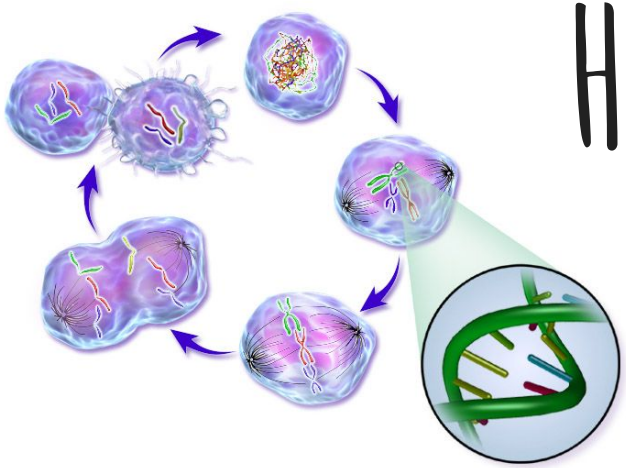


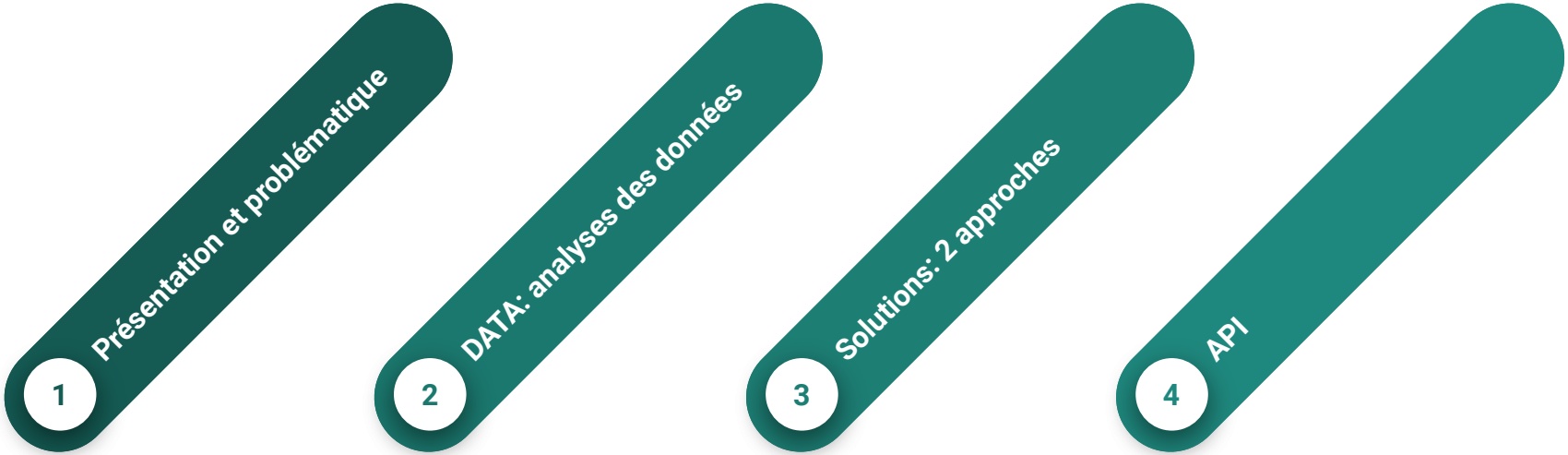
Human Protein Atlas Single Cell Competition

Wael GRIBAA
Amine OMRI
Sabina REXHA
Abdou Akim GOUMBALA
Meryem GASSAB

IA SCHOOL
L'ÉCOLE DE L'INTELLIGENCE ARTIFICIELLE

M2B

SOMMAIRE


1
Présentation et problématique
2
DATA: analyses des données
3
Solutions: 2 approches
4
API

Présentation

PRÉSENTATION

Organisée par human protein Atlas

Problème de segmentation d'instance

Vise à la classification des images unicellulaires

- Aider à caractériser l'hétérogénéité unicellulaire dans la collection d'images
- générant des annotations plus précises des localisations subcellulaires pour des milliers de protéines humaines dans des cellules individuelles.

Objectif ⇨ Prédir les étiquettes de localisation des organites protéiques pour chaque cellule de l'image

Problématique

Cette compétition est de nature **weakly-labeled** : sur la base des labels au niveau de l'image, l'objectif est de construire des modèles pour prédire les labels de chaque cellule individuelle dans l’image.

L'ensemble de test a une variabilité cellulaire plus élevée que l'ensemble d'entraînement et il est annoté pour chaque cellule.

DATA

DATA

En regardant ces données, il semble qu'il s'agisse d'un problème de classification multi-labels...

In [2]:
```
df_train.head()
```

Out[2]:

	ID	Label
0	5c27f04c-bb99-11e8-b2b9-ac1f6b6435d0	8|5|0
1	5fb643ee-bb99-11e8-b2b9-ac1f6b6435d0	14|0
2	60b57878-bb99-11e8-b2b9-ac1f6b6435d0	6|1
3	5c1a898e-bb99-11e8-b2b9-ac1f6b6435d0	16|10
4	5b931256-bb99-11e8-b2b9-ac1f6b6435d0	14|0

DATA

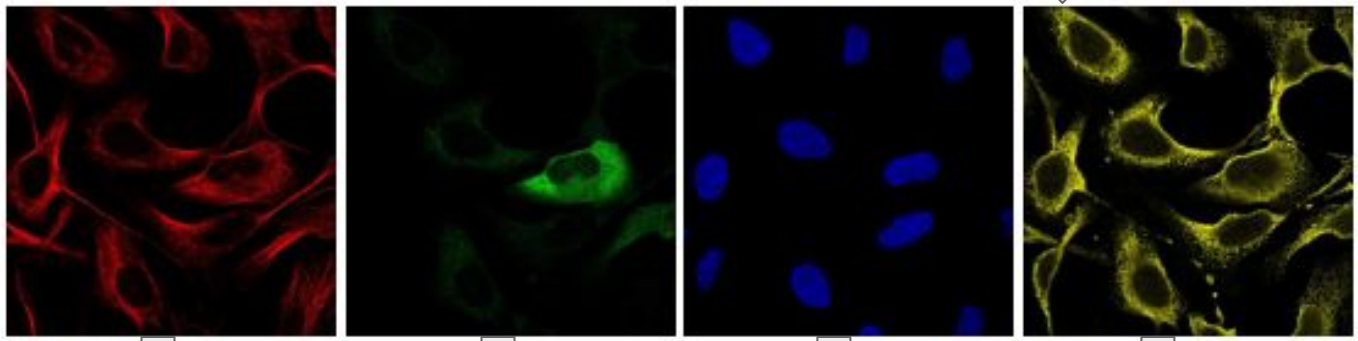
En regardant le modèle de soumission, nous réalisons qu'il ne s'agit pas d'une classification multi-labels, mais plutôt d'une **segmentation d'instance**. Pour chaque image, il est demandé de :

- Segmenter chaque cellule individuelle contenue dans l'image en prédisant le masque de chaque cellule.
- D’identifier la classe de cette cellule.

Certaines cellules peuvent être associées à plusieurs classes.

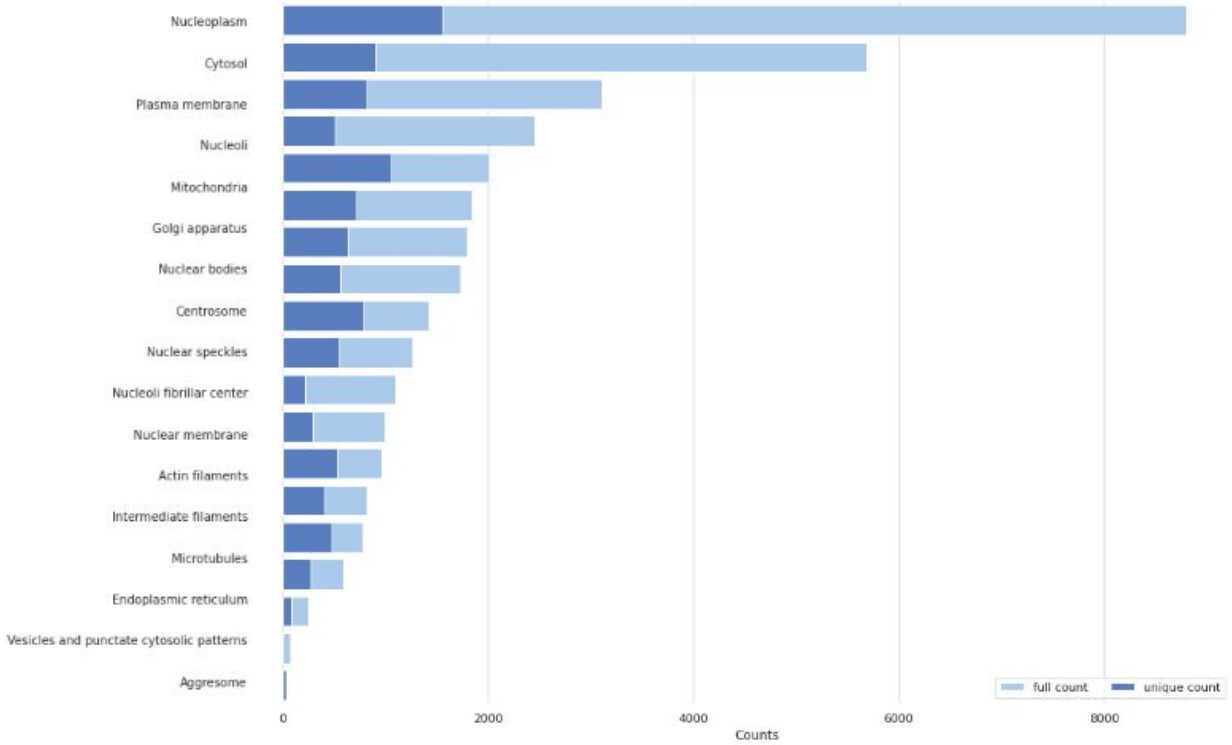
Data

Pour chaque échantillon → **4 fichiers image**

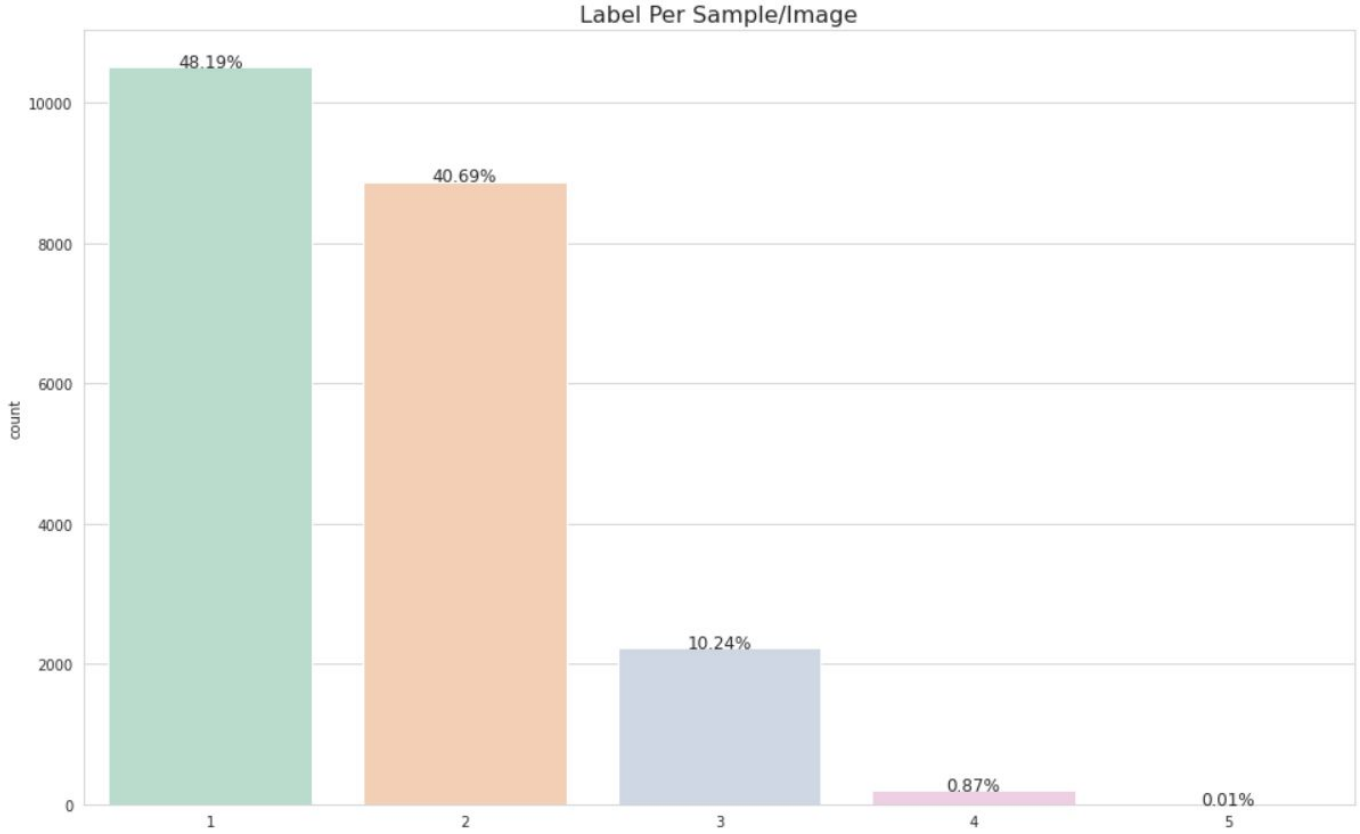


Rouge (Microtubules) | Vert (Protéine) | Bleu (Noyau) | Jaune (Réticulum endoplasmique)

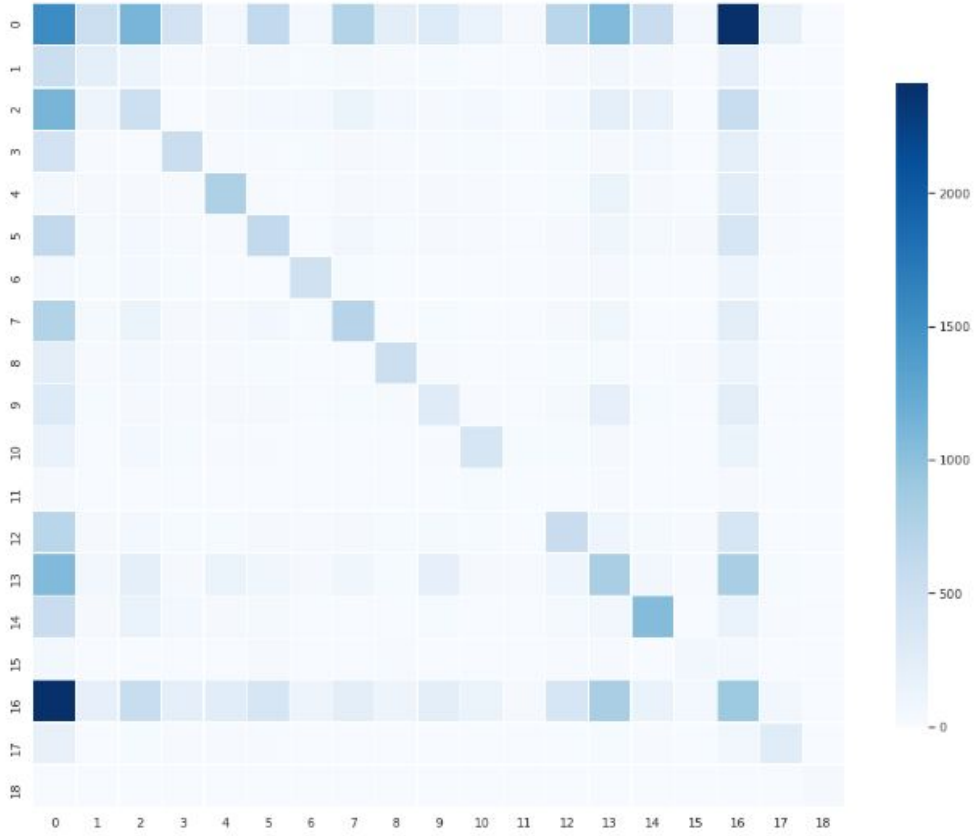
DATA: La distribution des labels



DATA: LES EXEMPLES AVEC MULTI-LABELS



DATA: LES EXEMPLES AVEC DES LABELS INDIVIDUELS



Solutions

Solution Baseline

Nous extrayons les cellules individuelles à l'aide du segmentateur cellulaire HPA, les étiquetons avec le label de l'image et entraînons un classifieur sur ces données.

Solution: Première approche

1. **Prédire les masques cellulaires en utilisant HPA CELLSEG**
   - Prédire le masque des noyaux
   - Prédire le masque de la cellule entière
2. **Créer des données labellisées au format COCO**
   - Obtenir les encodages RLE à partir du masque
   - Créer des Bounding Boxes à partir de la compression RLE de chaque masque dans l'image
3. **Utiliser MMdetection pour entraîner un Mask_rcnn sur le jeu de données customisé en utilisant la configuration par défaut.**
   - Utiliser le COCO LABELED FORMAT et les canaux empilés [rouge, vert, bleu] pour chaque image comme entrée du modèle.
   - Utiliser les données de test pour évaluer le modèle

Solution: Première approche: COCO Labeled Format

```
[27…  {'filename': 'ce1d3a68-bb99-11e8-b2b9-ac1f6b6435d0.jpg',
       'width': 2048,
       'height': 2048,
       'ann': {'bboxes': array([[ 572.,    0.,  788.,  212.],
               [1016.,    0., 1228.,  136.],
               [1928.,   56., 2048.,  356.],
               [1624.,  632., 1824.,  876.],
               [ 632.,  420.,  968.,  660.],
               [1200.,  556., 1352.,  832.],
               [ 300., 1044.,  572., 1432.],
               [ 616.,  952.,  832., 1268.],
               [1044., 1076., 1316., 1336.],
               [1104., 1588., 1336., 1776.],
               [ 812., 1956., 1020., 2048.],
               [   0., 1640.,  132., 1832.],
               [1924., 1912., 2048., 2048.],
               [1456., 1988., 1680., 2048.],
               [   0., 1992.,  168., 2048.]], dtype=float32),
        'labels': array([7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7]),
        'masks': [{'size': [2048, 2048],
          'counts': b'hRhS1h1Xn100000@P1@00000H`0H00000H`0H00000H<L00000H<L00000L8L0(
8L00000L4000000H<L0000000000000L400000004L000000000000000000000000000004L0000(
00000L400000000000000000000000000000000000000L400000000000000000000000000000L4
00L4000000L4000000L400000000000000H8000000000000H8000000L4000000L4000000L400(
4D8000008\\O<000008@800000<XO<00000`ng^2'},
         {'size': [2048, 2048],
          'counts': b'TP`o14lo100000L`0D0000008H0000004L0000008H0000004L000000<D0000(
000008H0000004L0000004L0000004L0000004L0000004L0000004L0000004L000000000000004L
004L000000000000000000000000000004L000000000000000000000000000000000000000000(
0000000L400000000000000000000L4000000L400000000000000L4000000L4000000L400000(
0000D<000004\\O`000000loWc1'},
         {'size': [2048, 2048],
          'counts': b'lV`h3X1hn100000@T1\\O00000Dd0H00000H`0H00000H`0H00000L<H00000D`
000H<L00000H<L00000H8000000L8L00000L8L000000000000L4000000H<L00000L4000000L8L(
```

Pour chaque cellule, nous :

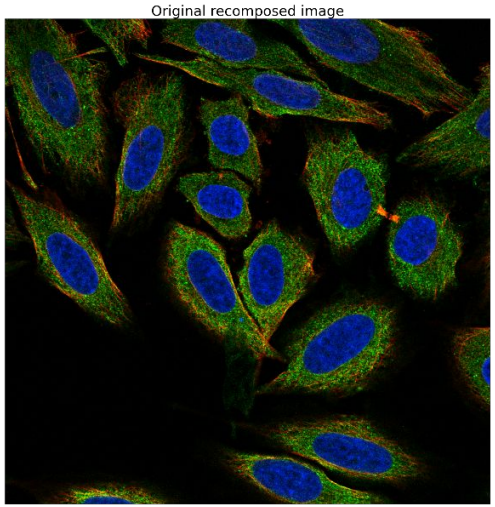
- créons un masque binaire (noir/blanc)
- codons le masque avec le codage RLE
- créons des BBoxes basées sur le RLE
- créons le format Json COCO labellisé.
- procédons au Feedforward du modèle masK-RCNN

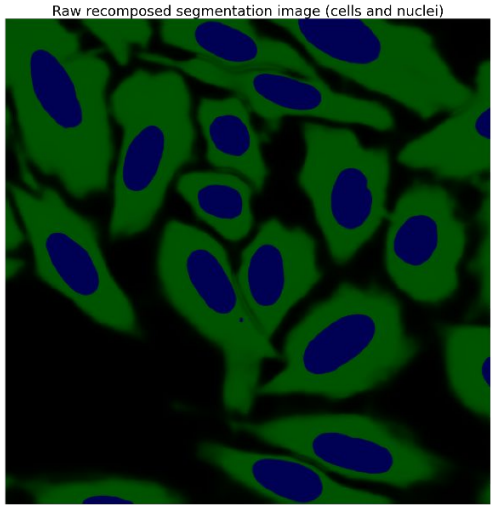
kaggle™

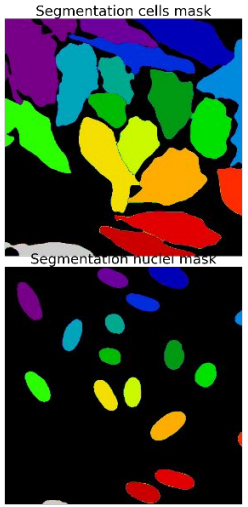
SOLUTION: SECONDE APPROCHE

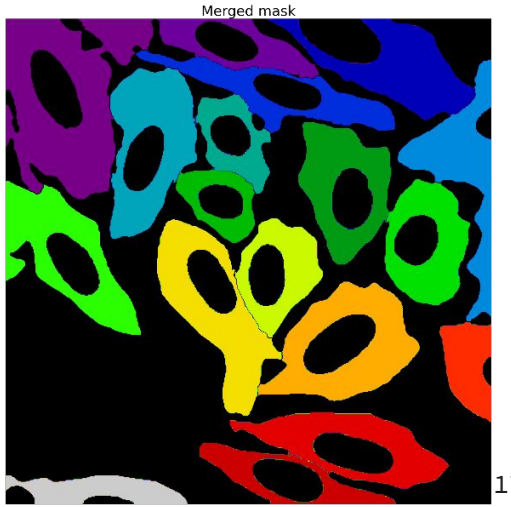
1 **Prédire les masques cellulaires en utilisant HPA CELLSEG**

- Prédire le masque des noyaux
- Prédire le masque de la cellule entière
- Soustraire les noyaux des cellules









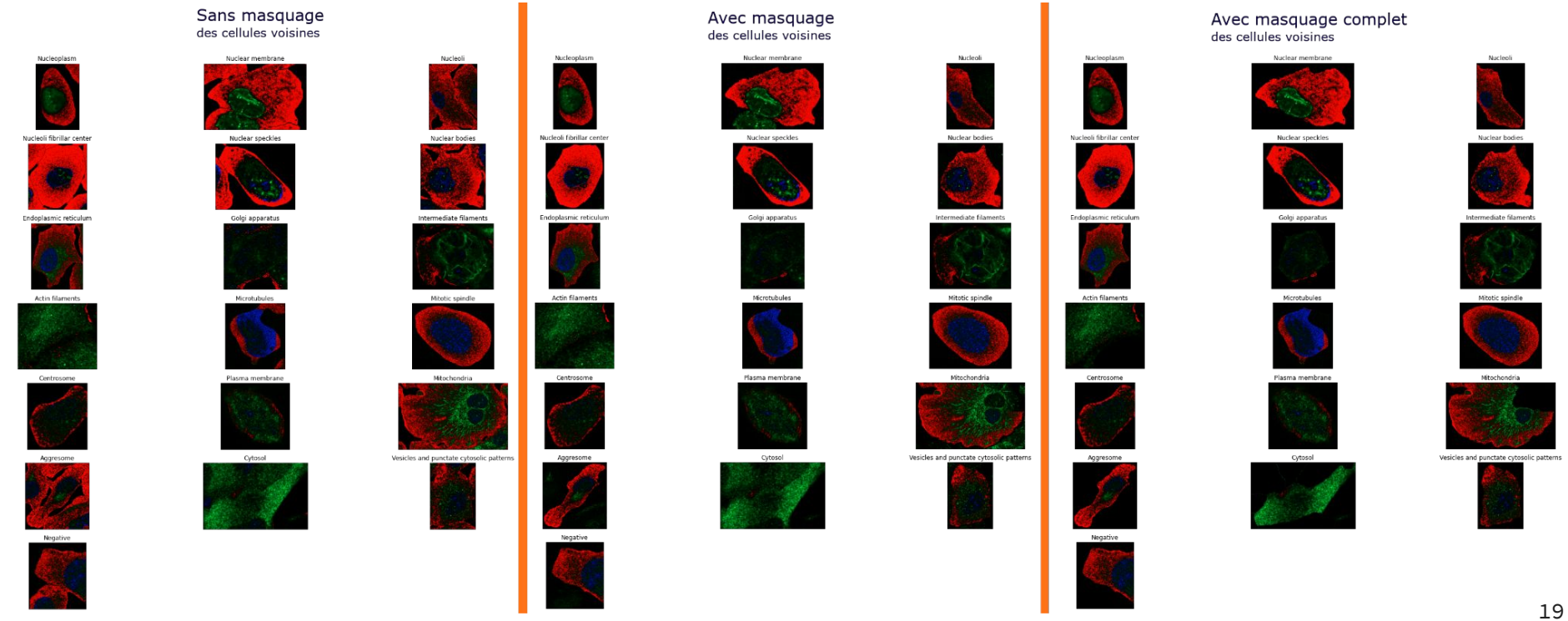
SOLUTION : SECONDE APPROCHE

2 **Extraire chaque cellule comme image unique ou “crop”**

- Calculer les bounding boxes à partir des masques
- Cropper puis binariser les masques
- Nettoyer la couche verte croppée et l’ajouter aux masques
- Redimensionner

SOLUTION : SECONDE APPROCHE

Choix du type de masquage : intuitif



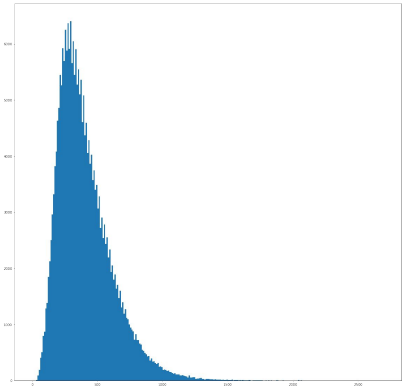
SOLUTION : SECONDE APPROCHE

2 **Extraire chaque cellule comme image unique ou “crop”**

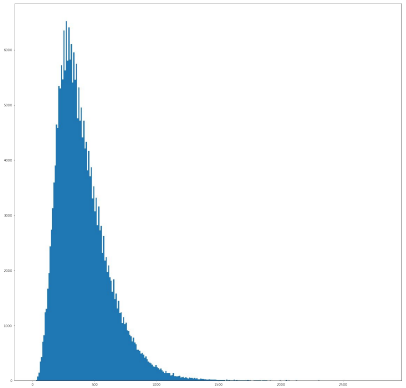
- Calculer les bounding boxes à partir des masques
- Cropper et binariser les masques
- Nettoyer la couche verte et l’ajouter aux masques
- Redimensionner

	Height	Width
count	248271.000000	248271.000000
mean	405.671178	410.887583
std	208.133892	210.459873
min	31.000000	29.000000
25%	256.000000	260.000000
50%	363.000000	367.000000
75%	512.000000	518.000000
max	2709.000000	2839.000000

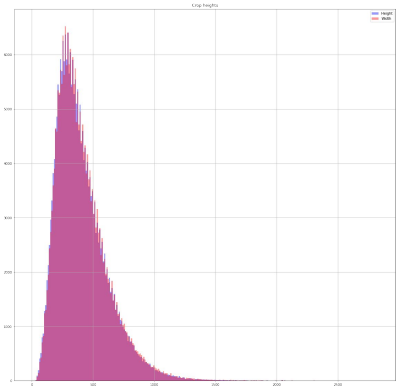
Hauteur



Largeur



Comparaison



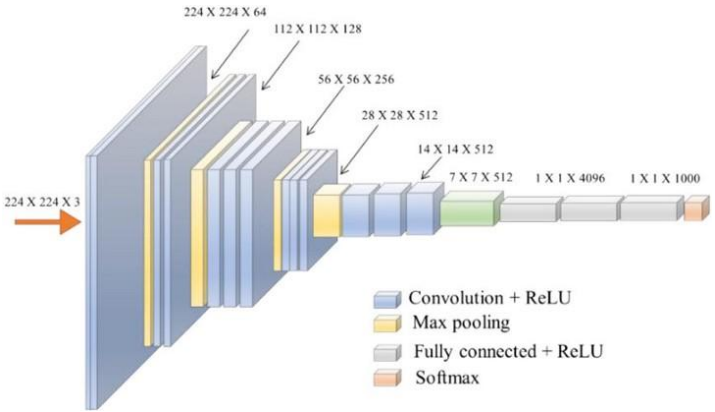
But

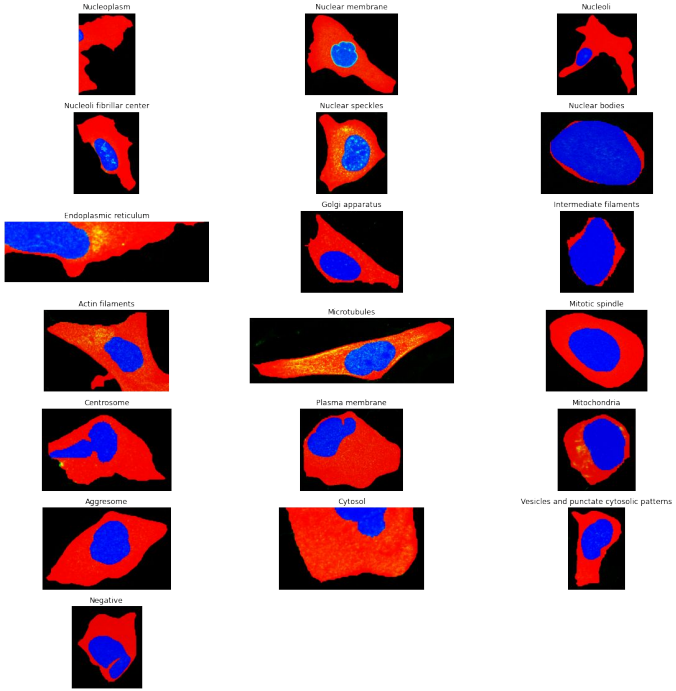
Uniformiser en perdant le minimum d’informations

SOLUTION: SECONDE APPROCHE

3 **Feeder les crops à un réseau de convolution “classique”**

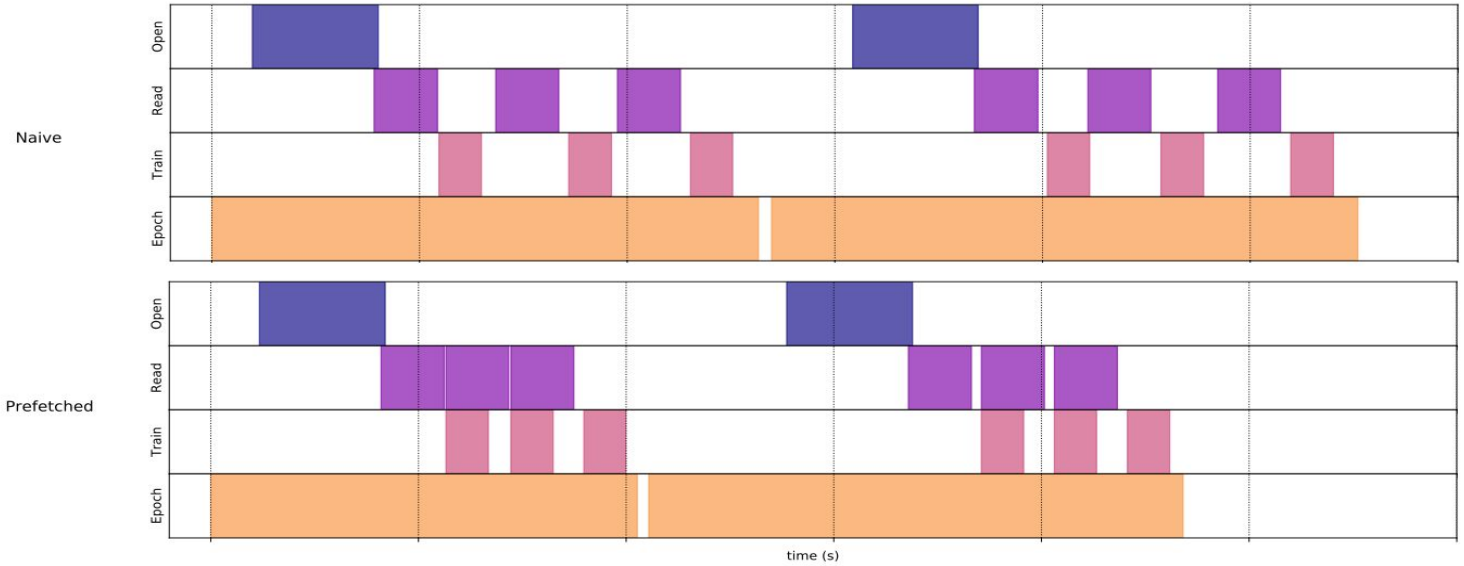
- Expérimenter et inclure des intuitions
- Utiliser les données de test pour évaluer le modèle





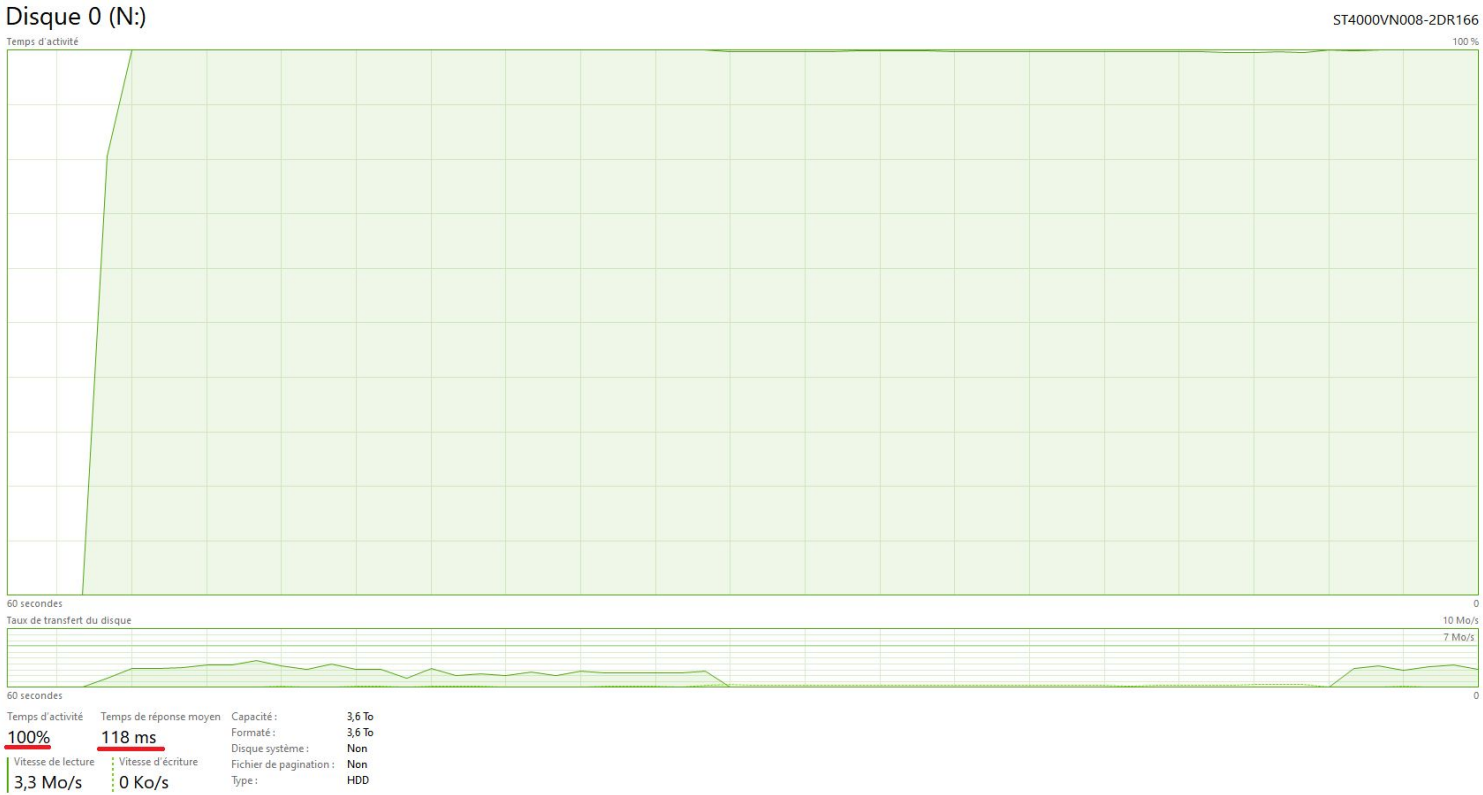
CONSTRUCTION DU MODÈLE - PRÉPARATION DES DONNÉES

- TensorFlow
- `tf.Dataset : batch, prefetch et cache`



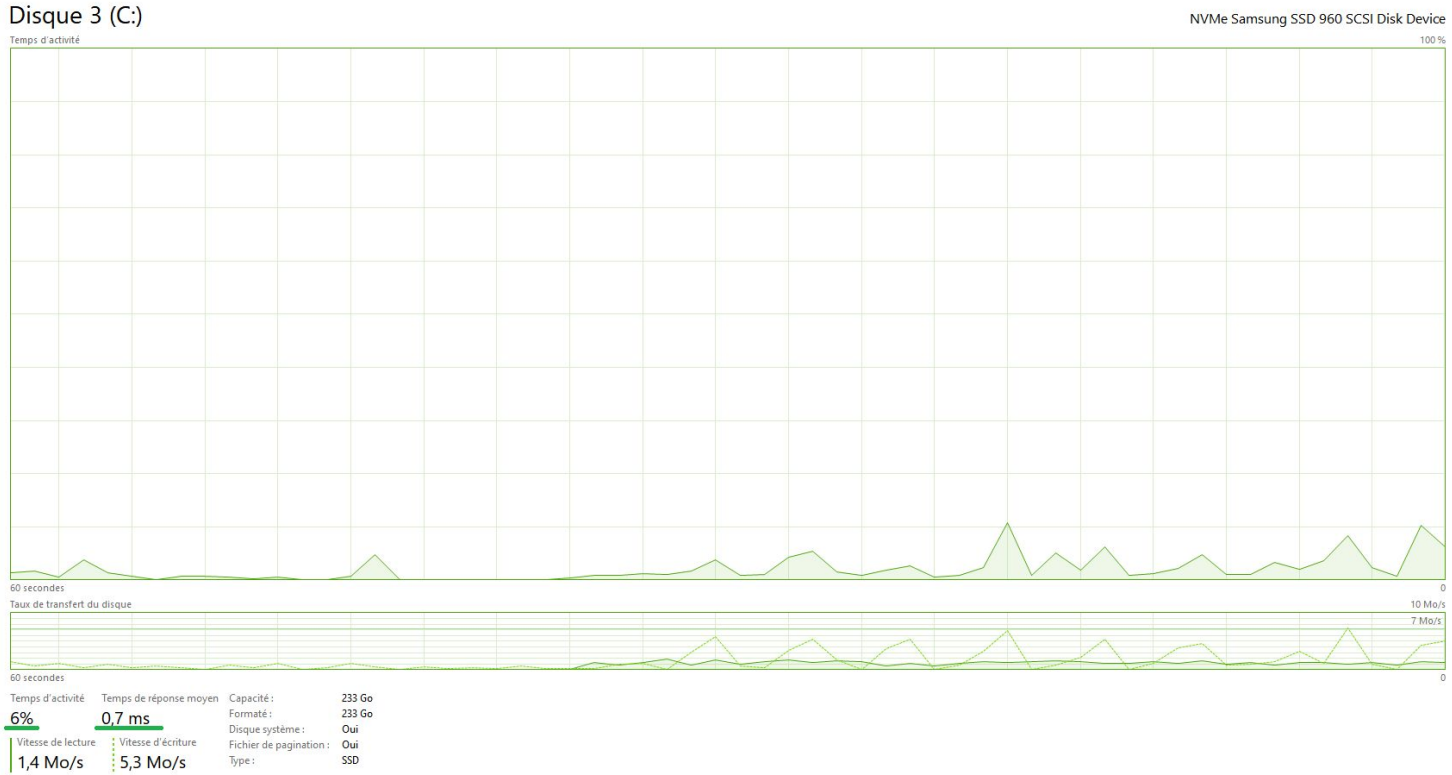
CONSTRUCTION DU MODÈLE - PRÉPARATION DES DONNÉES

60 minutes par EPOCH au mieux



CONSTRUCTION DU MODÈLE - PRÉPARATION DES DONNÉES

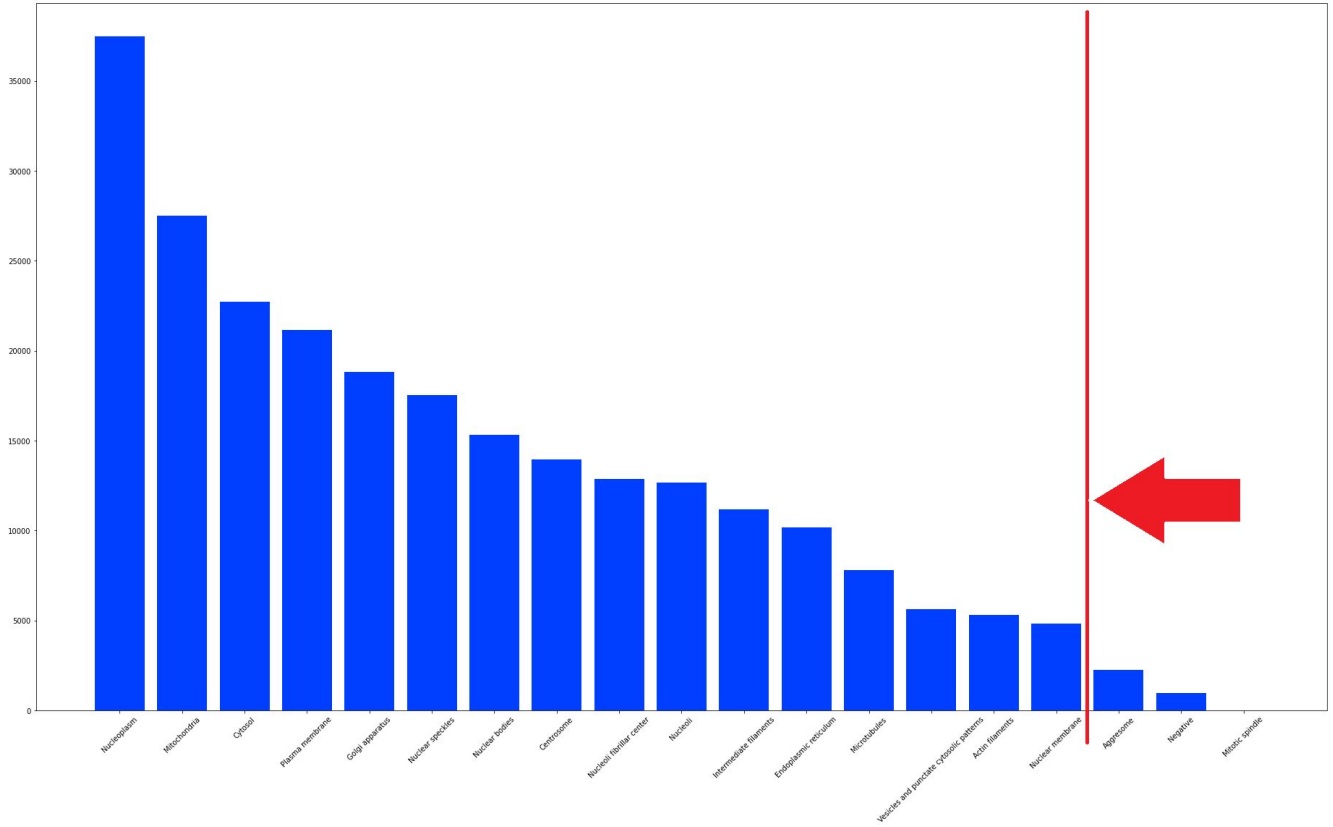
3 minutes par EPOCH au mieux



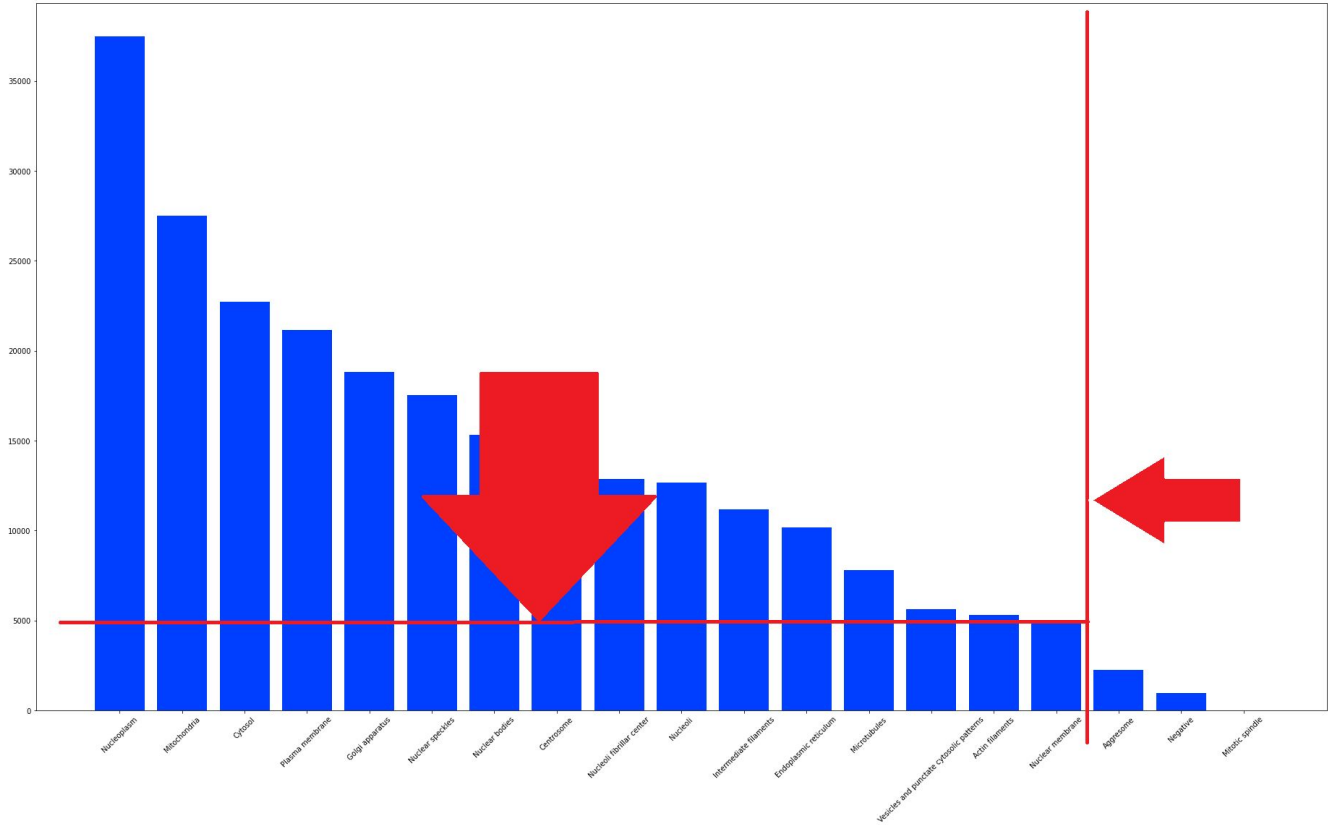
CONSTRUCTION DU MODÈLE - NETTOYAGE DES DONNÉES

- Défaut des tf.Dataset :
  - Pas de sample_weight
  - Pas de class_weight en multiclasse
- Solution : uniformiser la distribution des classes par crop

CONSTRUCTION DU MODÈLE - NETTOYAGE DES DONNÉES

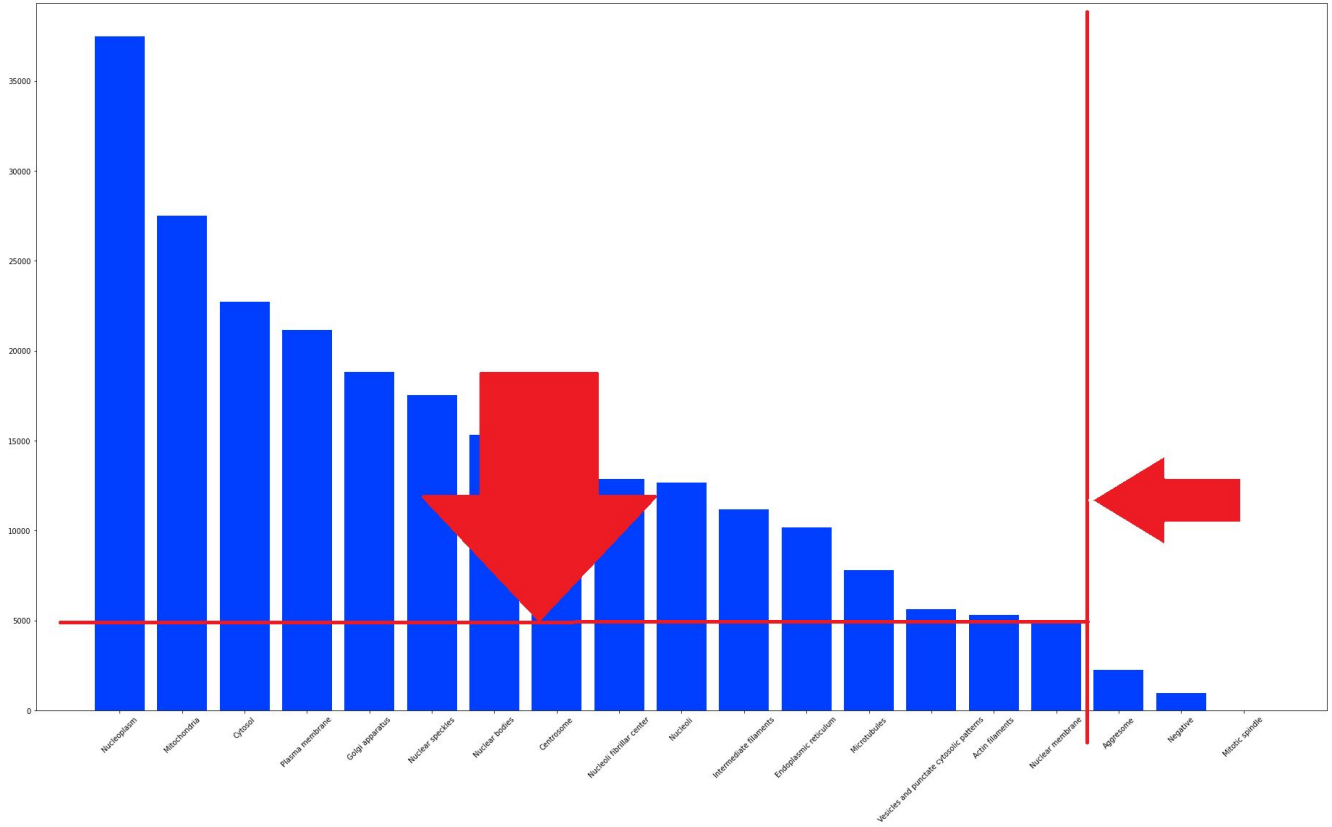


CONSTRUCTION DU MODÈLE - NETTOYAGE DES DONNÉES



CONSTRUCTION DU MODÈLE - NETTOYAGE DES DONNÉES

245’035
->
77’520
crops



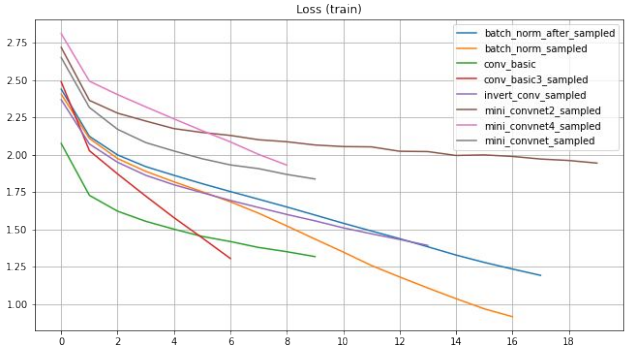
CONSTRUCTION DU MODÈLE - ARCHITECTURE

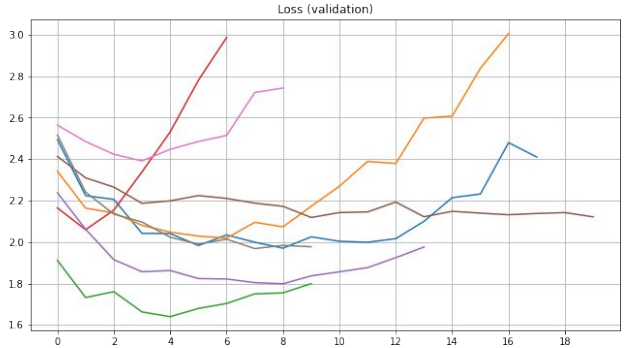
- Intuition 1 :
  Réseau classique

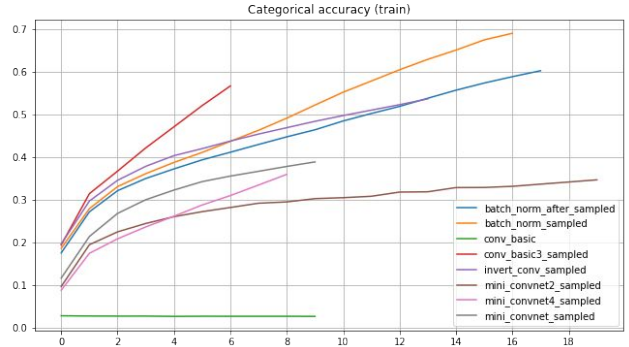
  Kernel de taille ~ fixe, convolution et max pooling
  -> Overfit
  “conv_basic3”

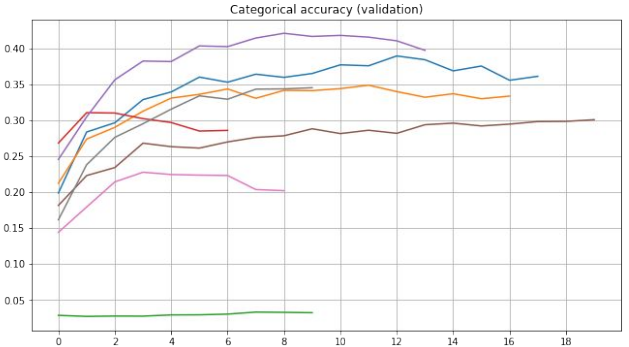
- Intuition 2 :
  Réseau de type convnet

  Pairs de convolutions consécutives
  -> Modèle rigide
  “conv_basic”









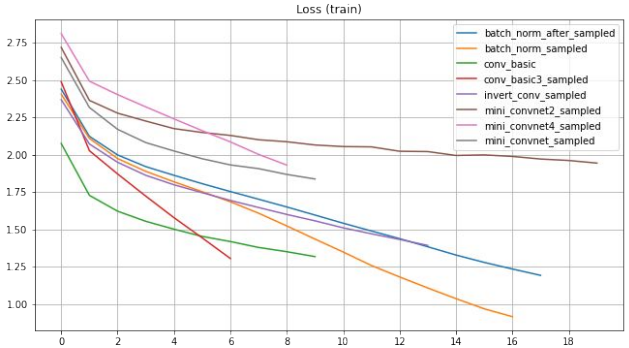
CONSTRUCTION DU MODÈLE - ARCHITECTURE

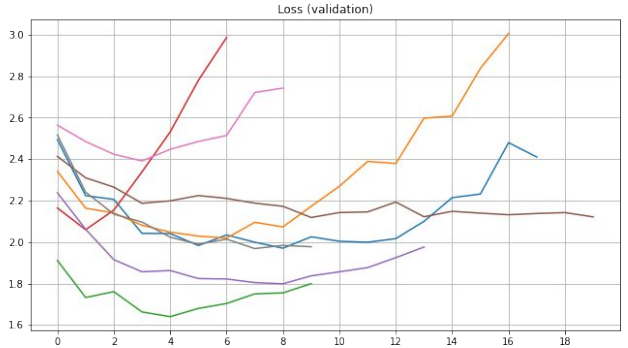
- Intuition 3 : Kernels de convolution larges

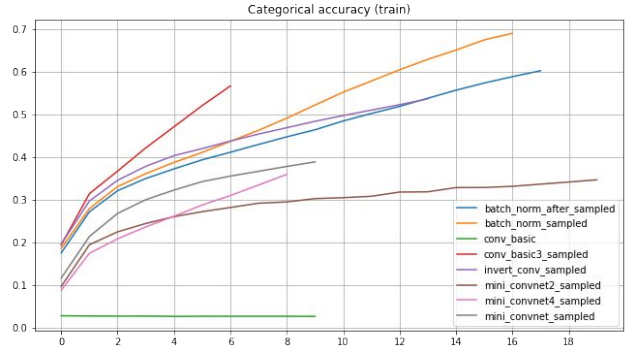
  -> Bons résultats “mini_convnet2”

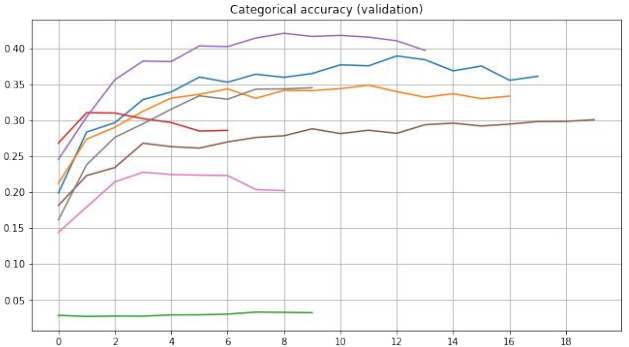
- Intuition 4 : Plus de neurones Dense

  -> Overfit “mini_convnet4”









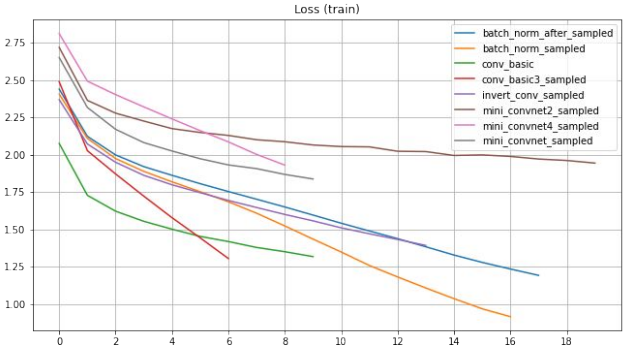
CONSTRUCTION DU MODÈLE - ARCHITECTURE

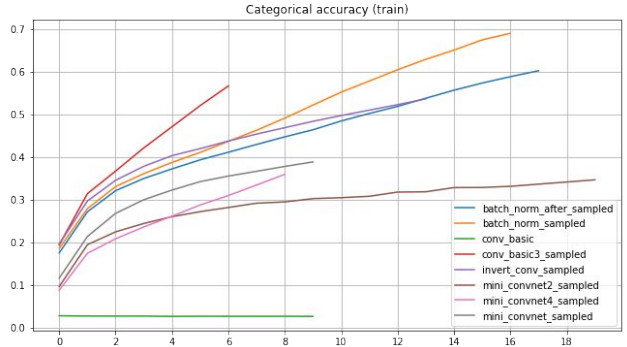
- Intuition 5 : Batch normalization

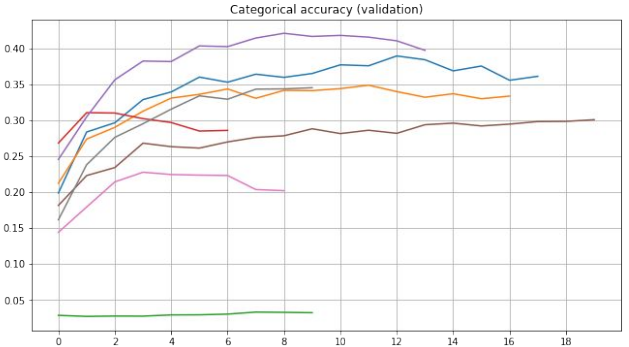
  -> Très bons résultats “batch_norm”

- Intuition 5 : BN après ReLU

  -> Très bons résultats “batch_norm_after”



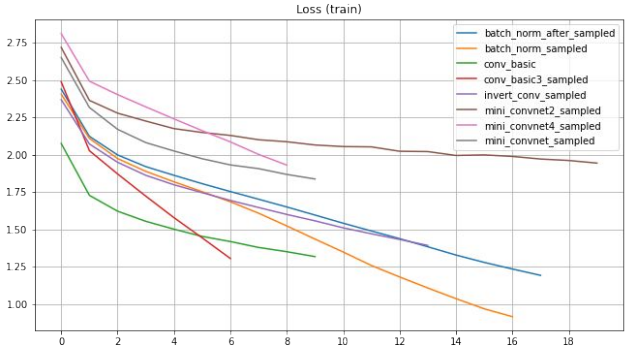


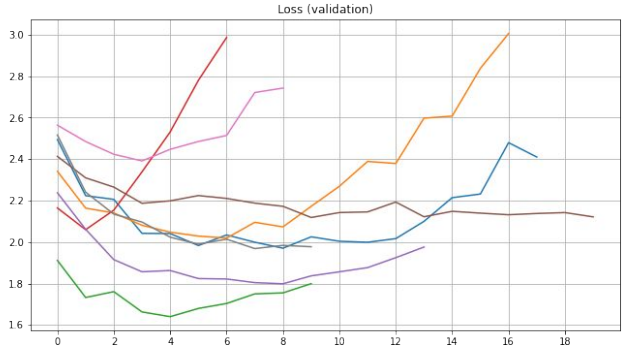


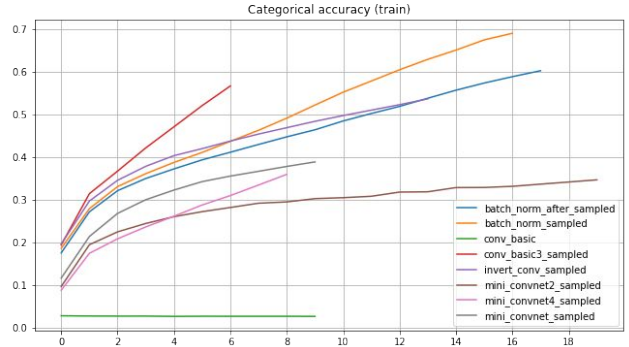
CONSTRUCTION DU MODÈLE - ARCHITECTURE

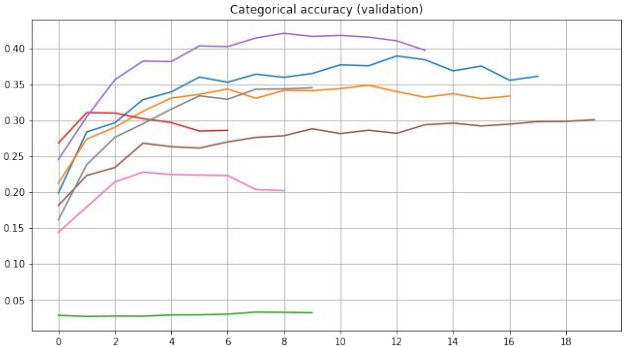
- Intuition 6: Inverser l'évolution de la taille des convolutions

  -> Meilleurs résultats "invert_conv"









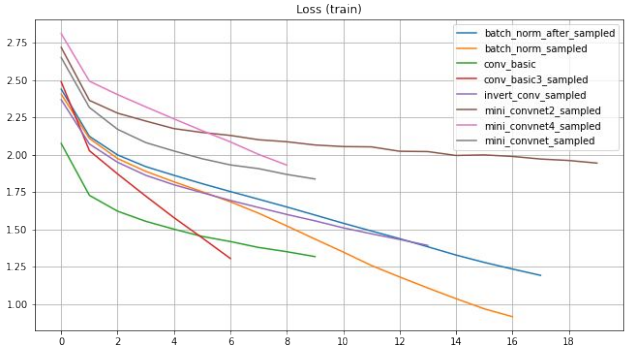
CONSTRUCTION DU MODÈLE - ARCHITECTURE

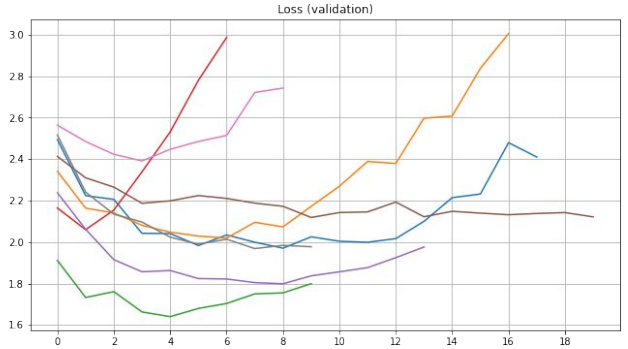
- Intuition 6: Inverser l'évolution de la taille des convolutions

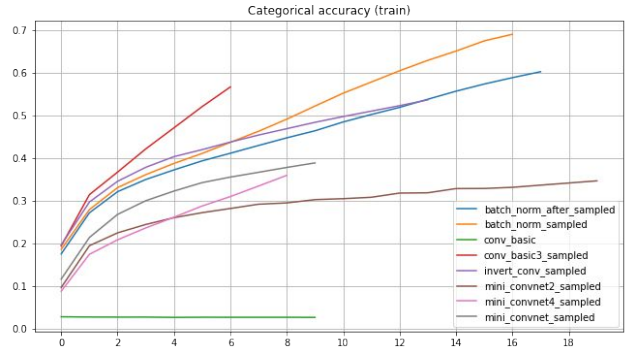
  -> Meilleurs résultats "invert_conv"

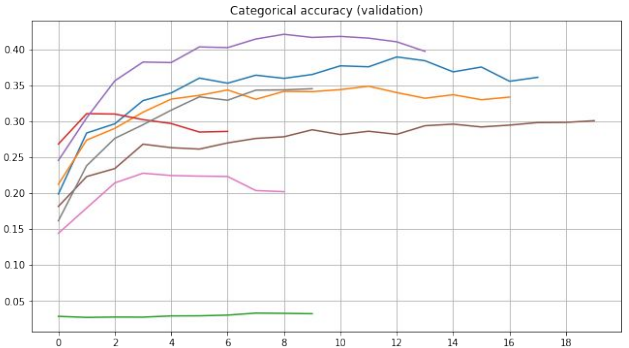
- Intuition 7: Ajout de Dropout

  -> en cours Excellents résultats









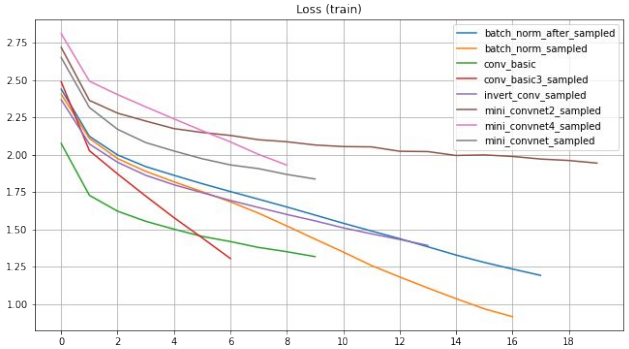
CONSTRUCTION DU MODÈLE - ARCHITECTURE

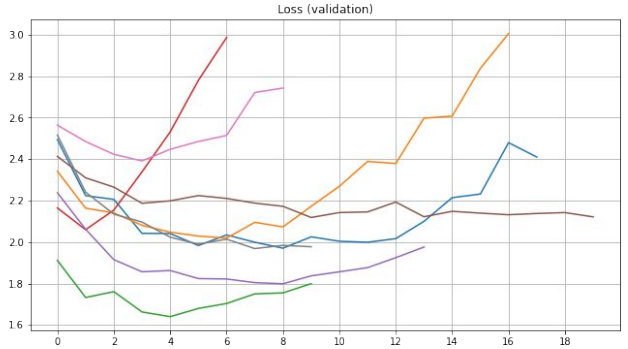
- Intuition 6: Inverser l'évolution de la taille des convolutions

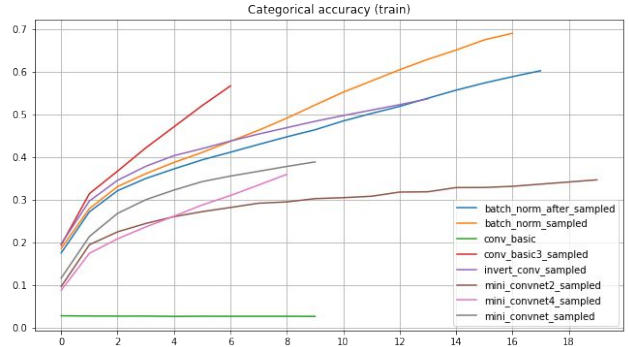
  -> Meilleurs résultats "invert_conv"

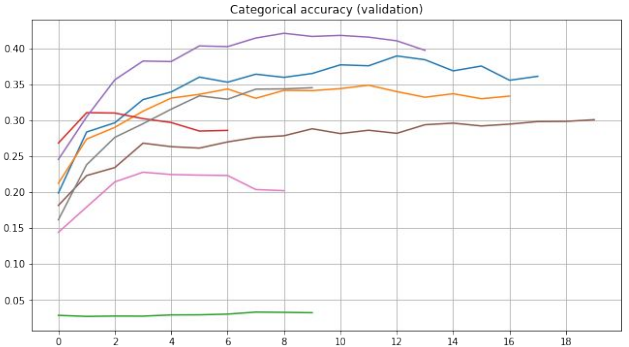
- Intuition 7: Ajout de Dropout

  -> en cours Excellents résultats

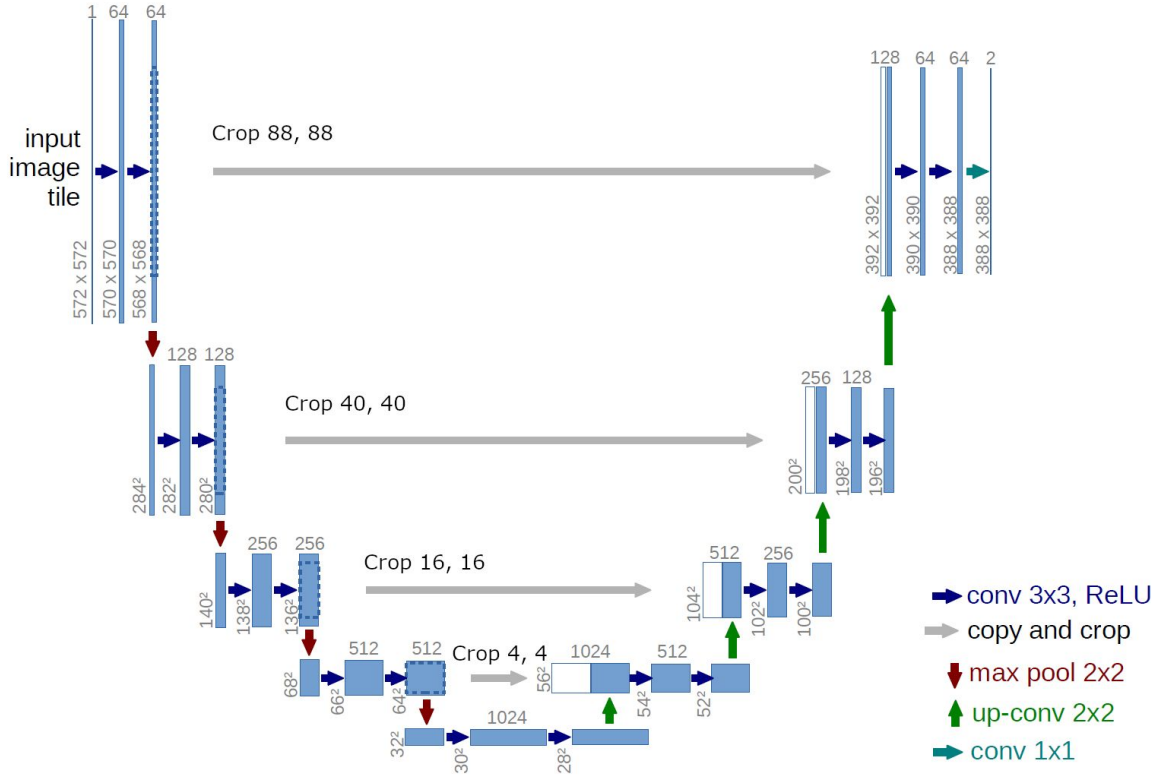








CONSTRUCTION DU MODÈLE - AUTRE ESSAI...



output softmax

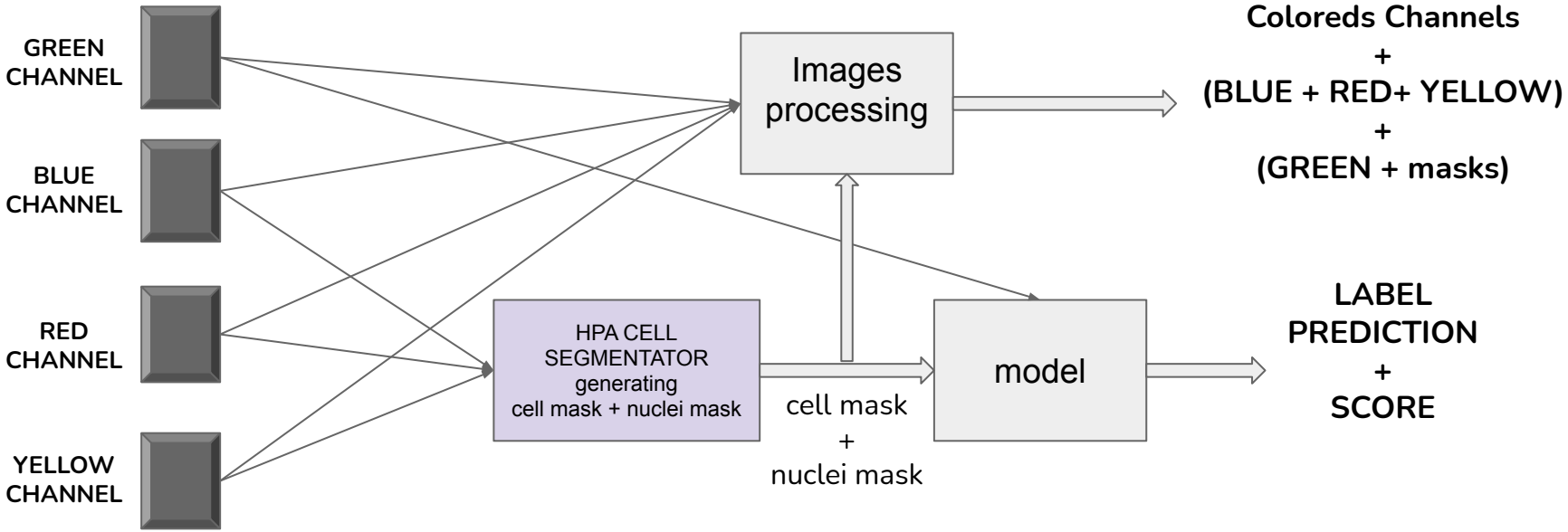
CONSTRUCTION DU MODÈLE - ARCHITECTURE

- En cours...

```
Epoch 1/20
1885/1885 [==============================] - 1003s 525ms/step - loss: 3.1015 - categorical_accuracy: 0.0750 -
val_loss: 2.6791 - val_categorical_accuracy: 0.1147
Epoch 2/20
1885/1885 [==============================] - 1011s 536ms/step - loss: 2.4796 - categorical_accuracy: 0.1542 -
val_loss: 2.4587 - val_categorical_accuracy: 0.1638
Epoch 3/20
1885/1885 [==============================] - 1014s 538ms/step - loss: 2.3348 - categorical_accuracy: 0.2076 -
val_loss: 2.2597 - val_categorical_accuracy: 0.2400
Epoch 4/20
1885/1885 [==============================] - 1045s 554ms/step - loss: 2.2327 - categorical_accuracy: 0.2437 -
val_loss: 2.2246 - val_categorical_accuracy: 0.2478
Epoch 5/20
1885/1885 [==============================] - 1062s 563ms/step - loss: 2.1736 - categorical_accuracy: 0.2605 -
val_loss: 2.3913 - val_categorical_accuracy: 0.2021
Epoch 6/20
1885/1885 [==============================] - 1122s 595ms/step - loss: 2.1364 - categorical_accuracy: 0.2814 -
val_loss: 2.2063 - val_categorical_accuracy: 0.2546
Epoch 7/20
1885/1885 [==============================] - 1161s 616ms/step - loss: 2.0882 - categorical_accuracy: 0.2950 -
val_loss: 2.1813 - val_categorical_accuracy: 0.2747
Epoch 8/20
1885/1885 [==============================] - 1158s 614ms/step - loss: 2.0622 - categorical_accuracy: 0.3024 -
val_loss: 2.1891 - val_categorical_accuracy: 0.2825
Epoch 9/20
 814/1885 [===========>..................] - ETA: 11:07 - loss: 2.0374 - categorical_accuracy: 0.3234
```

API

API : Architecture

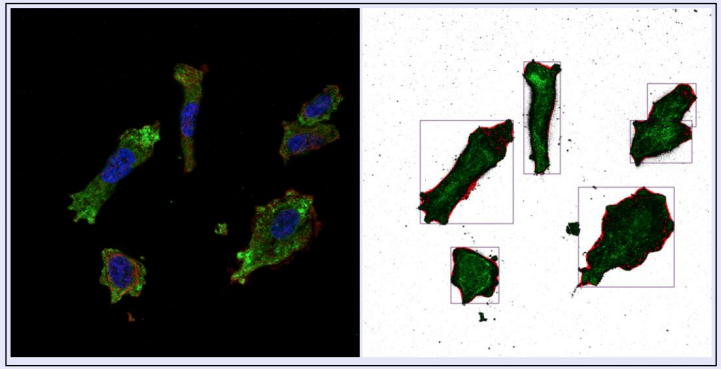


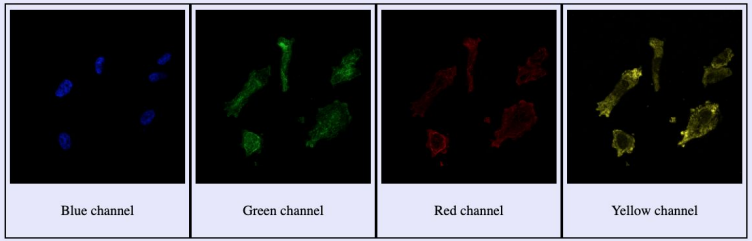
API : INTERFACE

Hum **ion**

Veuillez sélectionner toute ge rouge, image jaune.

Human Protein Atlas - Single Cell Classification





Prédiction : Nucleoplasm
Confiance : 12.13 %