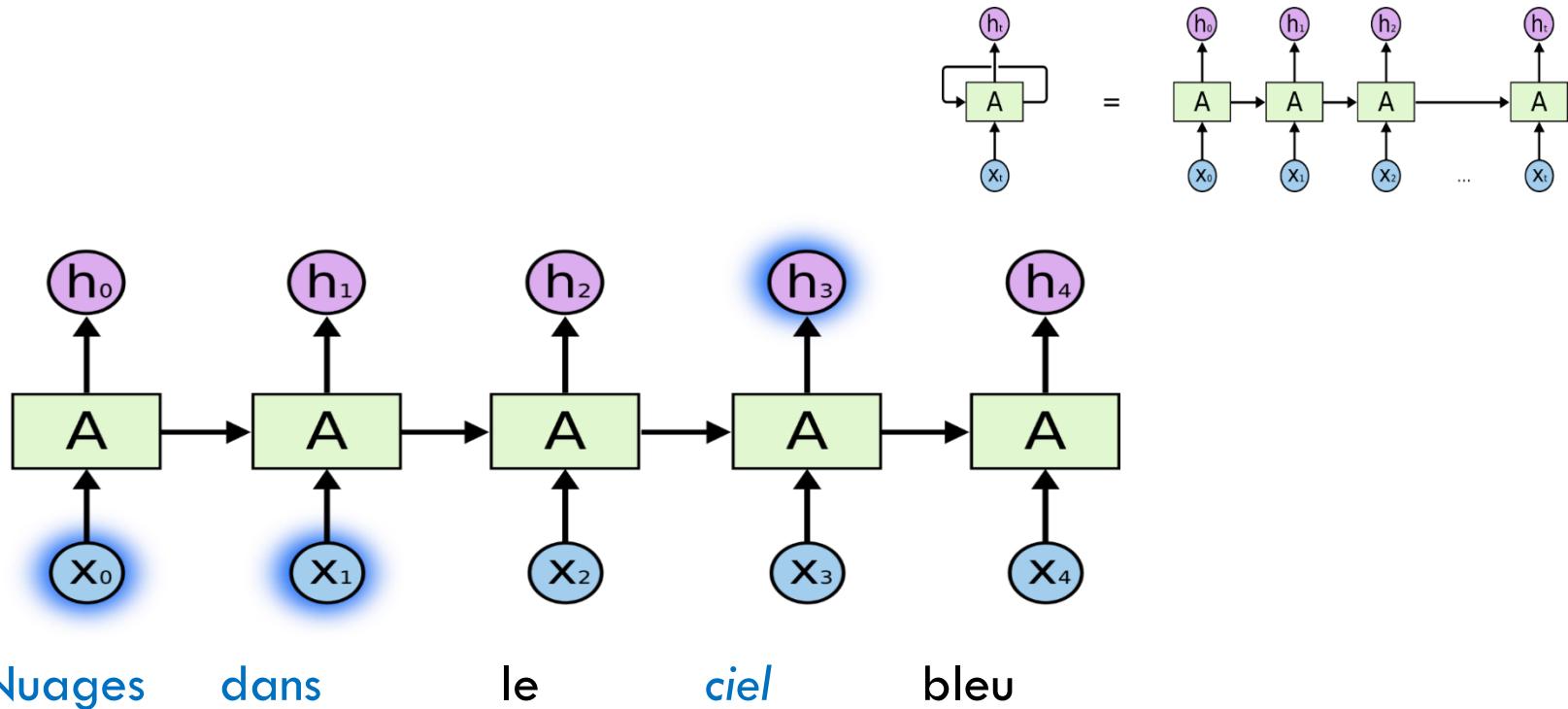
MALHEUREUSEMENT

Les RNR n'apprennent pas facilement



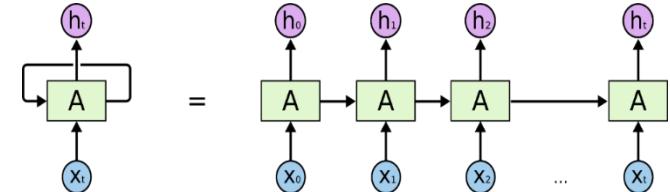
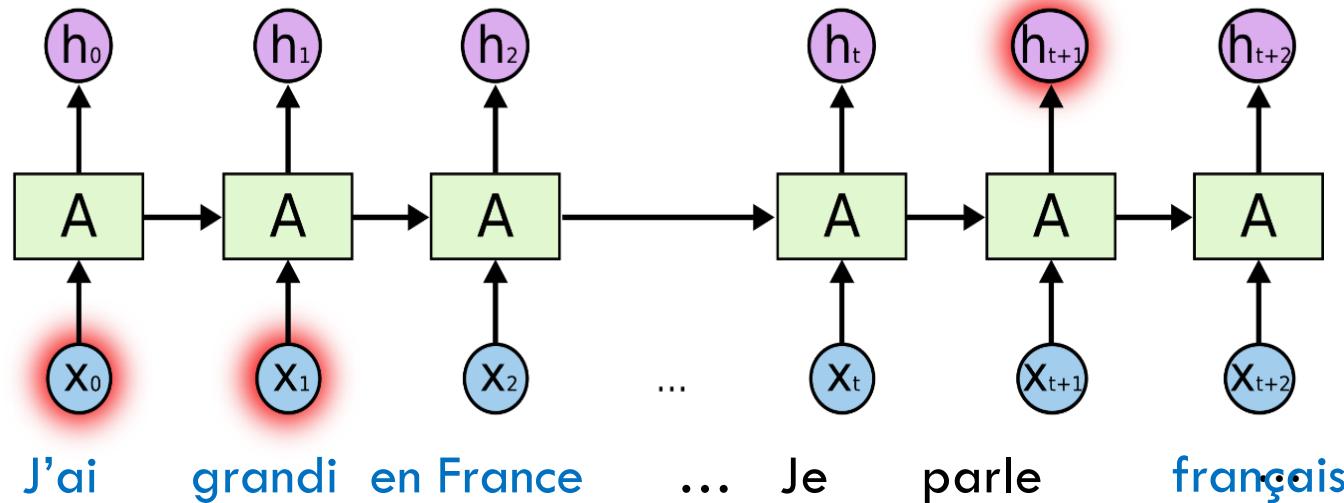
- Déplier le réseau pour l'apprentissage mène aux problèmes de gradient!

LONG SHORT TERM MEMORY



Le contexte est proche du mot à prédire; peu d'itérations les séparent

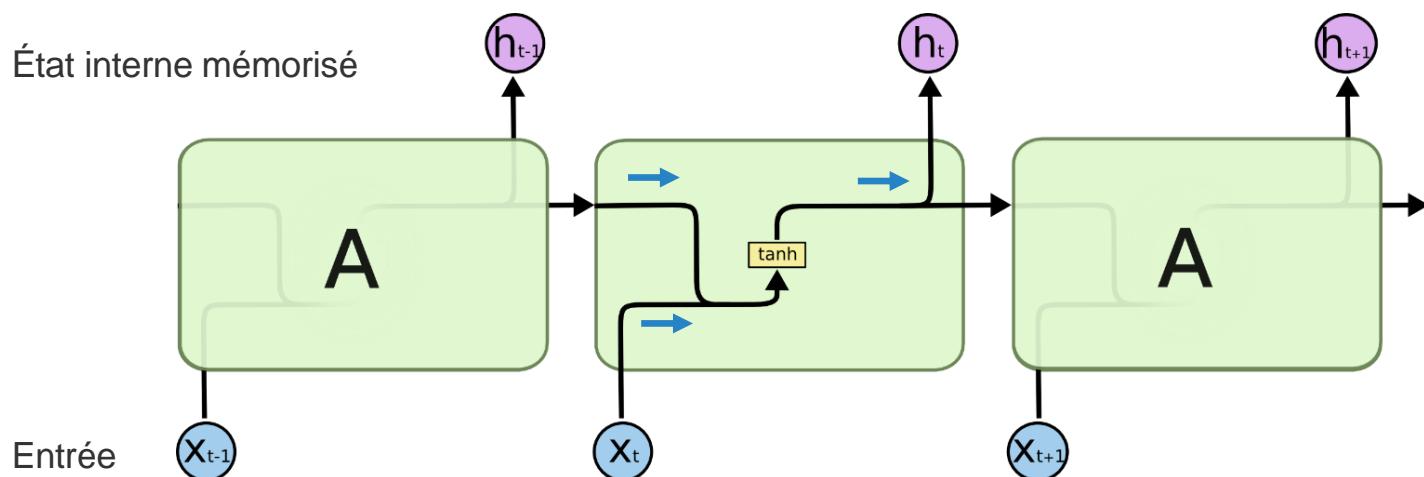
LSTM



Le contexte est loin du mot à prédire; beaucoup d'itérations les séparent!
=> problème de gradient possible

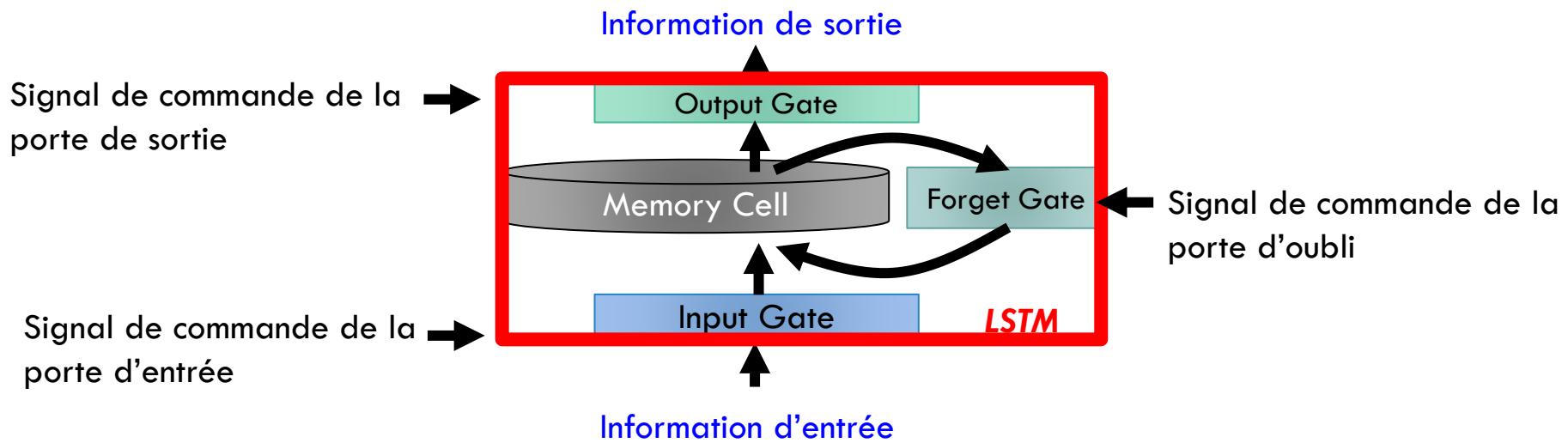
IL FAUT EMPÊCHER LE GRADIENT DE DISPARAÎTRE...

- Normalement, la mémoire du réseau est $h_t = \tanh(W \cdot [h_{t-1}, x_t] + b)$ et implique un seul niveau de traitement, créant le risque du **vanishing gradient**.

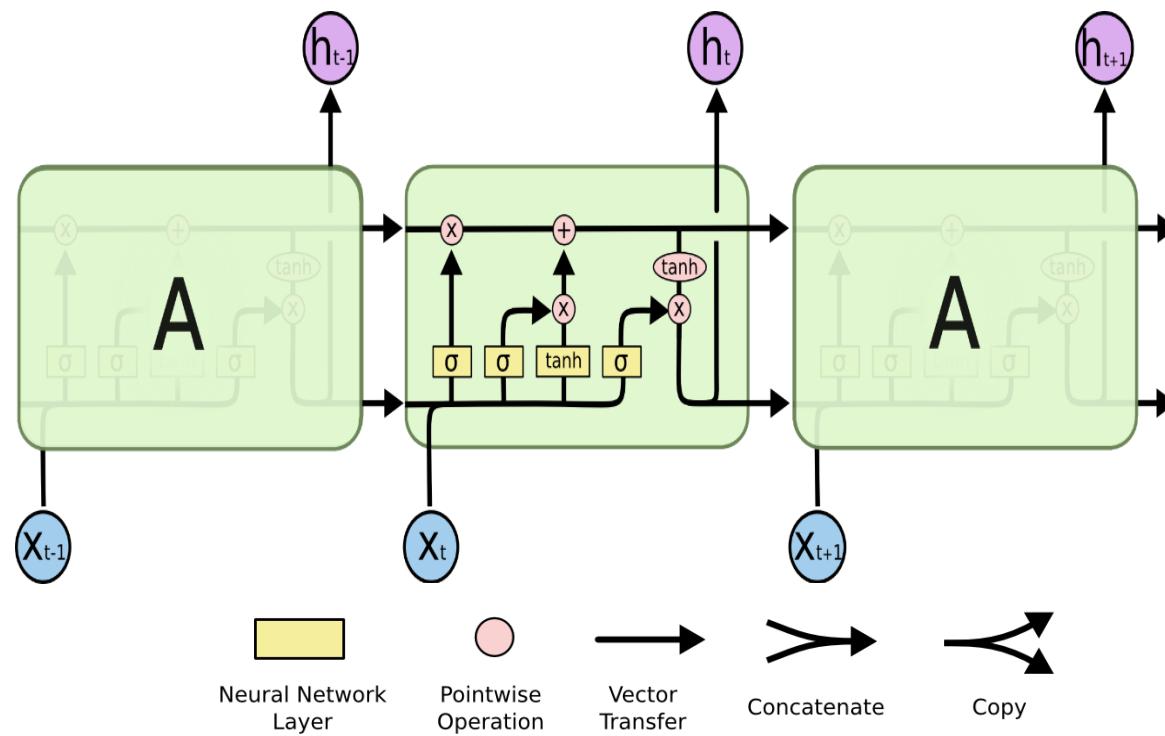


LSTM

- Ajoute une mémoire de contexte qui affecte le flux d'information et son traitement (cell state)
- Trois portes décident ce qu'une cellule doit oublier de son passé, retenir pour son état courant, et rendre disponible pour le futur



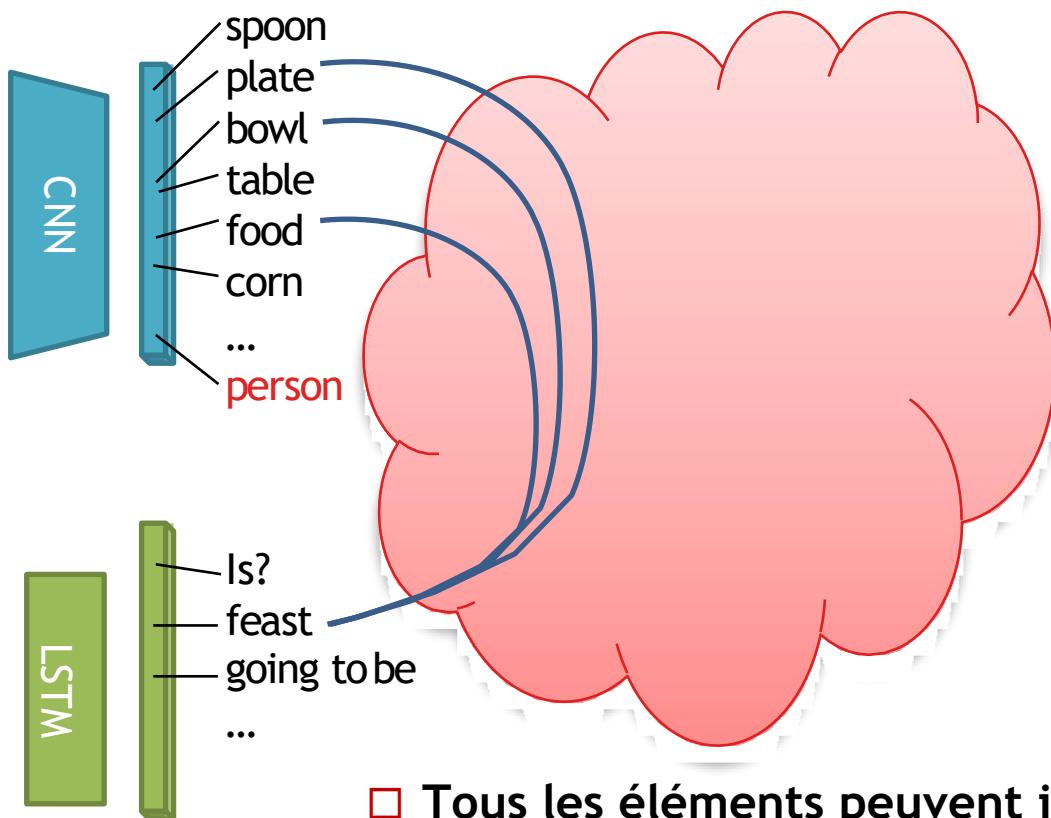
LSTM



FUSION DES CARACTÉRISTIQUES

MCB: Multimodal Compact
Bilinear Pooling

FUSION DE CARACTÉRISTIQUES

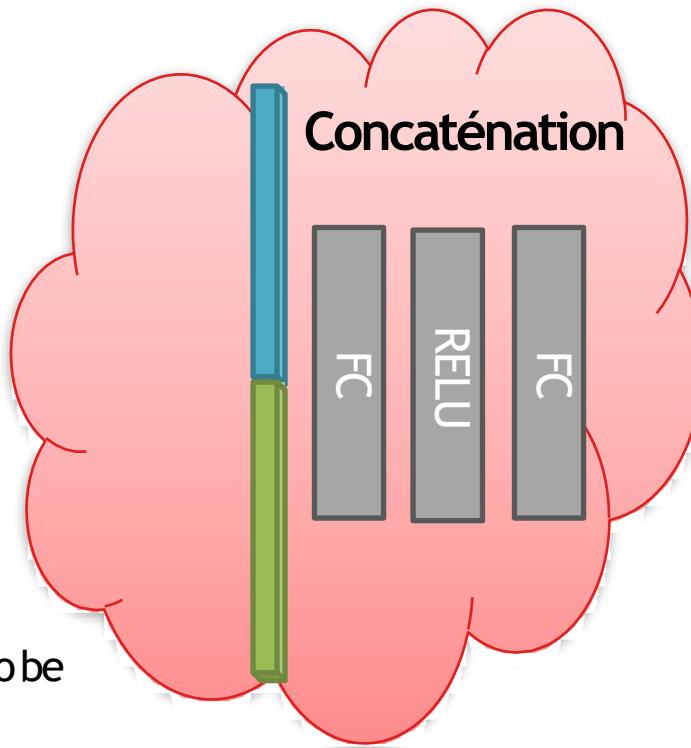
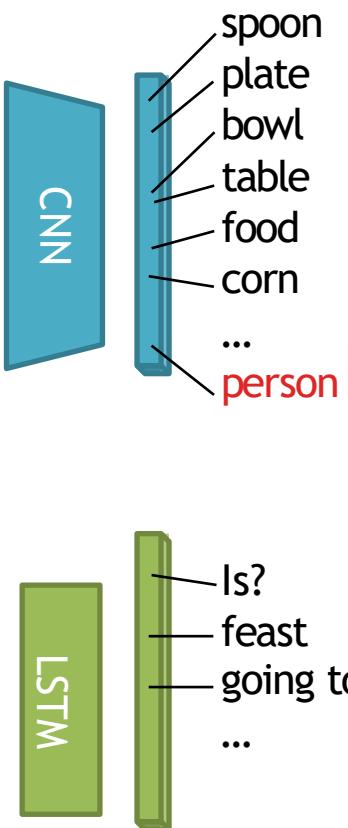


- Tous les éléments peuvent interagir
- Interaction multiplicative

FUSION DE CARACTÉRISTIQUES



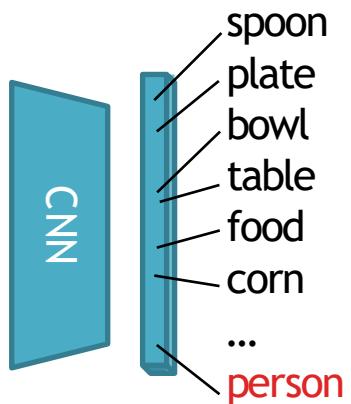
*Is this going to be
a feast?*



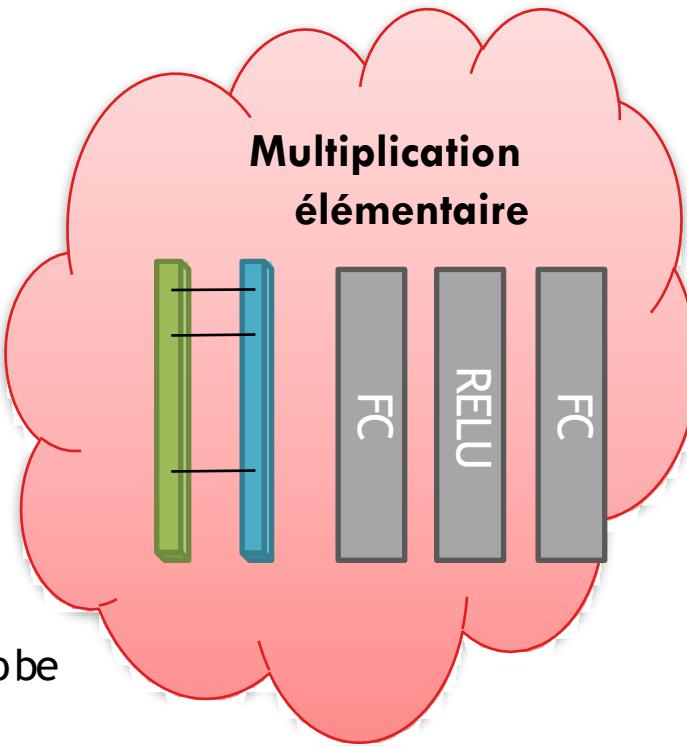
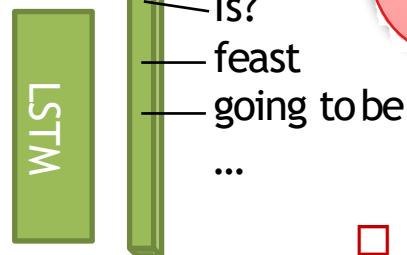
Yes

- Tous les éléments peuvent interagir
- Interaction multiplicative
 - Difficile d'apprendre l'output de la classification

FUSION DE CARACTÉRISTIQUES



*Is this going to be
a feast?*

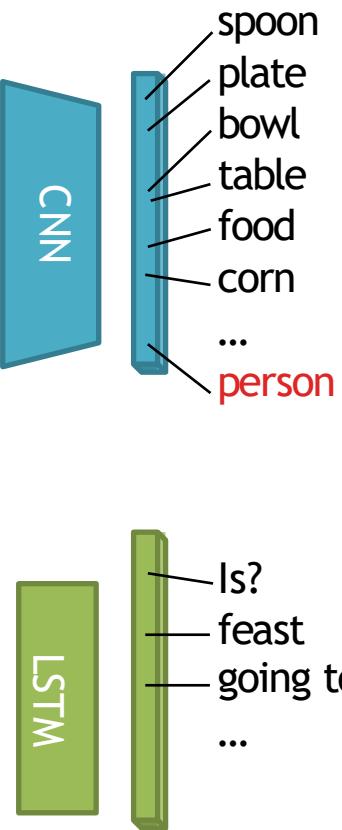


- Tous les éléments peuvent interagir
- Interaction multiplicative
 - Difficile d'apprendre l'incorporation des entrées

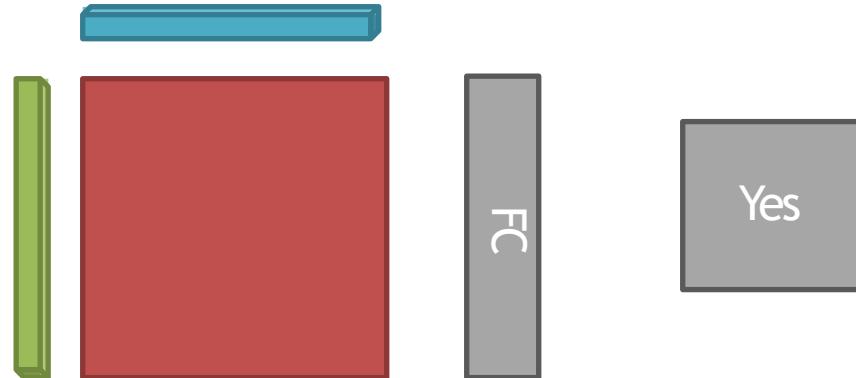
FUSION DE CARACTÉRISTIQUES



*Is this going to be
a feast?*



Outer Product /
Bilinear Pooling [Lin ICCV 2015]



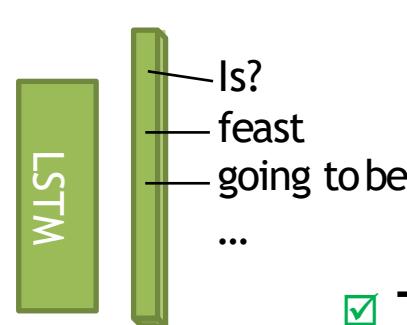
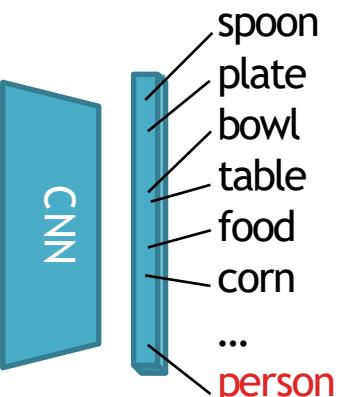
- Tous les éléments peuvent interagir
- Interaction multiplicative

FUSION DE CARACTÉRISTIQUES

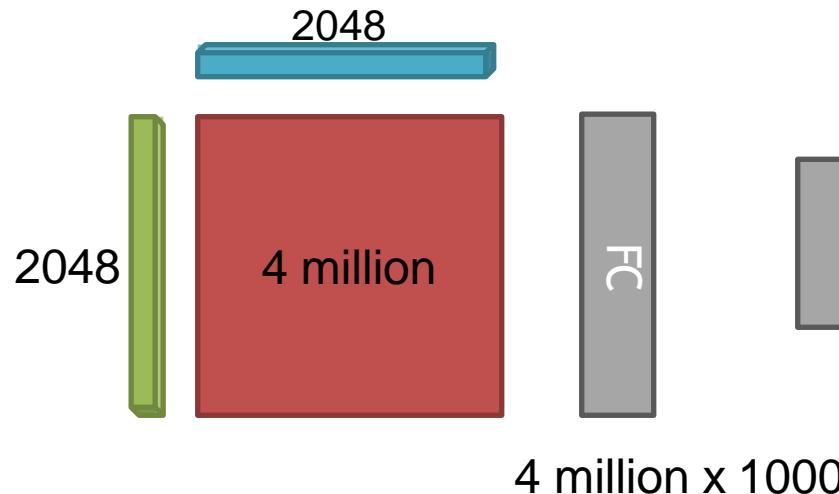
[Lin ICCV 2015]



*Is this going to be
a feast?*

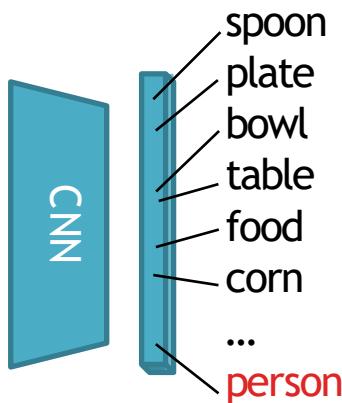


Outer Product /
Bilinear Pooling

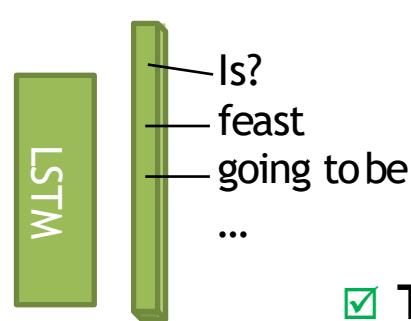


- Tous les éléments peuvent interagir
- Interaction multiplicative
- Nombre élevé d'activations et de calculs
- Nombre élevé de paramètres

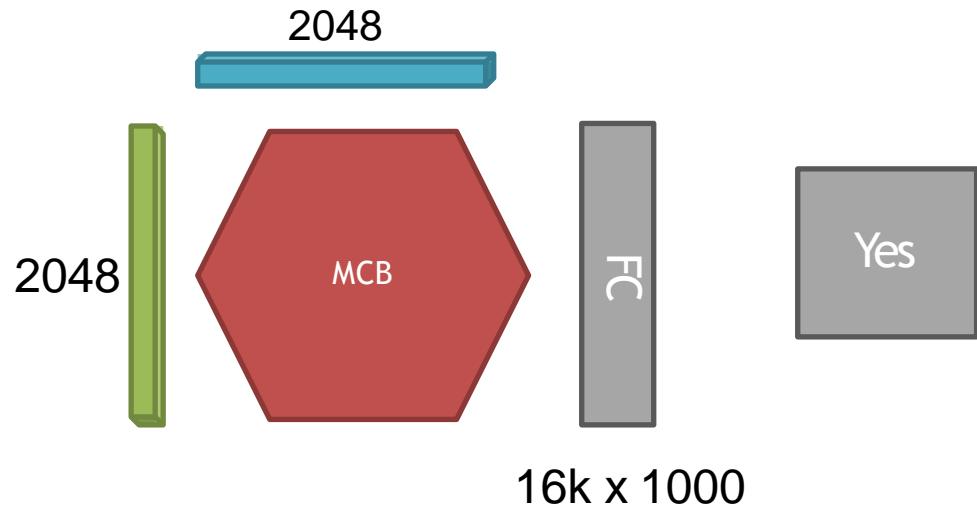
FUSION DE CARACTÉRISTIQUES



*Is this going to be
a feast?*



Compact
Bilinear Pooling [Gao CVPR 16]

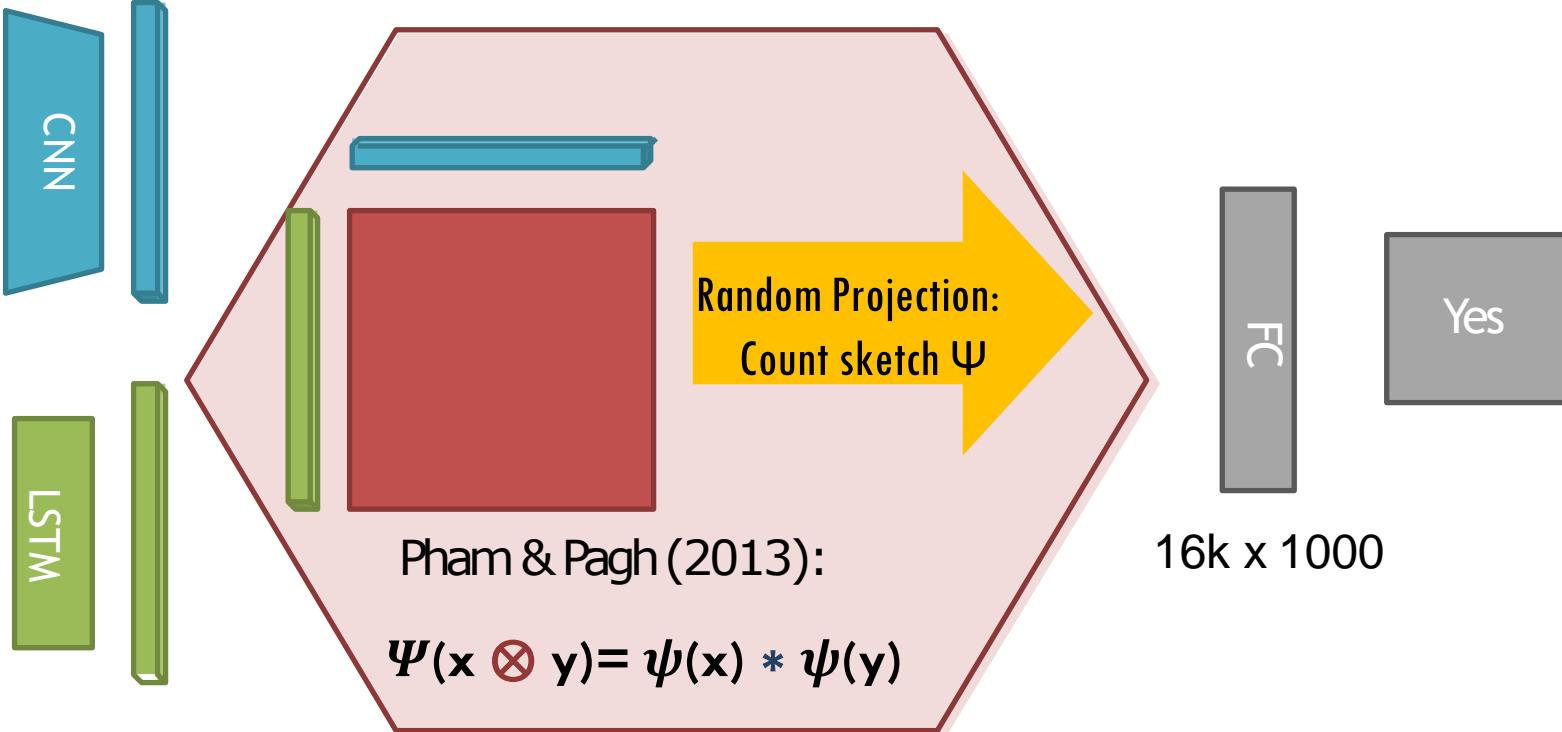


[ICLR Workshops 2016] Fine-grained pose prediction, normalization, and recognition N Zhang, E Shelhamer, Y Gao, T Darrell

[Gao CVPR 16] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. CVPR 2016

- Tous les éléments peuvent interagir
- Interaction multiplicative
- Nombre élevé d'activations et de calculs
- Nombre élevé de paramètres

FUSION DE CARACTÉRISTIQUES

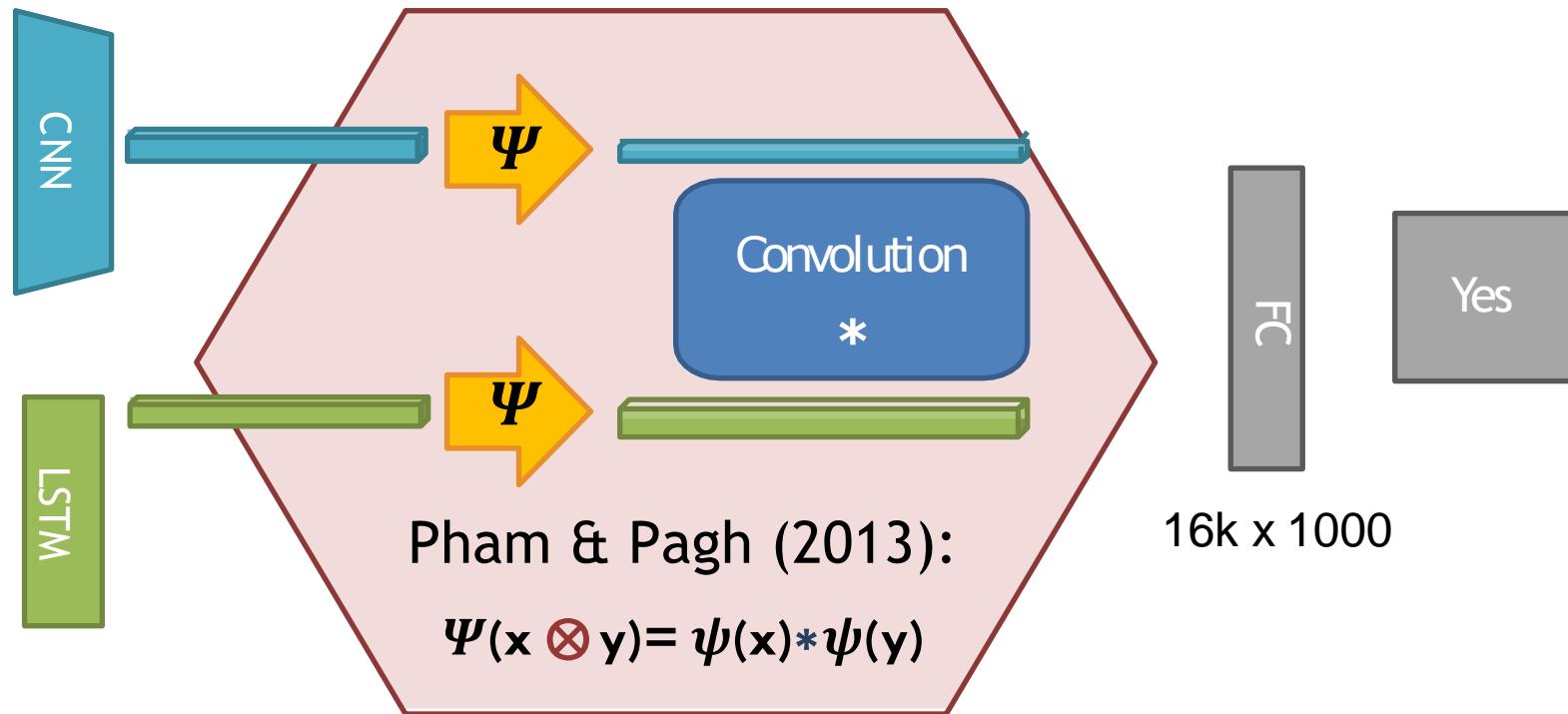


- Tous les éléments peuvent interagir
- Interaction multiplicative
- Nombre élevé d'activations et de calculs
- Nombre élevé de paramètres

[Countsketch] M. Charikar, K. Chen, M. Farach-Colton. Finding frequent items in data streams. Automata, languages and programming 2002.

[Pham&Pagh 13] N. Pham, R. Pagh. Fast and scalable polynomial kernels via explicit feature maps. KDD 2013

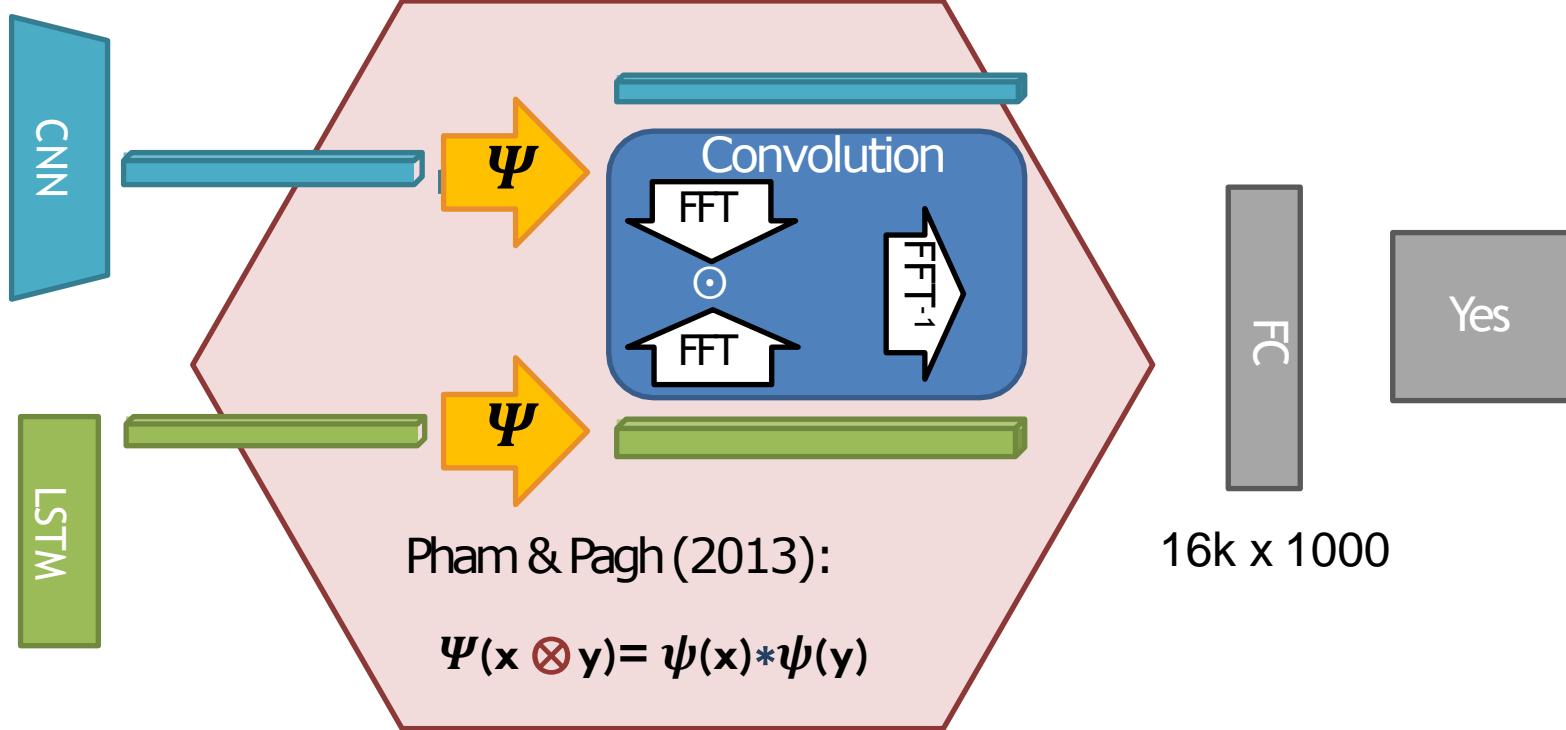
FUSION DE CARACTÉRISTIQUES



- Tous les éléments peuvent interagir
- Interaction multiplicative
- Nombre élevé d'activations et de calculs
- Nombre élevé de paramètres

[Countsketch] M. Charikar, K. Chen, M. Farach-Colton. Finding frequent items in data streams. Automata, languages and programming 2002.
[Pham&Pagh 13] N. Pham, R. Pagh. Fast and scalable polynomial kernels via explicit feature maps. KDD 2013

FUSION DE CARACTÉRISTIQUES



- Tous les éléments peuvent interagir
- Interaction multiplicative
- Nombre élevé d'activations et de calculs
- Nombre élevé de paramètres

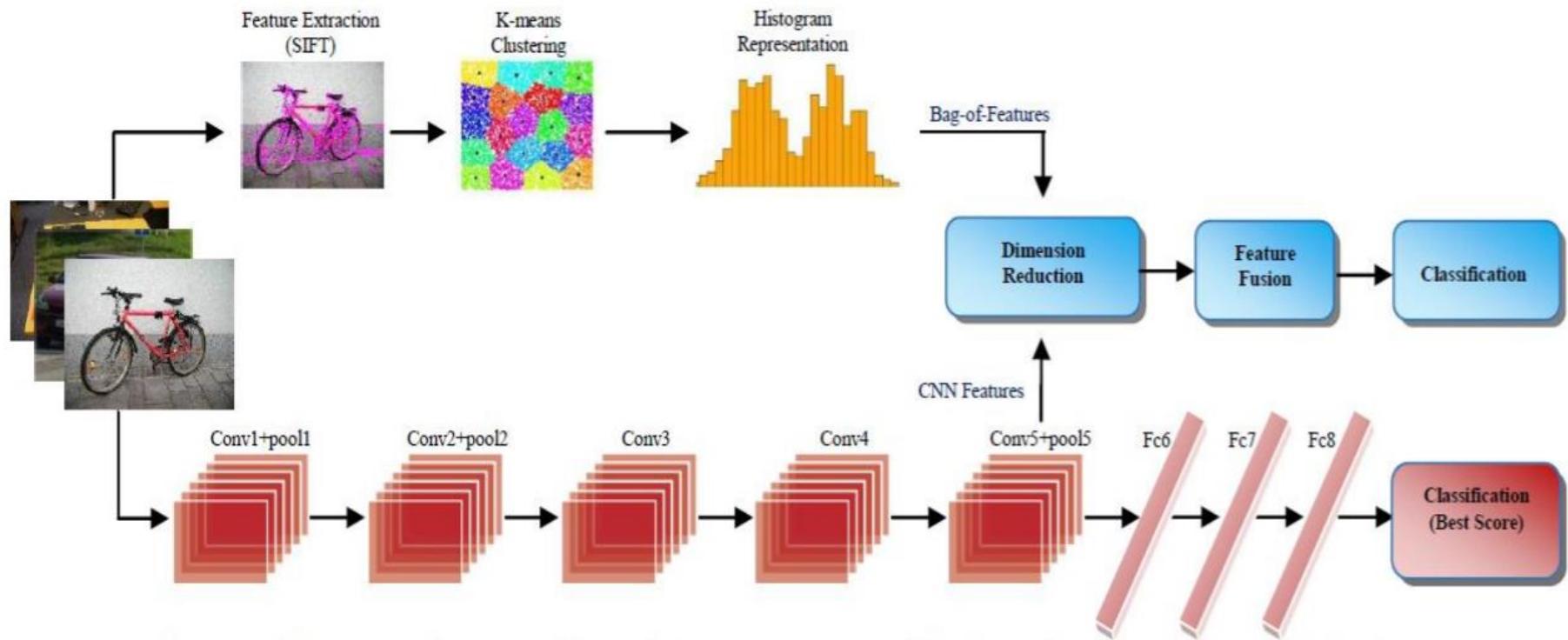
[Countsketch] M. Charikar, K. Chen, M. Farach-Colton. Finding frequent items in data streams. Automata, languages and programming 2002.

[Pham&Pagh 13] N. Pham, R. Pagh. Fast and scalable polynomial kernels via explicit feature maps. KDD 2013

APPLICATIONS DE DEEP LEARNING

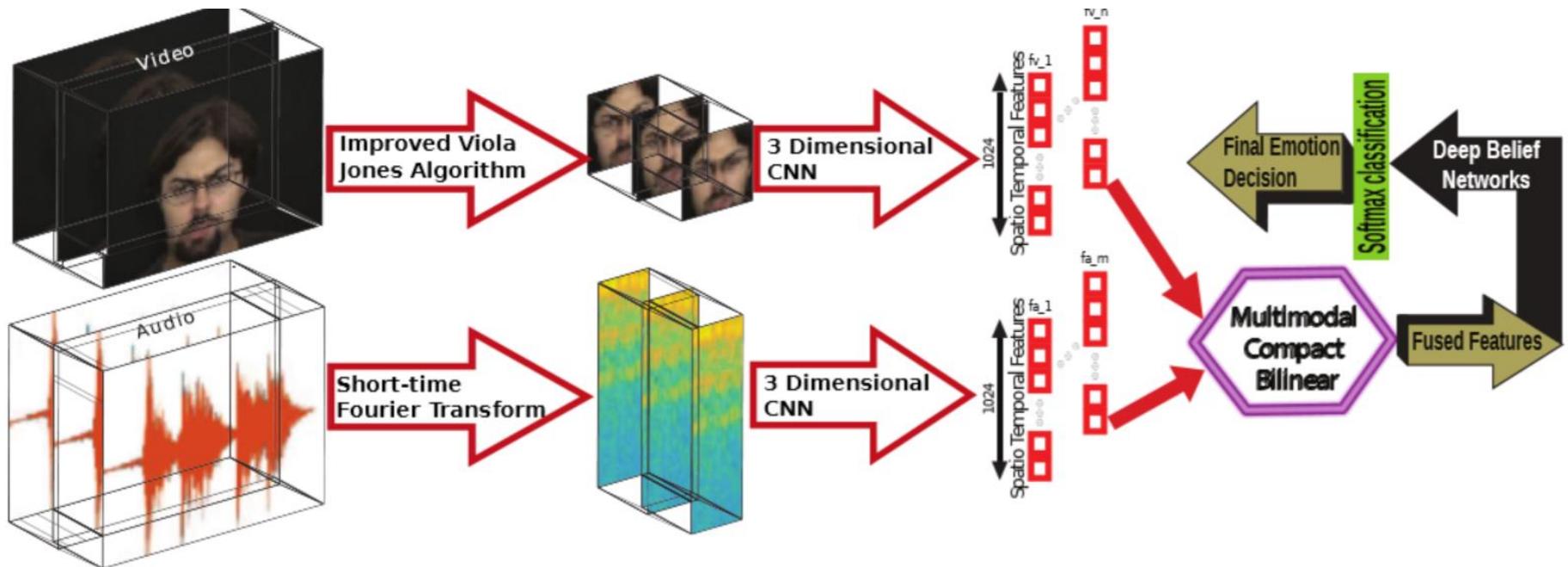
CNN-LSTM-RNN

FUSION DE CARACTÉRISTIQUES



Feature Fusion for Efficient Object Classification Using Deep and Shallow Learning

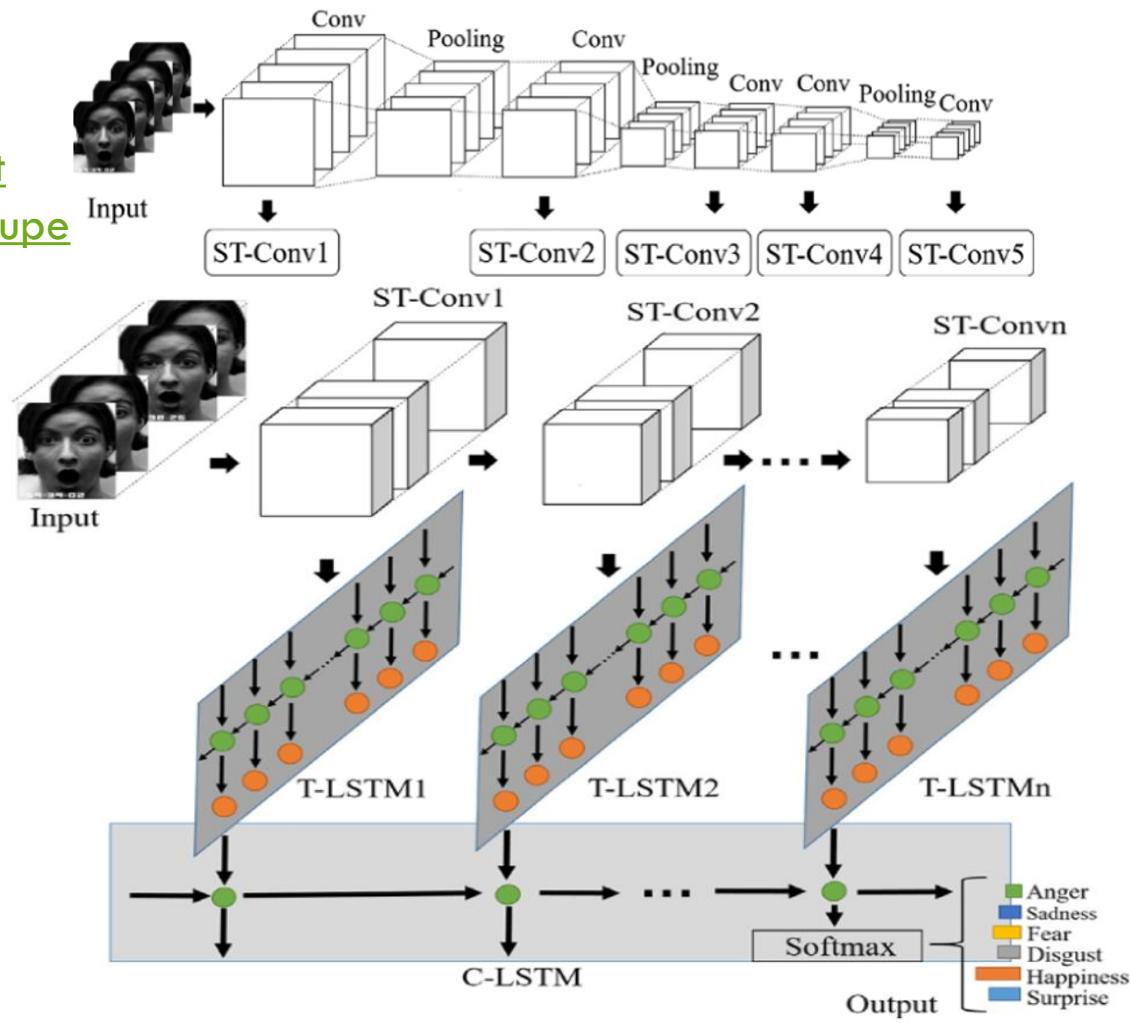
FUSION DE CARACTÉRISTIQUES: 3D-CNN ET LSTM



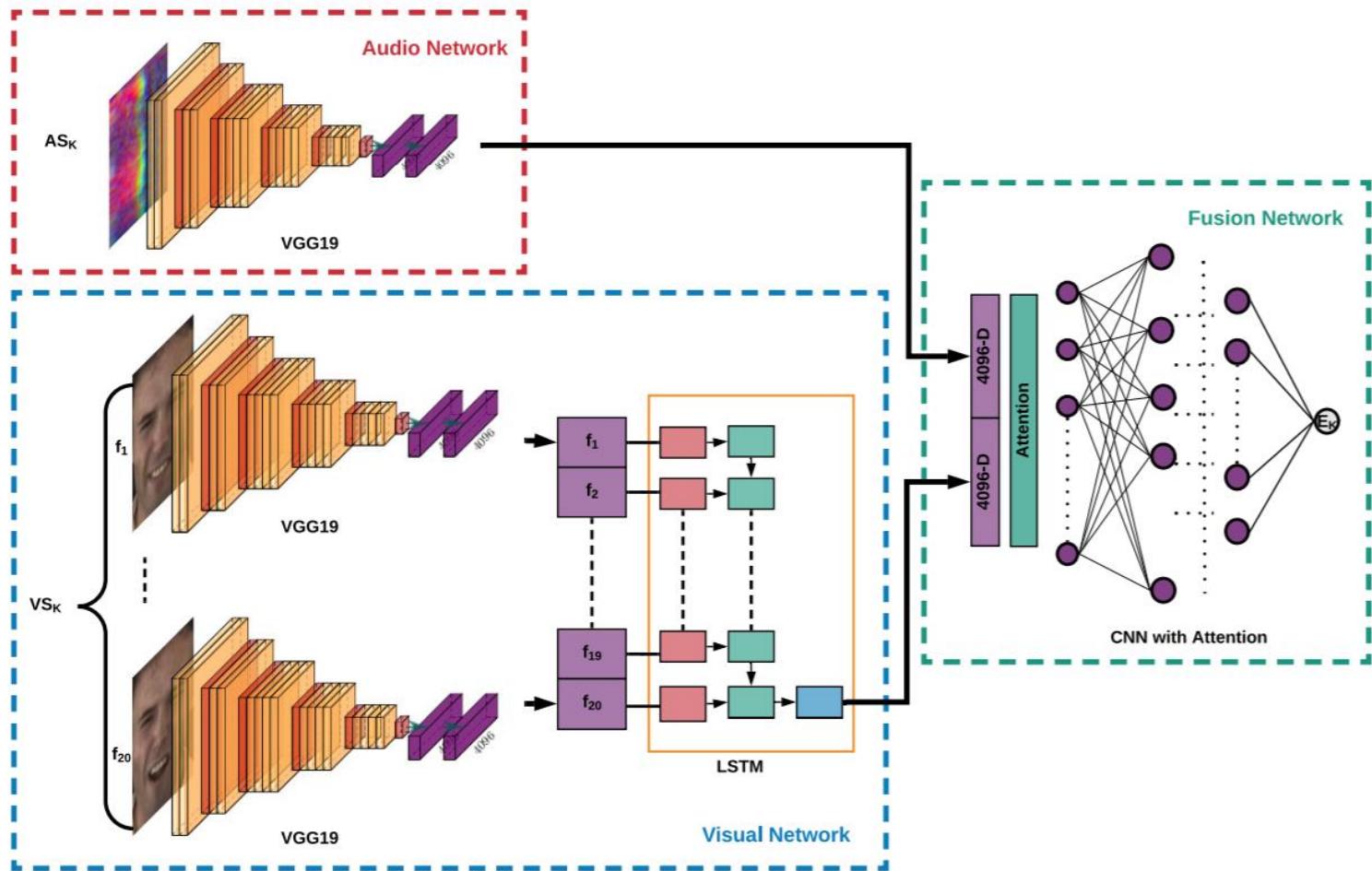
Deep Spatio-Temporal Feature Fusion with Compact Bilinear Pooling For Multimodal Emotion Recognition

3D-CNN ET LSTM POUR LA RECONNAISSANCE DES EXPRESSIONS FACIALES

Architecture basée sur 3D-CC et l'imbrication de LSTM, qui regroupe LSTM temporelle T-LSTL et convolutionnelle C-LSTM).

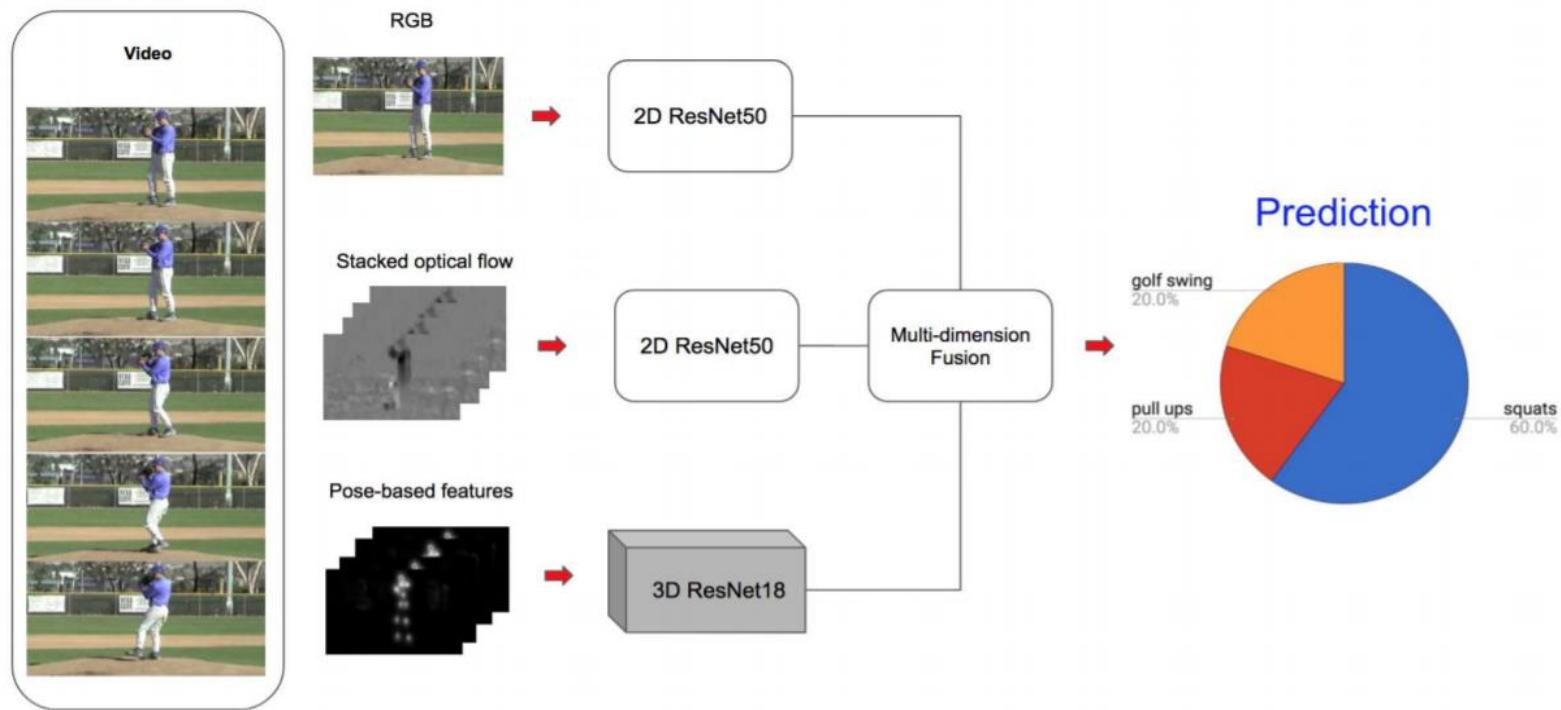


CNN+LSTM



Learning Salient Features for Multimodal Emotion Recognition with Recurrent Neural Networks and Attention Based Fusion

RECONNAISSANCE DE L'ACTION HUMAINE BASÉE SUR CNN ET LA FUSION MULTIDIMENSIONNELLE.



Overview of the temporal pose-based convolutional neural network with multidimensional fusion



YOLO9000: Better, Faster, Stronger

18th July, 2017

San Lee

Redmon, Joseph, et al. "YOLO9000:
Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017."

NOUVELLES AVANCÉES DE LA DÉTECTION
ET LA RECONNAISSANCE D'OBJETS

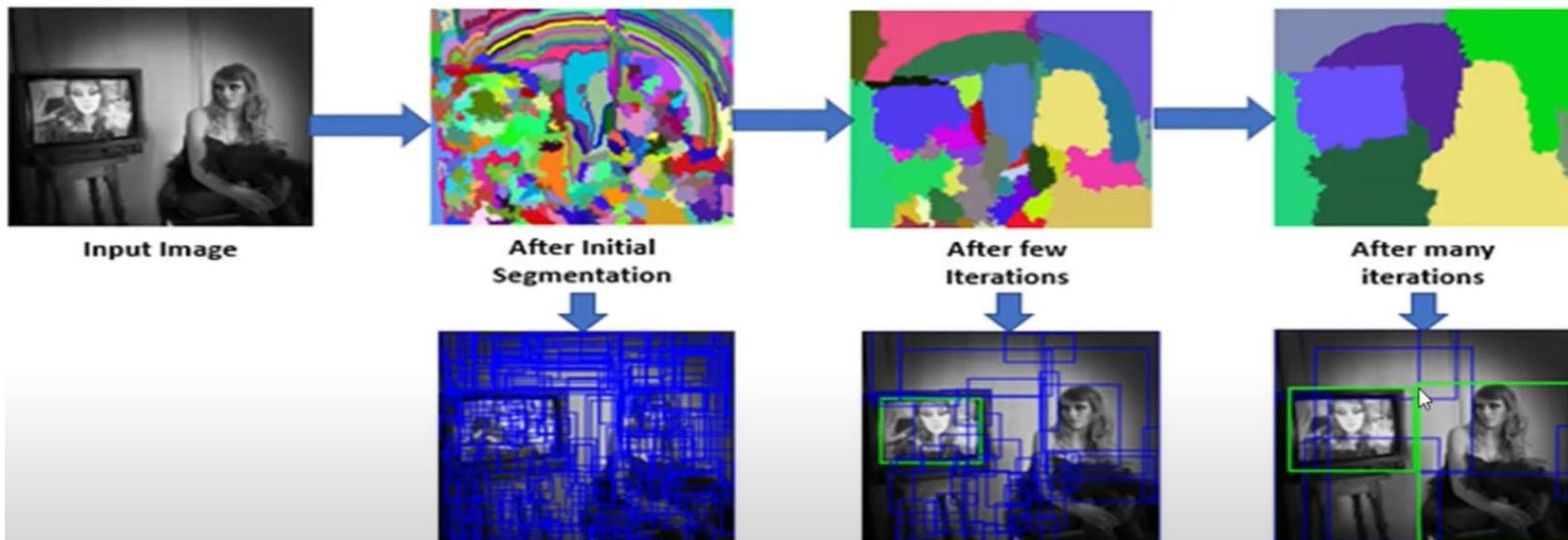
LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS

R-CNN

En 2014, les régions avec des caractéristiques CNN (R-CNN) étaient une bouffée d'air pour la détection d'objets et la segmentation sémantique, car les méthodes de pointe précédentes exigeaient beaucoup de puissance de calcul et s'appuyaient principalement sur des caractéristiques de bas niveau, telles que les arêtes, les gradients et les coins.

LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: RECHERCHE SELECTIVE

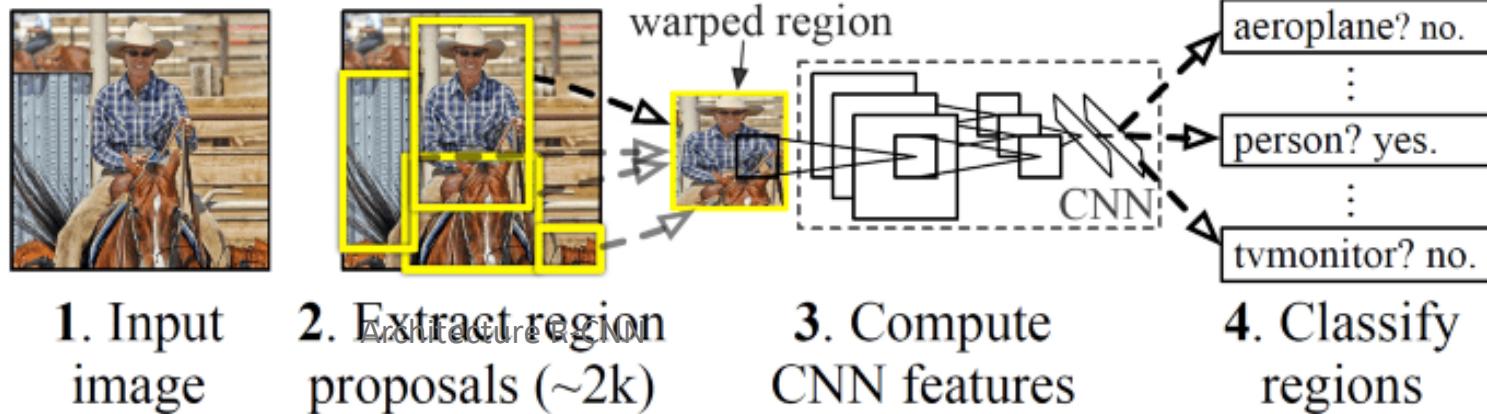
Le système R-CNN est composé de trois modules principaux. Le module le plus avancé extrait environ 2000 propositions de régions à l'aide d'un algorithme de segmentation appelé recherche sélective, pour déterminer quelles parties d'une image sont les plus susceptibles de contenir un objet. L'algorithme scanne l'image avec des fenêtres de différentes échelles et cherche les pixels adjacents qui partagent des couleurs et des textures.



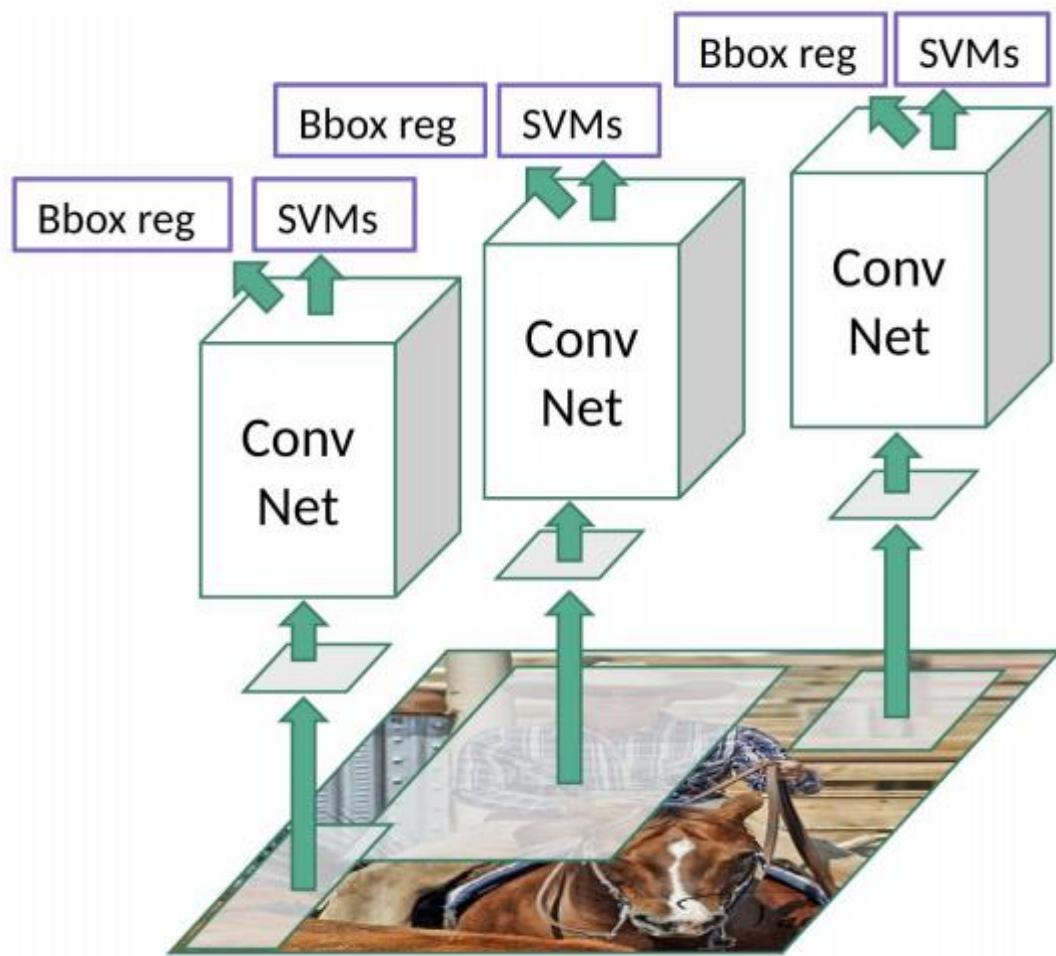
LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: CLASSIFICATION

Le deuxième module est un grand réseau de neurones convolutif qui extrait un vecteur de caractéristiques de longueur fixe de chaque proposition renvoyée par la recherche sélective. Indépendamment de la taille ou du ratio. La région candidate subit une déformation de l'image pour avoir la taille d'entrée requise.

Enfin, le dernier module classe chaque région avec des SVM linéaires.



LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: ARCHITECTURE



R-CNN est très lent à entraîner et à tester, et pas très précis selon les normes actuelles. Néanmoins, il s'agit d'une méthode essentielle qui a ouvert la voie à **Fast R-CNN**, et à l'état-de-l'art actuel **Faster R-CNN** et **Mask R-CNN**.

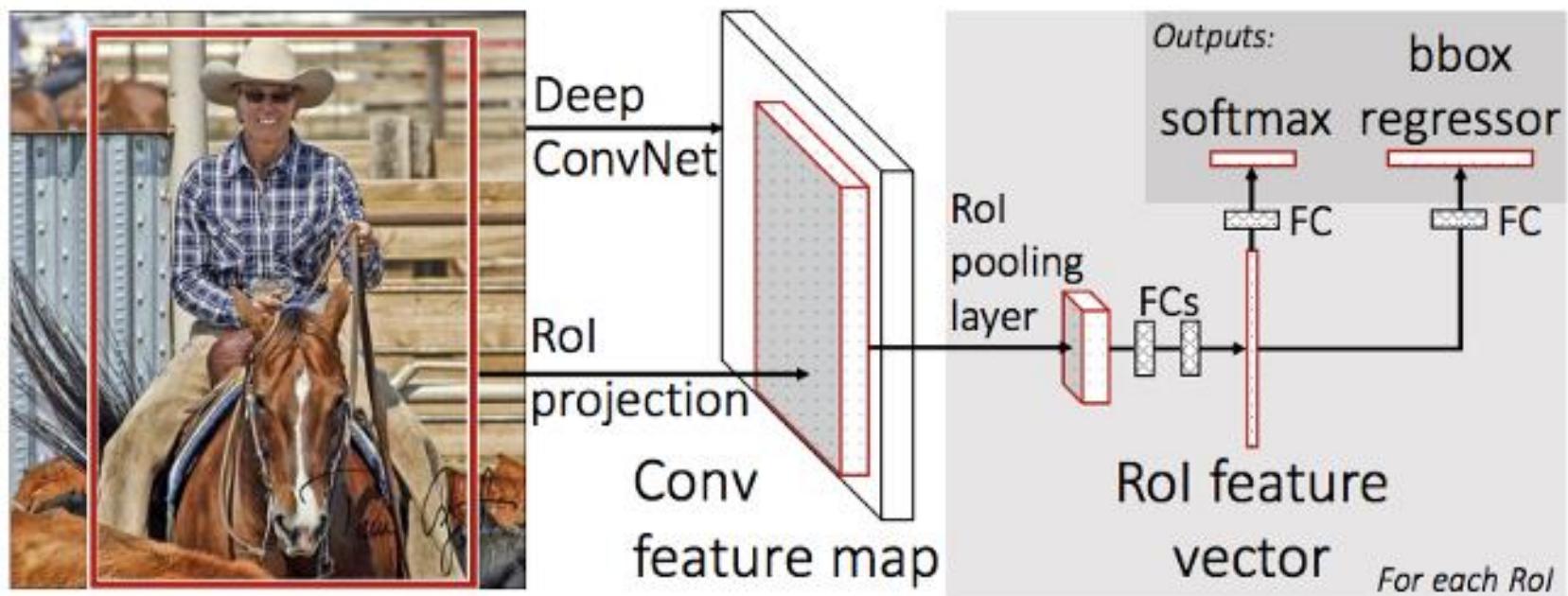
LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: FAST R-CNN

Une grande amélioration par rapport à R-CNN est qu'au lieu de faire ~ 2000 forward pour chaque proposition de région, Fast R-CNN calcule une carte de caractéristiques (Feature Maps) pour l'image d'entrée en un seul forward du réseau, ce qui le rend beaucoup plus rapide.

L'entrée pour Fast R-CNN est une image, ainsi qu'un ensemble de propositions d'objets. Tout d'abord, ils sont passés à travers un réseau entièrement convolutif pour obtenir la carte des caractéristiques. Ensuite, pour chaque proposition d'objet, un vecteur de caractéristiques de longueur fixe est extrait de la carte des caractéristiques à l'aide d'une couche ROI pooling .

Fast R-CNN mappe chacun de ces ROI dans un vecteur de caractéristiques à l'aide de couches entièrement connectées, pour finalement produire la probabilité softmax et la boîte englobante, qui sont respectivement la classe et la position de l'objet.

LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: ARCHITECTURE FAST R-CNN



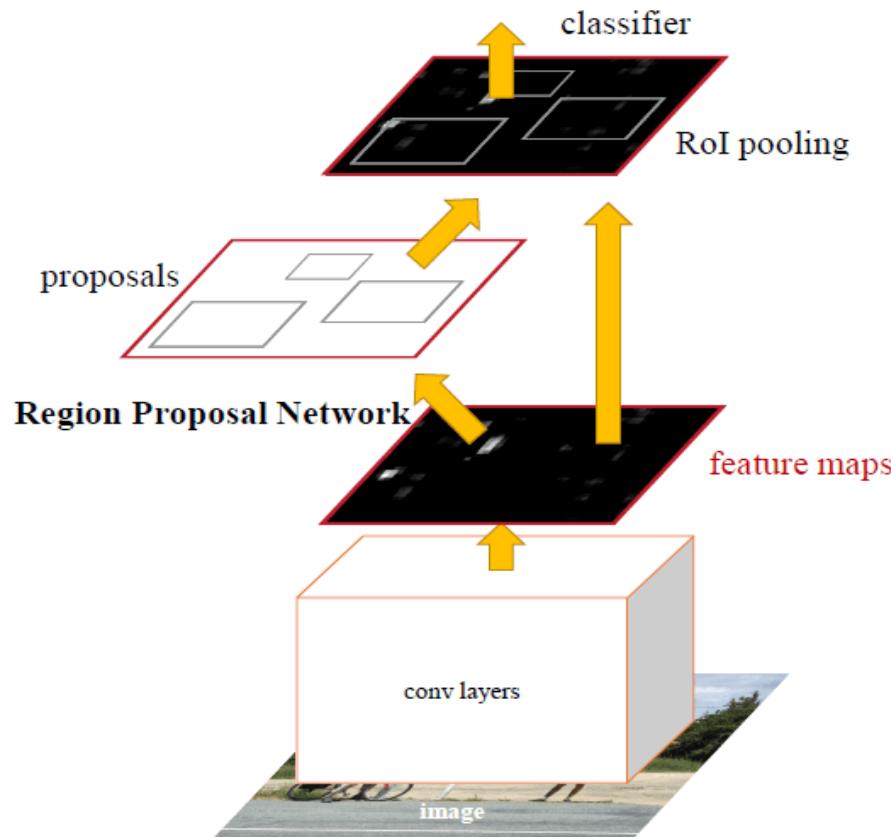
LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS

Faster R-CNN

Il s'avère que Fast R-CNN est encore assez lent, et c'est principalement parce que le CNN est bloqué par l'algorithme de la **recherche sélective**. Faster R-CNN résout ce problème en abandonnant la méthode traditionnelle de proposition de région et en s'appuyant sur une approche d'apprentissage en profondeur. Il se compose de deux modules : un CNN appelé Region Proposal Network (RPN) et le détecteur Fast R-CNN. Les deux modules sont fusionnés en un seul réseau.

La création des régions de propositions se fait en faisant glisser un petit réseau sur la dernière couche de convolution partagée du réseau. Le petit réseau nécessite une fenêtre($n \times n$) de la carte de caractéristiques en entrée. Chaque fenêtre glissante est mappée sur une entité de dimension inférieure, donc comme avant, elle est alimentée vers deux couches entièrement connectées : une couche de classification de boîte et une couche de régression de boîte.

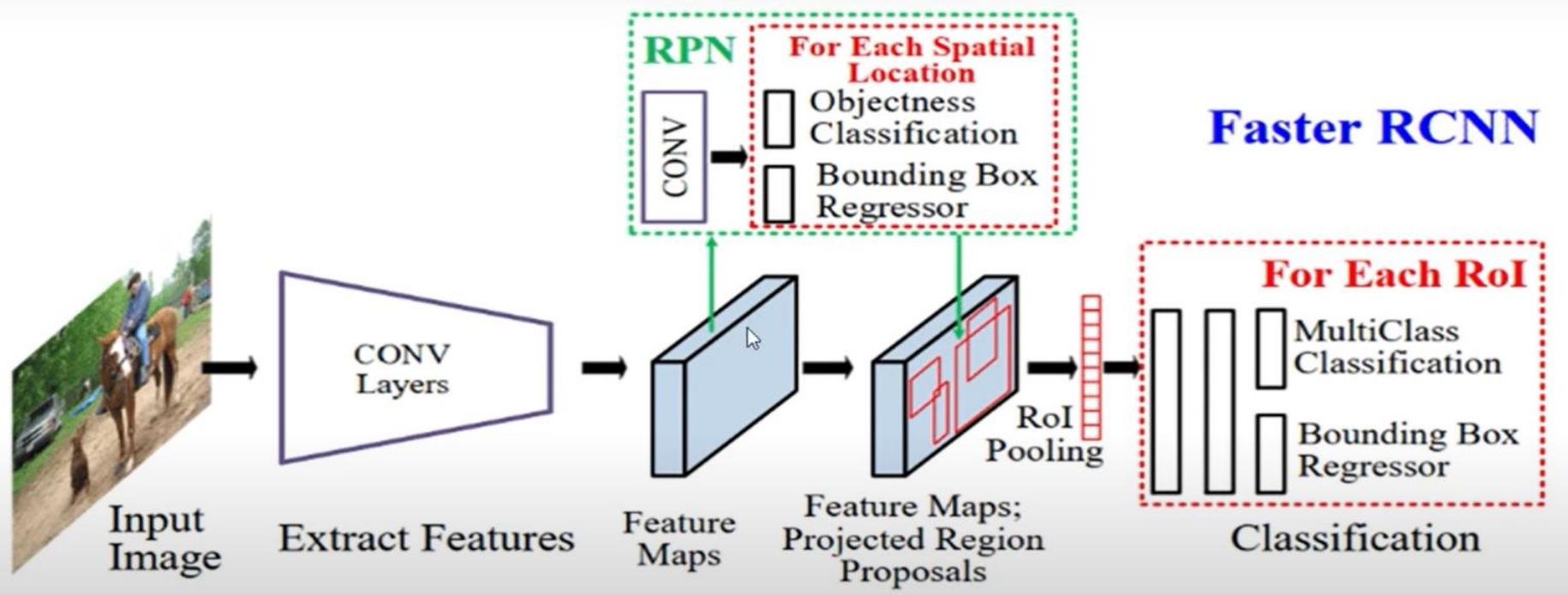
LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS



Faster R-CNN framework

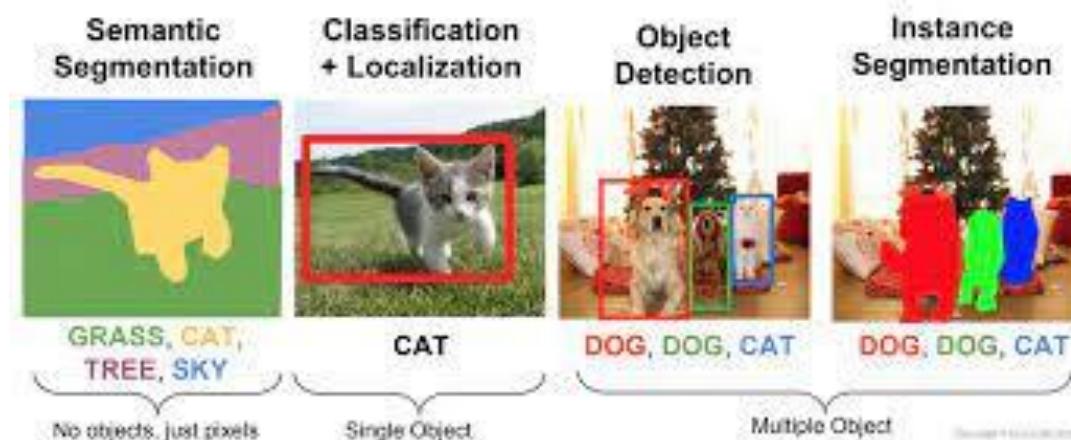
LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: ARCHITECTURE FASTER RCNN

Regional Proposal Network



LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: MASK R-CNN

Le R-CNN plus rapide est considéré comme à la pointe de la technologie et constitue certainement l'une des meilleures options pour la détection d'objets. Cependant, il ne fournit pas de segmentation sur les objets détectés, c'est-à-dire qu'il n'est pas capable de localiser les pixels exacts de l'objet, mais uniquement la boîte englobante qui l'entoure. Dans de nombreux cas, ce n'est pas nécessaire, mais quand c'est le cas, Mask R-CNN devrait être le premier à venir à l'esprit.



LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: MASK R-CNN

Les auteurs de Mask R-CNN de Facebook AI Research (FAIR) ont étendu Faster R-CNN pour effectuer une segmentation d'instance, ainsi que la classe et la boîte englobante. La segmentation d'instance est une combinaison de détection d'objet et de segmentation sémantique, ce qui signifie qu'elle effectue à la fois la détection de tous les objets d'une image et la segmentation de chaque instance tout en la différenciant du reste des instances.



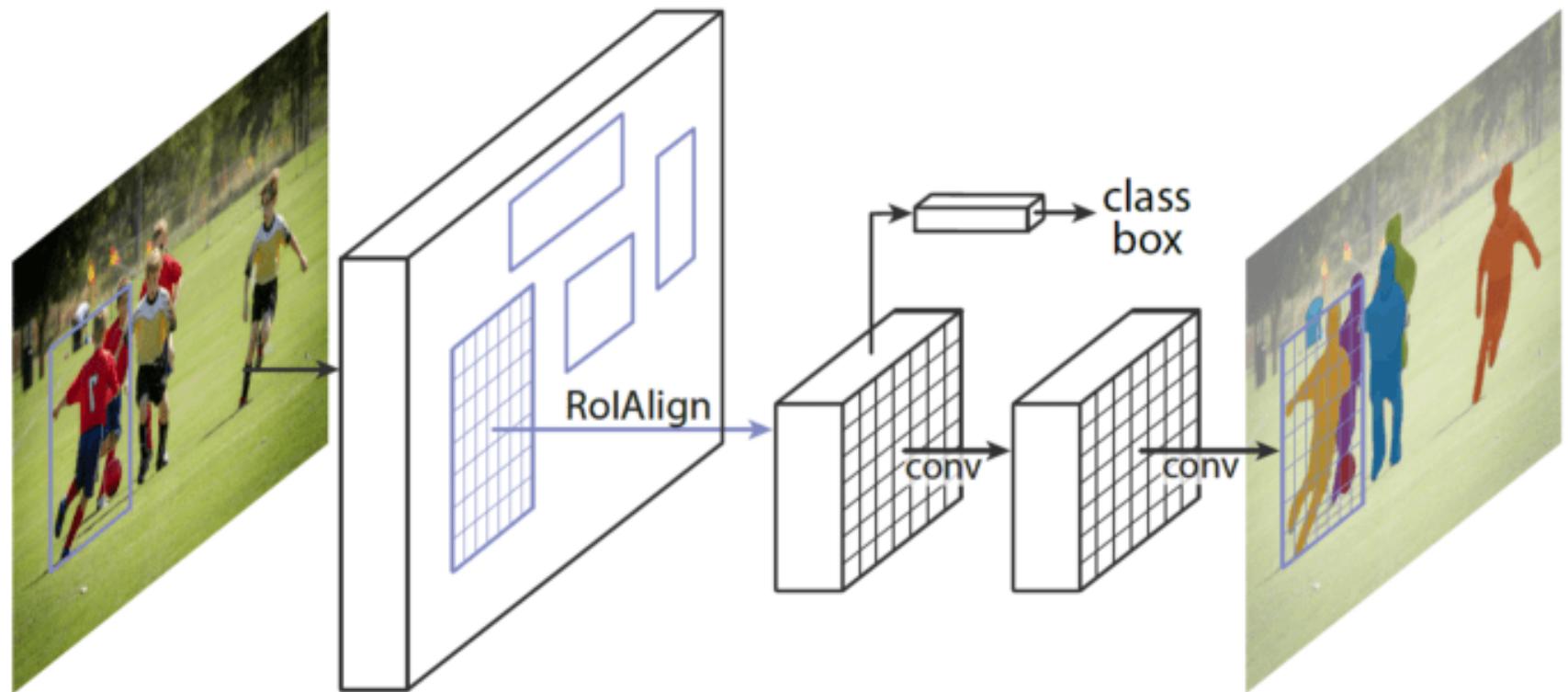
LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: MASK R-CNN

La première étape (proposition de région) de Mask R-CNN est identique à son prédécesseur.

Pour la deuxième étape, on génère un masque binaire pour chaque ROI en parallèle à la classe et à la boîte englobante. Ce masque binaire indique si le pixel fait partie de n'importe quel objet, sans se soucier des catégories.

La classe des pixels serait attribuée simplement par la boîte englobante dans laquelle ils résident, ce qui rend le modèle beaucoup plus facile à entraîner.

LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS



Mask R-CNN framework

LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: YOLO

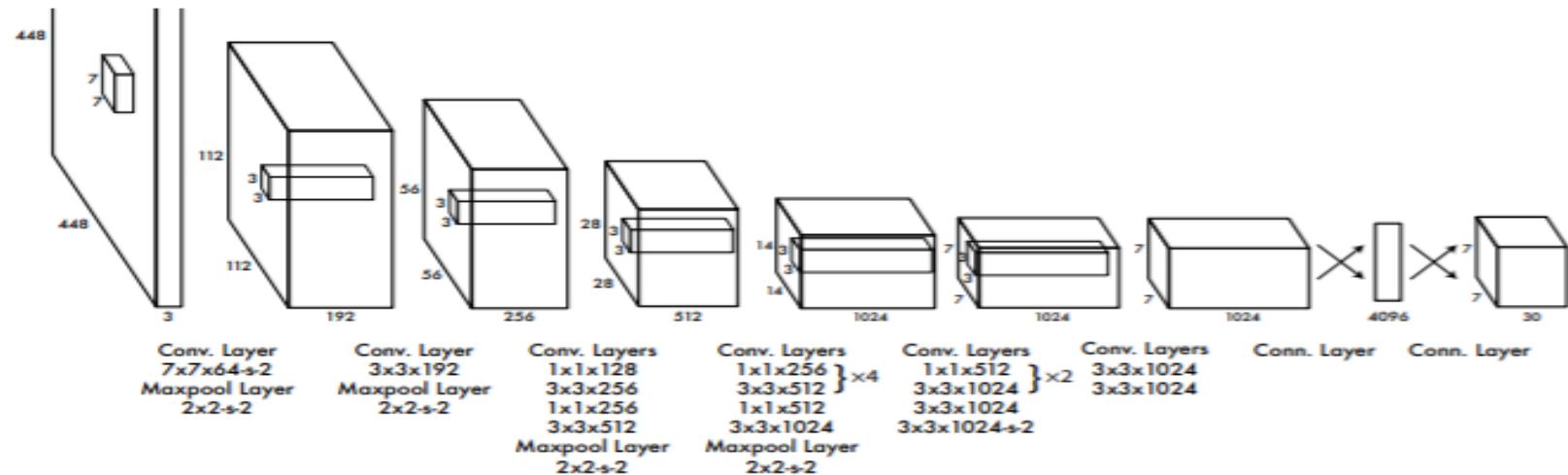
Nous basculons d'une solution axée sur la précision à une solution axée sur la vitesse. You Only Look Once (YOLO) est la méthode de détection d'objet la plus populaire aujourd'hui. Il est capable de traiter des vidéos en temps réel avec un délai minimal, tout en conservant une précision respectable. Et comme son nom l'indique, il n'a besoin que d'une seule propagation Forward pour détecter tous les objets d'une image.

YOLO est conçu à base de Darknet, un framework de réseau neuronal open source écrit en C et CUDA, développé par le même auteur qui a créé YOLO, Joseph Redmon.

LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: YOLO



LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: ARCHITECTURE YOLO



Cette architecture prend une image en entrée et la redimensionne à 448*448.

Cette image est ensuite transmise dans le réseau CNN.

Ce modèle comporte 24 couches de convolution, 4 couches max-pooling suivies de 2 couches entièrement connectées .

LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: DÉTECTION

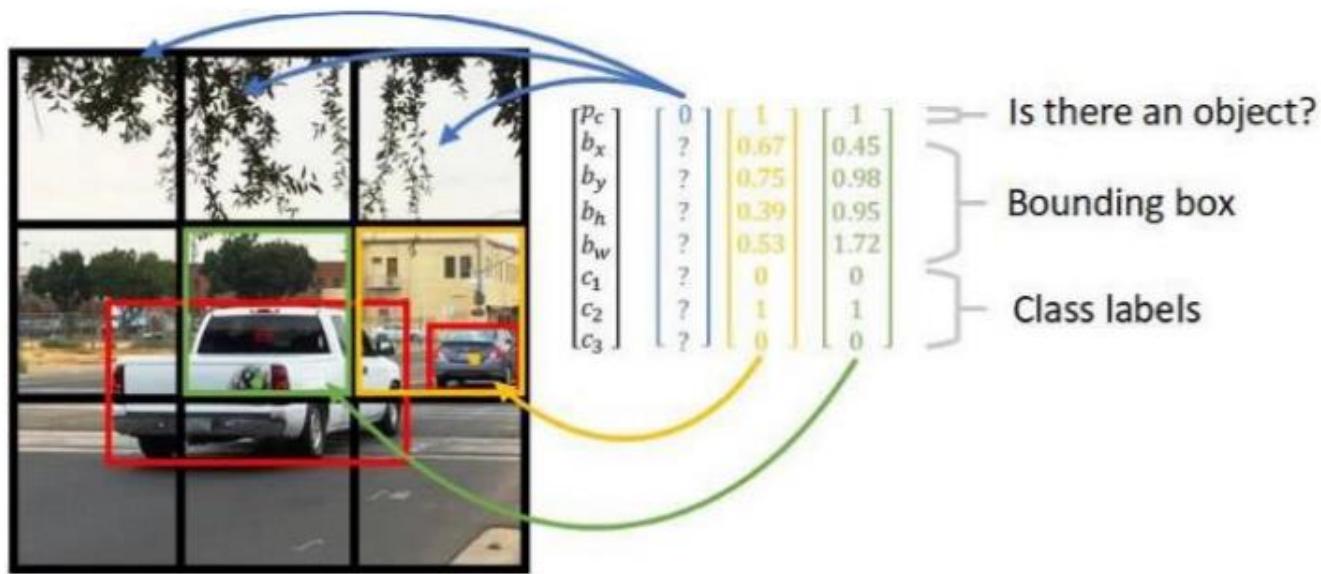
Cette architecture divise l'image en une grille de taille $S \times S$. Si le centre de la boîte englobante de l'objet se trouve dans cette cellule, alors cette cellule est responsable de la détection de cet objet. Chaque cellule prédit B boîtes englobantes avec leur score de confiance. Chaque score de confiance montre à quel point il est précis que la boîte englobante prédite contient un objet et avec quelle précision il prédit les coordonnées de la boîte englobante par rapport à la prédiction de la vérité terrain.



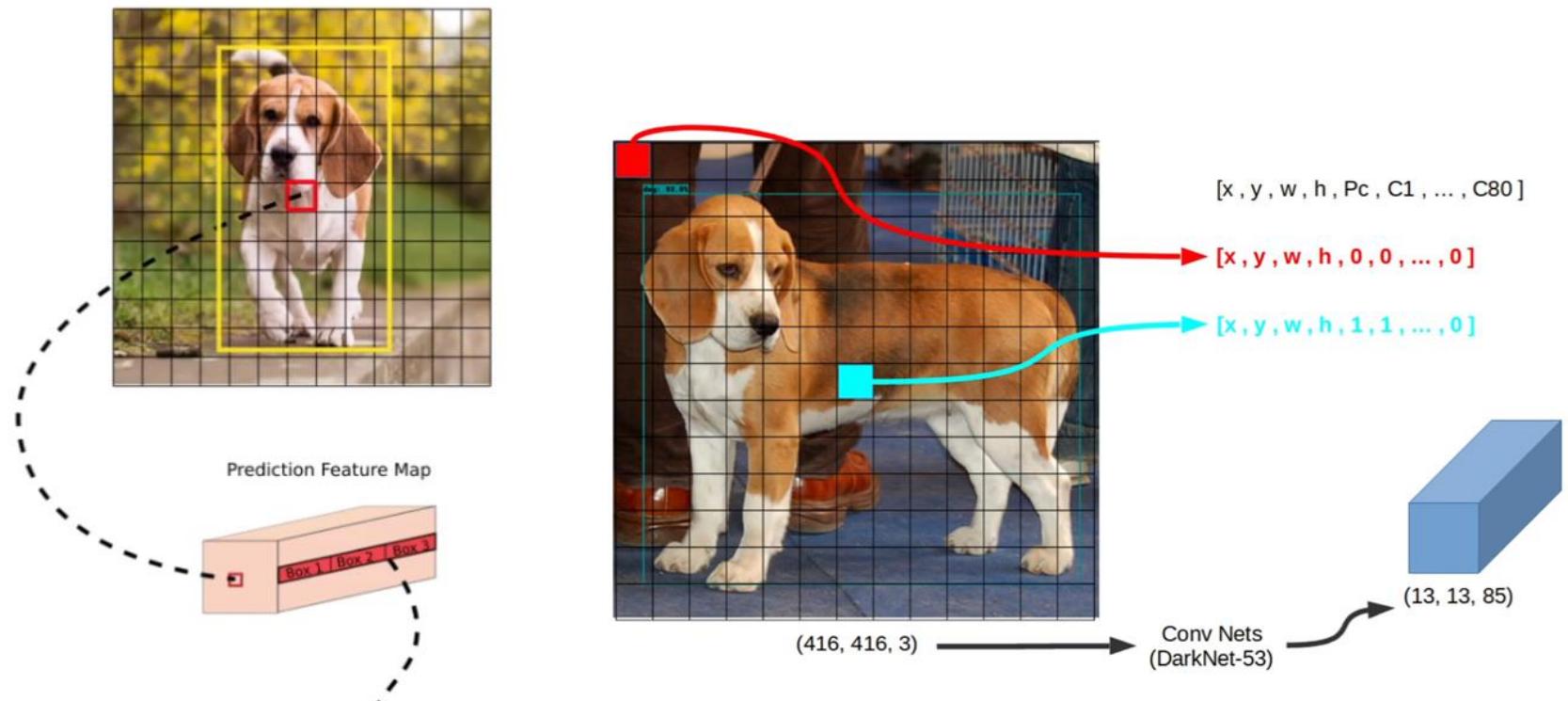
$$\text{Pr}(Object) * IOU_{pred}^{truth}$$

LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: ARCHITECTURE YOLO

L'algorithme applique un seul réseau de neurones à l'image d'entrée. Ensuite, ce réseau divise cette image en régions qui fournissent les boîtes englobantes et prédisent également les probabilités pour chaque région. Ces boîtes englobantes générées sont pondérées par les probabilités prédites.



LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: ARCHITECTURE YOLO



t_x	t_y	t_w	t_h	p_o	p_1	p_2	p_c
-------	-------	-------	-------	-------	-------	-------	------	-------

Box Co-ordinates

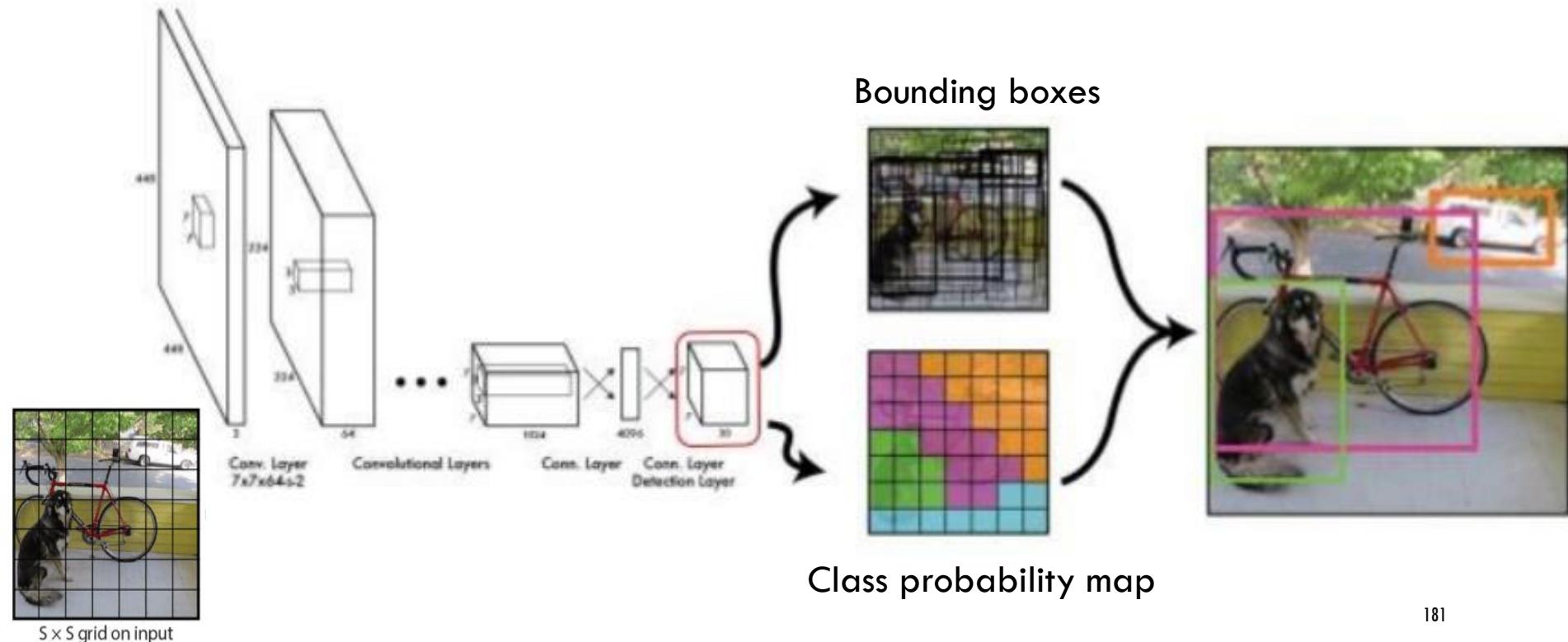
Objectness Score

Class Scores

$\times B$

LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: ARCHITECTURE YOLO

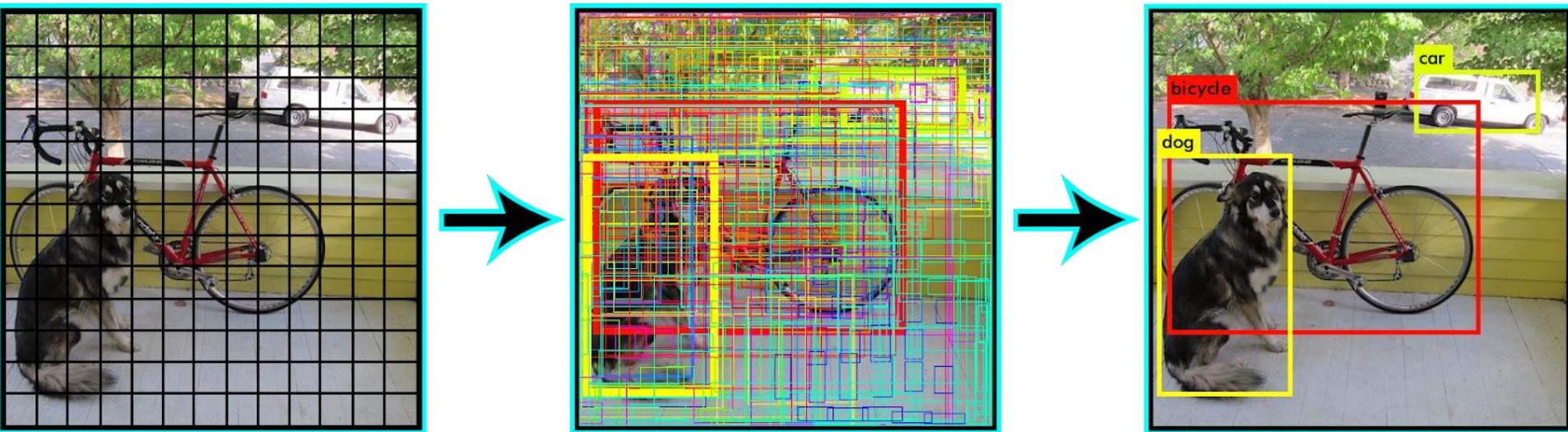
YOLO: You Only Look Once



LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: SUPPRESSION DES FAUX BOITES

Cela génère de nombreuses prédictions en double en raison de plusieurs cellules prédisant le même objet avec différentes prédictions de boîte englobante.

YOLO utilise la suppression non maximale pour résoudre ce problème.



LES NOUVELLES AVANCÉES DE LA DÉTECTION D'OBJETS: SUPPRESSION DES FAUX BOITES

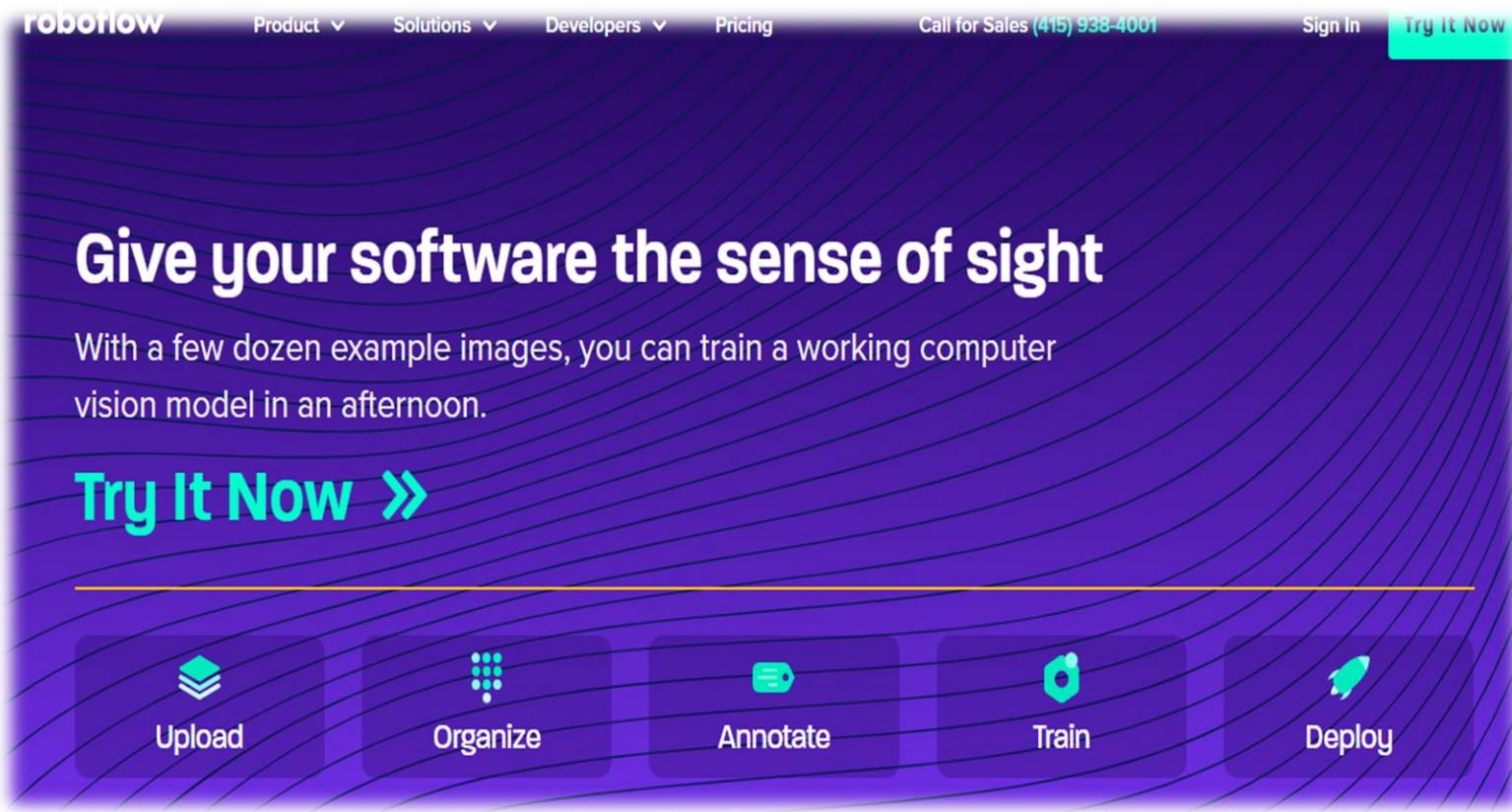
Dans la suppression non maximale, YOLO supprime toutes les boîtes englobantes qui ont des scores de probabilité inférieurs.

YOLO y parvient en examinant d'abord les scores de probabilité associés à chaque décision et en prenant le plus grand. Suite à cela, il supprime les boîtes englobantes ayant la plus grande Intersection sur l'Union (IoU) avec la boîte englobante à haute probabilité actuelle.

La technique de suppression non-max (non-max suppression) garantit que l'algorithme de détection d'objet ne détecte chaque objet qu'une seule fois et qu'il rejette toute fausse détection, il donne ensuite les objets reconnus avec les boîtes englobantes.

INTRODUCTION TO ROBOFLOW AND YOLO TOOLS IN THE FIELD OF IMAGE PROCESSING

ROBOFLOW



The image shows the homepage of the Roboflow website. The header features the Roboflow logo, navigation links for Product, Solutions, Developers, Pricing, and a call-to-action button "Try It Now". Below the header, a large purple banner with white text reads "Give your software the sense of sight" followed by a subtext: "With a few dozen example images, you can train a working computer vision model in an afternoon." A prominent blue button labeled "Try It Now >" is centered below the subtext. At the bottom, there is a horizontal bar with five icons: Upload (stacked boxes), Organize (grid), Annotate (tag), Train (blue circle), and Deploy (green rocket). The background of the page has a subtle grid pattern.

roboflow

Product ▾ Solutions ▾ Developers ▾ Pricing

Call for Sales (415) 938-4001

Sign In Try It Now

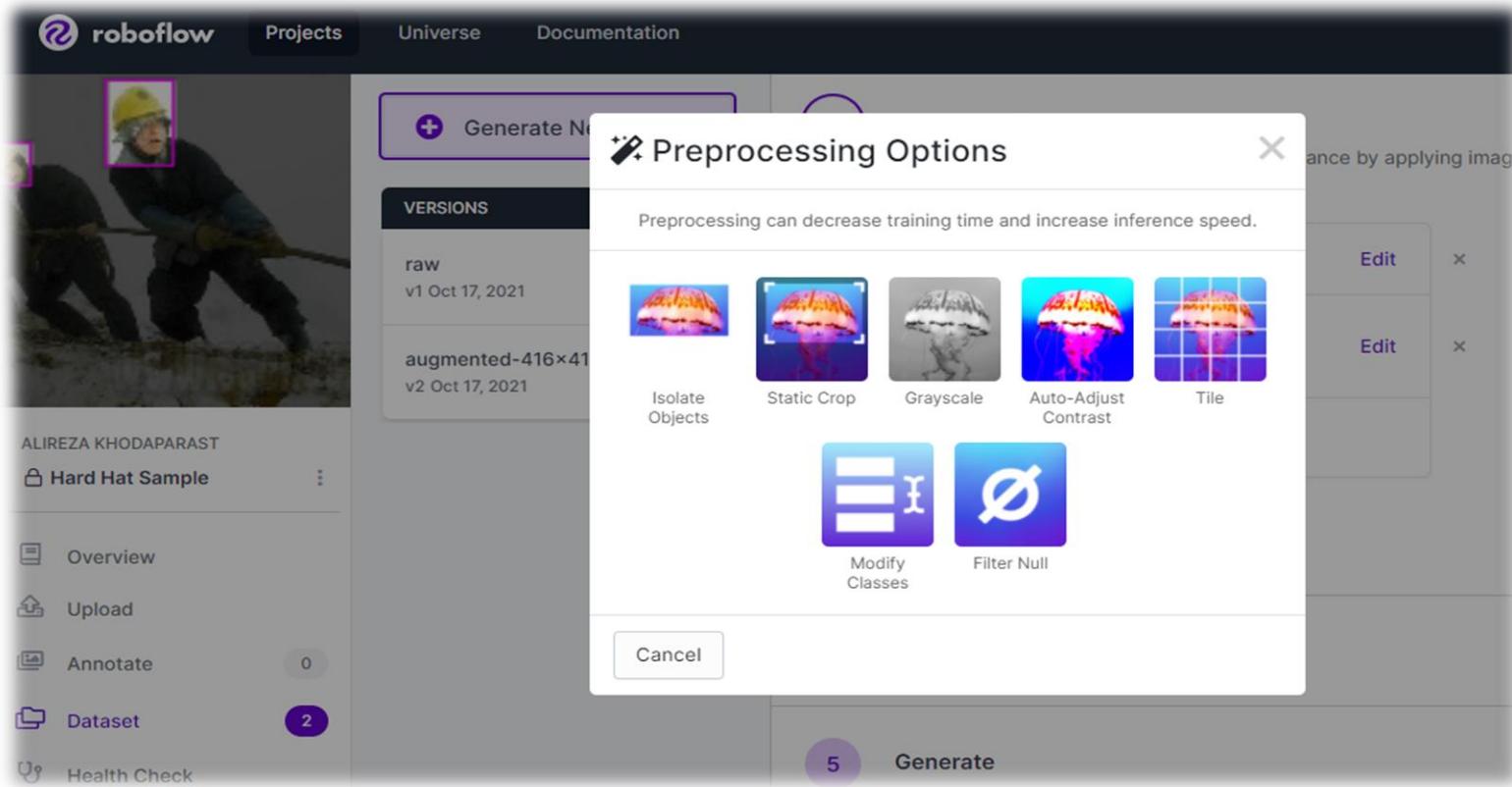
Give your software the sense of sight

With a few dozen example images, you can train a working computer vision model in an afternoon.

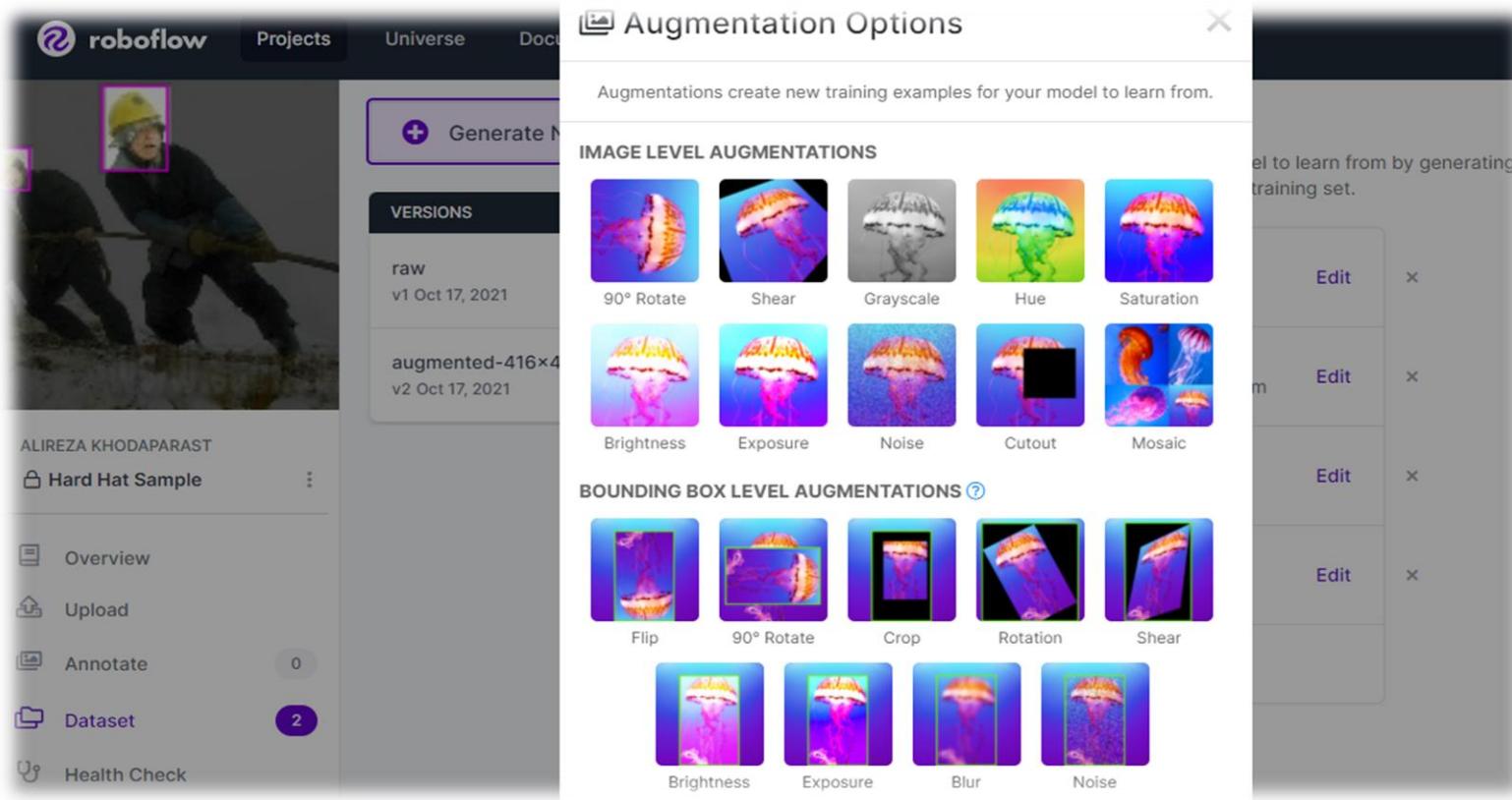
Try It Now >

Upload Organize Annotate Train Deploy

OPTIONS DE PRÉTRAITEMENT



OPTIONS D'AUGMENTATION DU DATASET



VERIFICATION DU DATASET

roboflow Projects Universe Documentation Alireza Khodaparast

Hard Hat Sample » Dataset Health Check

Generated on July 03, 2020 at 11:28 pm. [Regenerate](#)

Images 100 0 missing annotations 0 null examples	Annotations 386 3.9 per image (average) across 3 classes	Average Image Size 0.17 mp from 0.05 mp to 0.61 mp	Median Image Ratio 500×333 wide
--	--	--	--

At the top, you'll see fast stats about the size and makeup of your dataset.

Skip Back Next →

over represented
under represented
under represented

Dimension Insights

Size Distribution

The purple box indicates the median width by median height image (500×333).

small	1
medium	90
large	9



COMMENT ENTRAÎNER YOLOV5 SUR DES OBJETS PERSONNALISÉS

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** Roboflow_Custom_YOLOv5.ipynb
- Toolbar:** File, Edit, View, Insert, Runtime, Tools, Help, All changes saved
- Code/Text Buttons:** + Code, + Text
- Section Header:** How to Train YOLOv5 on Custom Objects
- Text Content:** This tutorial is based on the [YOLOv5 repository](#) by [Ultralytics](#). This notebook shows training on **your own custom objects**. Many thanks to Ultralytics for putting this repository together - we hope that in combination with clean data management tools at Roboflow, this technology will become easily accessible to any developer wishing to use computer vision in their projects.
- Section Header:** Accompanying Blog Post
- Text Content:** We recommend that you follow along in this notebook while reading the blog post on [how to train YOLOv5](#), concurrently.
- Section Header:** Steps Covered in this Tutorial
- Text Content:** In this tutorial, we will walk through the steps required to train YOLOv5 on your custom objects. We use a [public blood cell detection dataset](#), which is open source and free to use. You can also use this notebook on your own data.
- Text Content:** To train our detector we take the following steps:
- List:**
 - Install YOLOv5 dependencies
 - Download custom YOLOv5 object detection data
 - Write our YOLOv5 Training configuration
 - Run YOLOv5 training
 - Evaluate YOLOv5 performance
 - Visualize YOLOv5 training data
 - Run YOLOv5 inference on test images
 - Export saved YOLOv5 weights for future inference