Projet 'Analyse de données'

Formation Data Scientist
ECAM 2018-2019

Antoine LEBLANC
Chérif DIALLO
Yaroslav KONYSHEV

28.01.2019

Objectifs

- **Pour qui ?**

  Gas and Hydrocarbon

- **Pour quoi ?**

  Identifier des présences de corps salés sur des images sismiques

- **Faire quoi ?**

  Créer un modèle algorithmique efficace et performant.

-

Plan

I - Introduction Générale et présentation du projet

1. Gas and Hydrocarbon (G&H) et les hydrocarbures
2. Données et premières appréhensions
3. Méthode et méthodologie

II - Le réseau de neurones à convolution

1. Schéma du Réseau de neurone à convolution
2. Définition des termes et paramètres
   - Convolution
   - Padding et Strides
   - ReLu
   - MaxPooling
   - Dropout
3. Exemple

Plan

III - Training

1. Train/validation split
2. Image preprocessing
3. Image augmentation
4. Metric
5. Model params
6. Model fit

IV - Conclusion

1. Pistes d’amélioration
2. Difficultés rencontrées

I - Introduction et présentation du projet

1 - Gas and Hydrocarbon (G&H) et les hydrocarbures

2 - Données et premières appréhensions

3 - Méthode et méthodologie

1 - Gas and Hydrocarbon (G&H) et les hydrocarbures

- G&H, qu’est ce ?

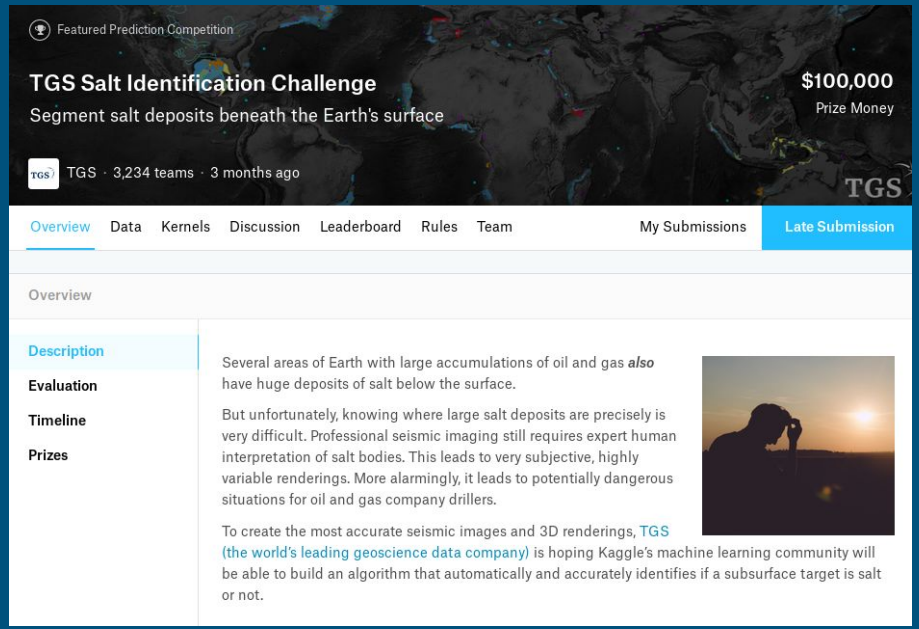
  Multinationale pétrolière

- Hydrocarbures et sel oui, mais quel rapport ?

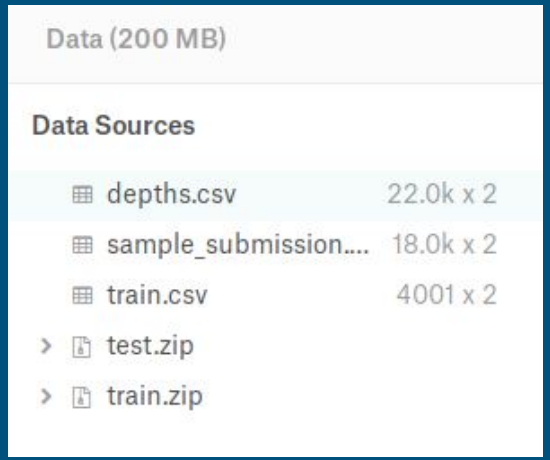
  Roches salines ⟶ piège à hydrocarbures

2 -Données et premières appréhensions

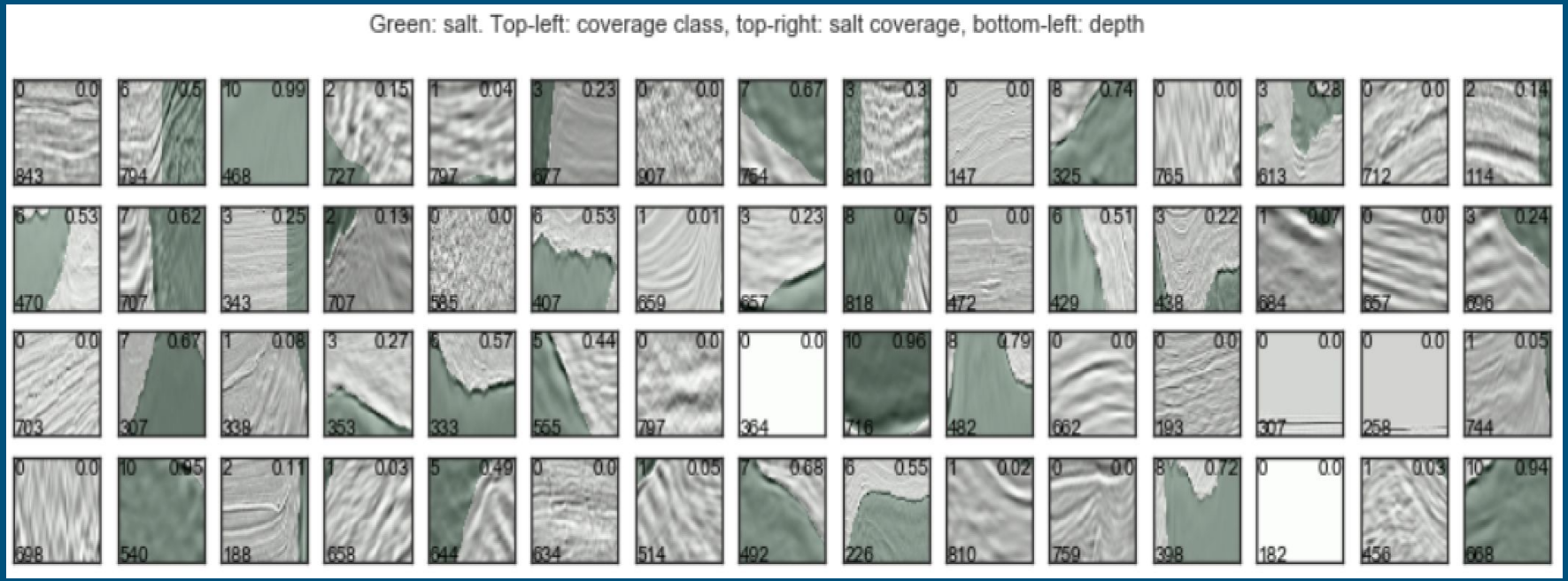
https://www.kaggle.com/c/tgs-salt-identification-challenge



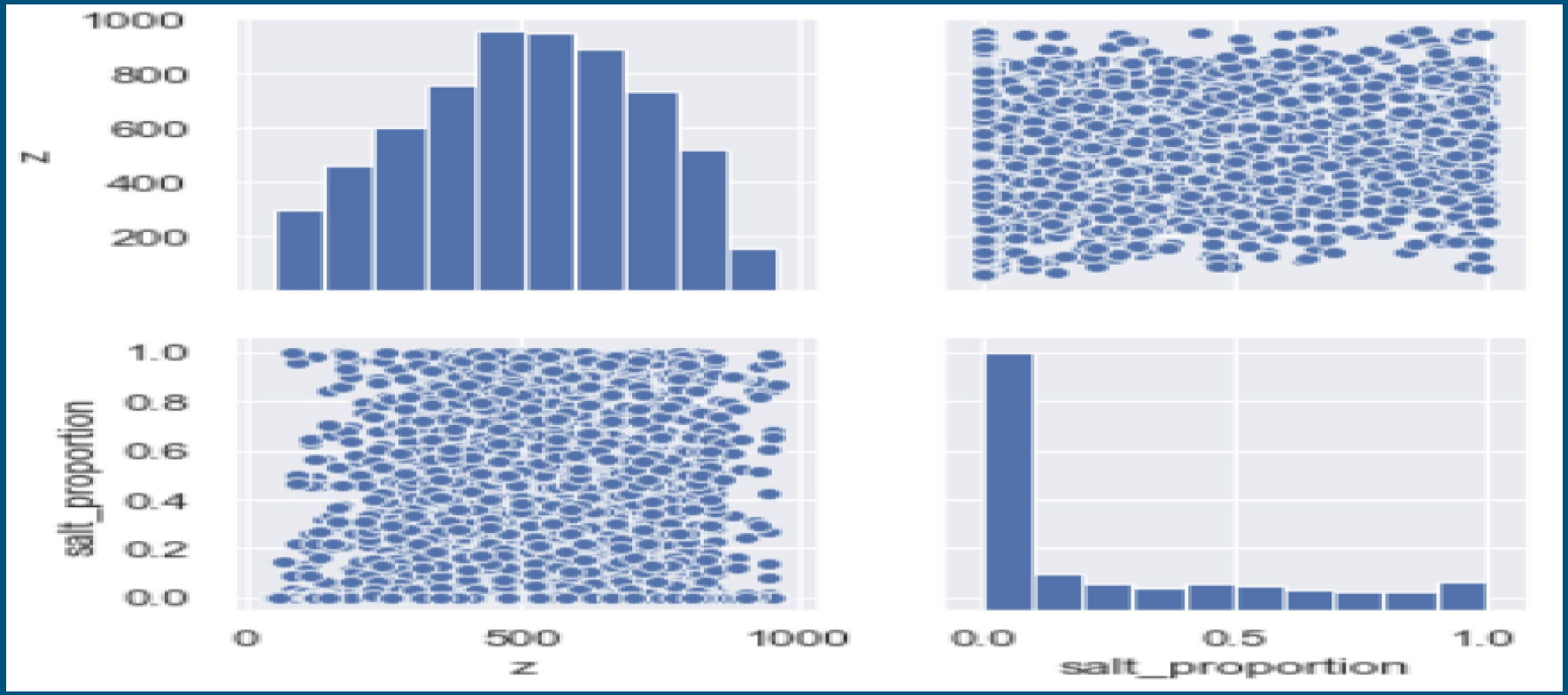
4000 images 101x101 pixels



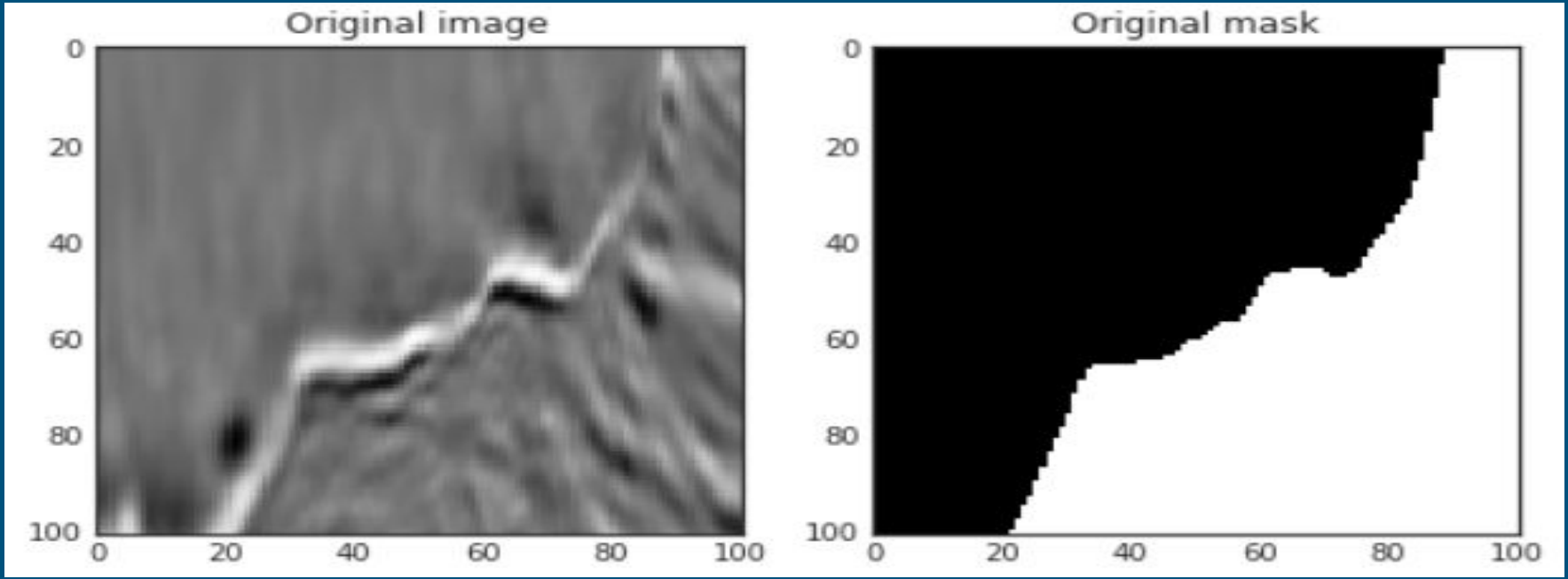
2 -Données et premières appréhensions



2 -Données et premières appréhensions

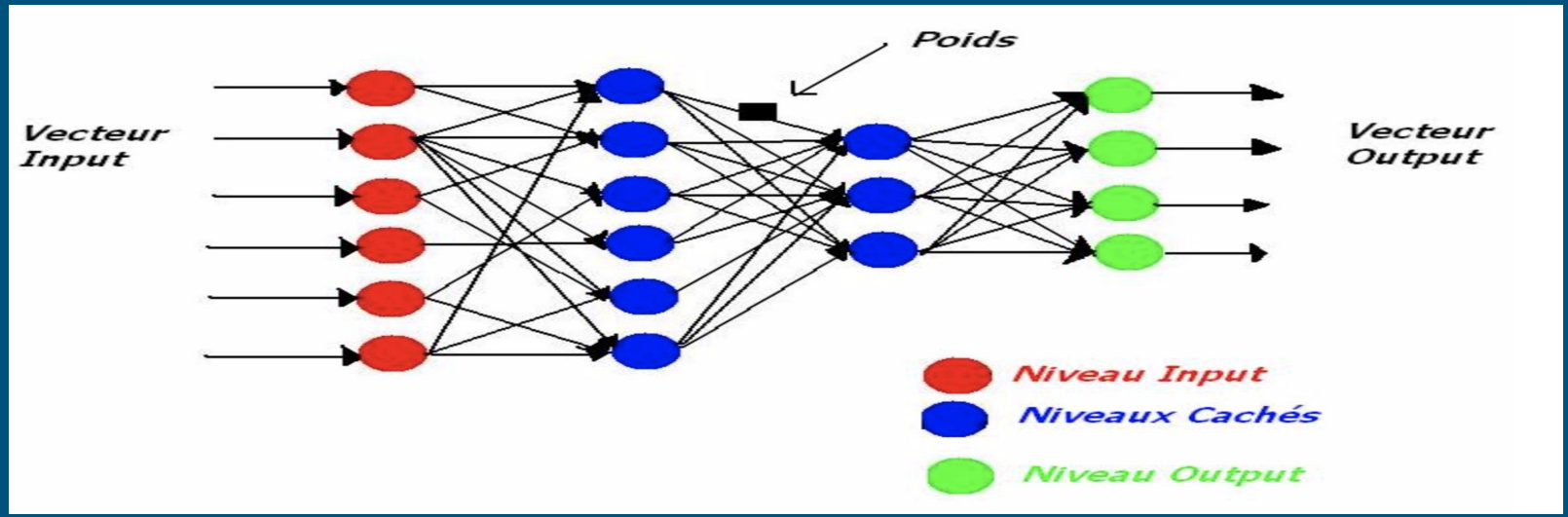


2 - Données et premières appréhensions



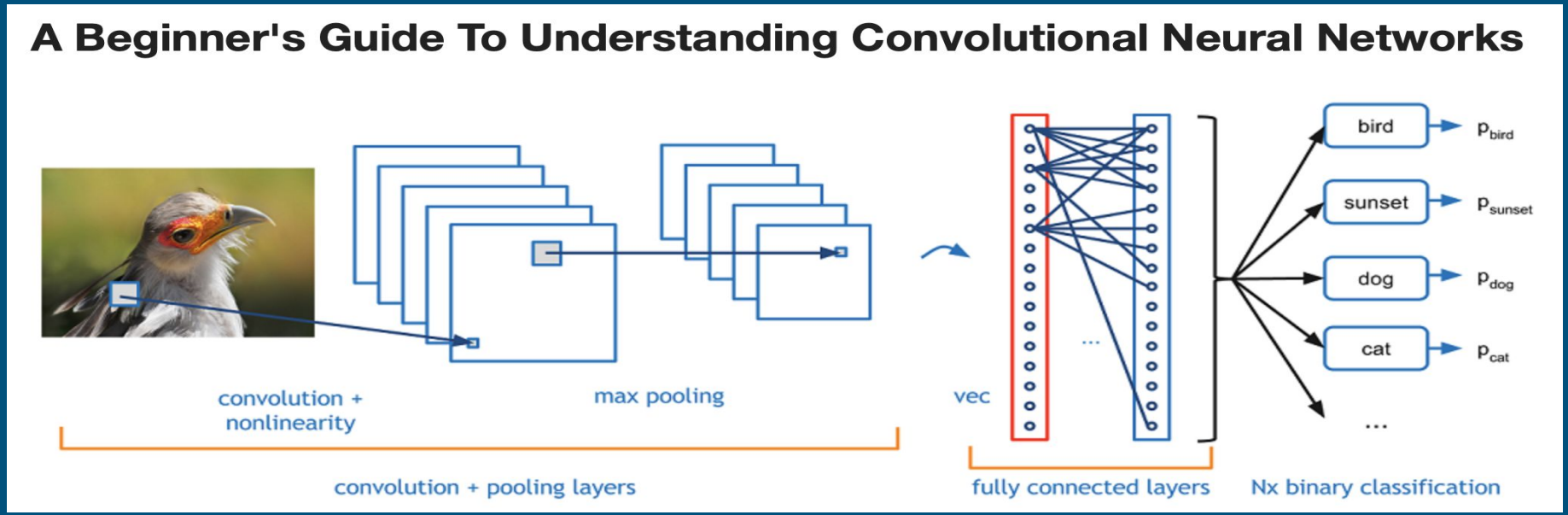
3 - Méthode et méthodologie

1- Réseaux de neurones (artificiels)



3 - Méthode et méthodologie

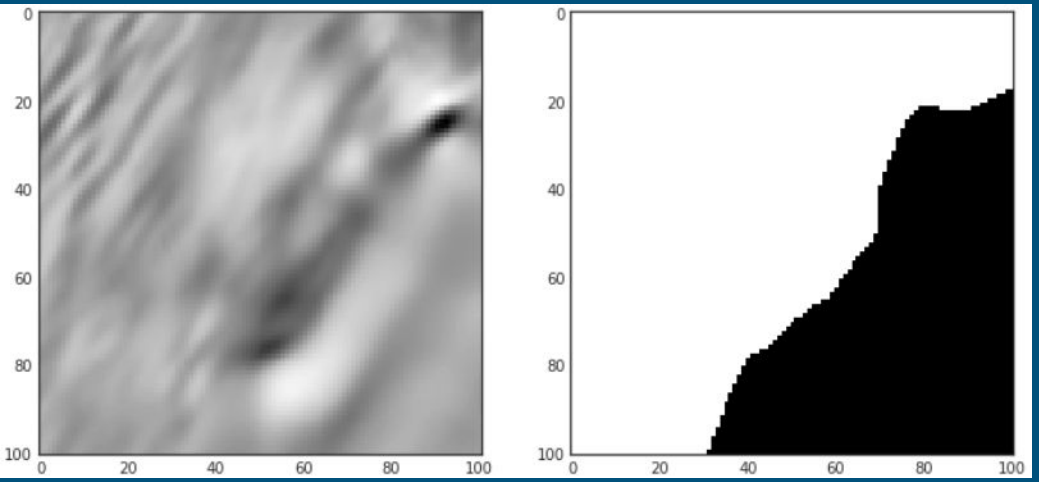
2- Réseaux de neurones Convolutionnel (CNN)



Convolutional NN Avec U-Net

Reconnaître des formes, des pattern.

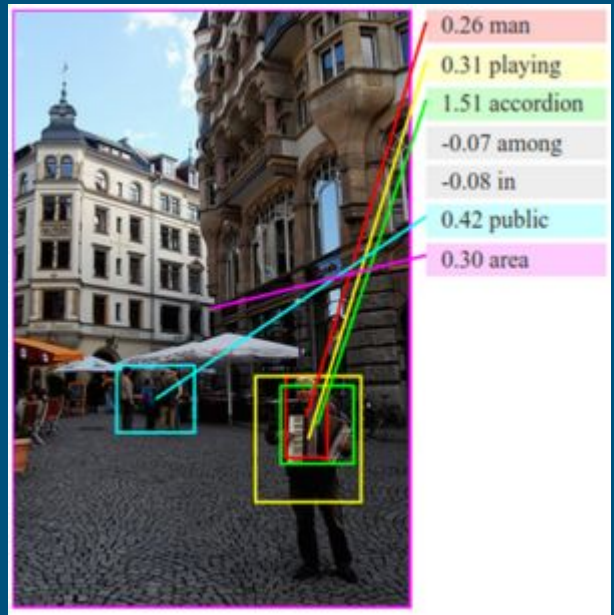
Not fully connected layer



Classical NN

Discrimination, reconnaissance

Fully connected layer

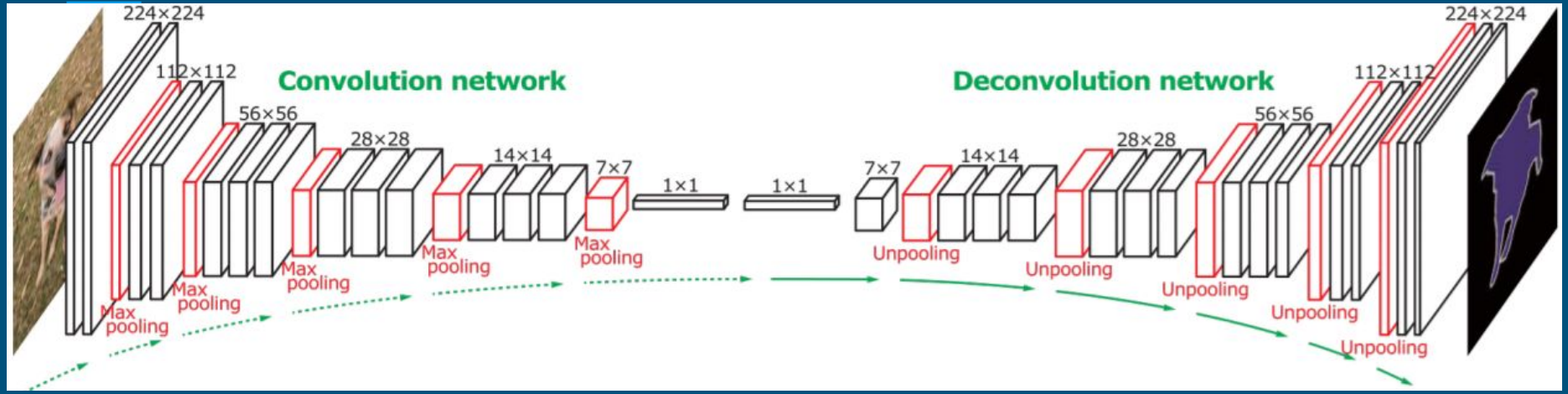


II - Le réseau de neurones à convolution et architecture U-Net

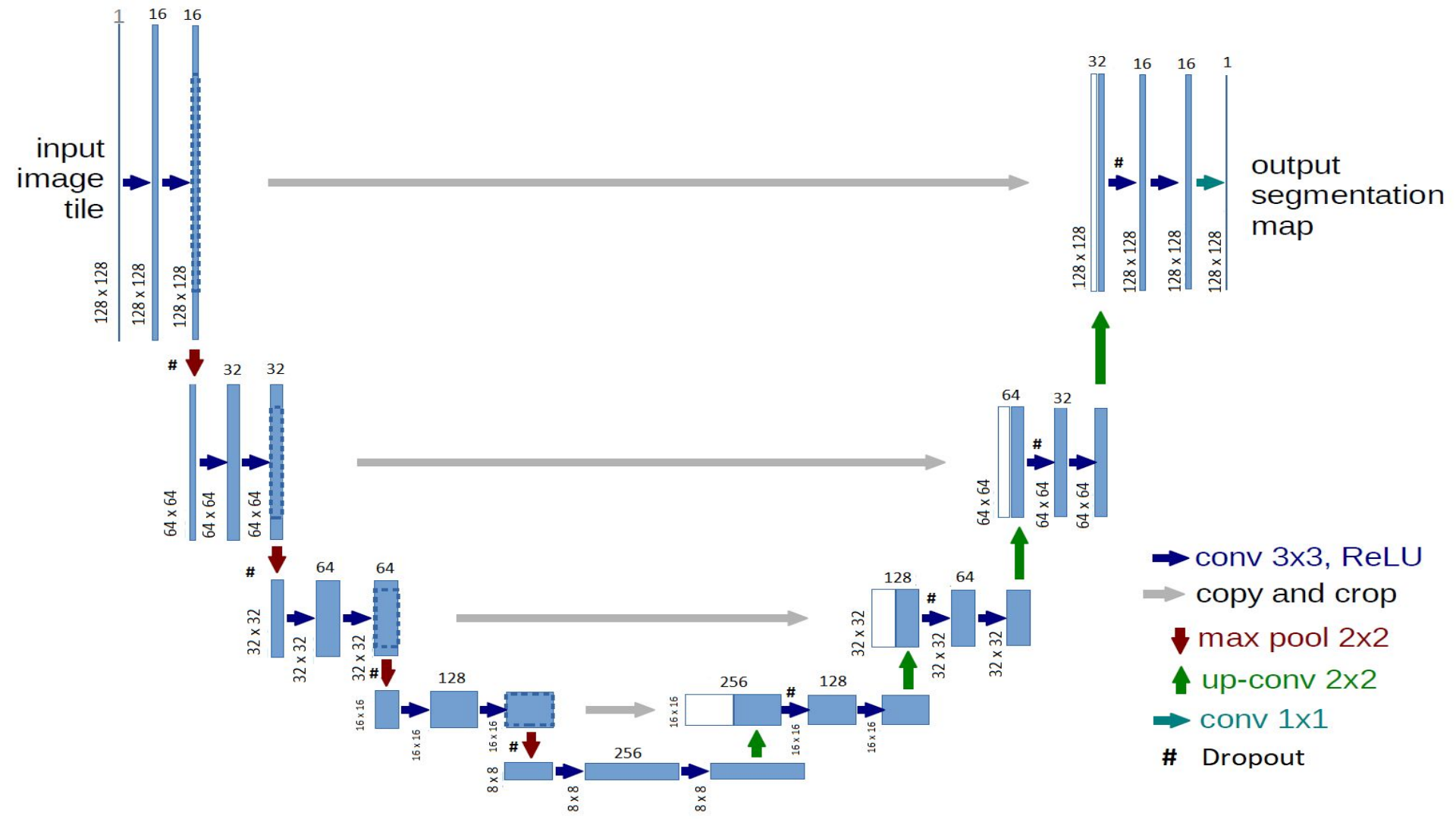
A - Schéma du Réseau de neurone à convolution avec U-Net
B - Définition des termes et paramètres
- Convolution
- Padding et Striding
- ReLu
- MaxPooling
- Dropout

C - Exemple

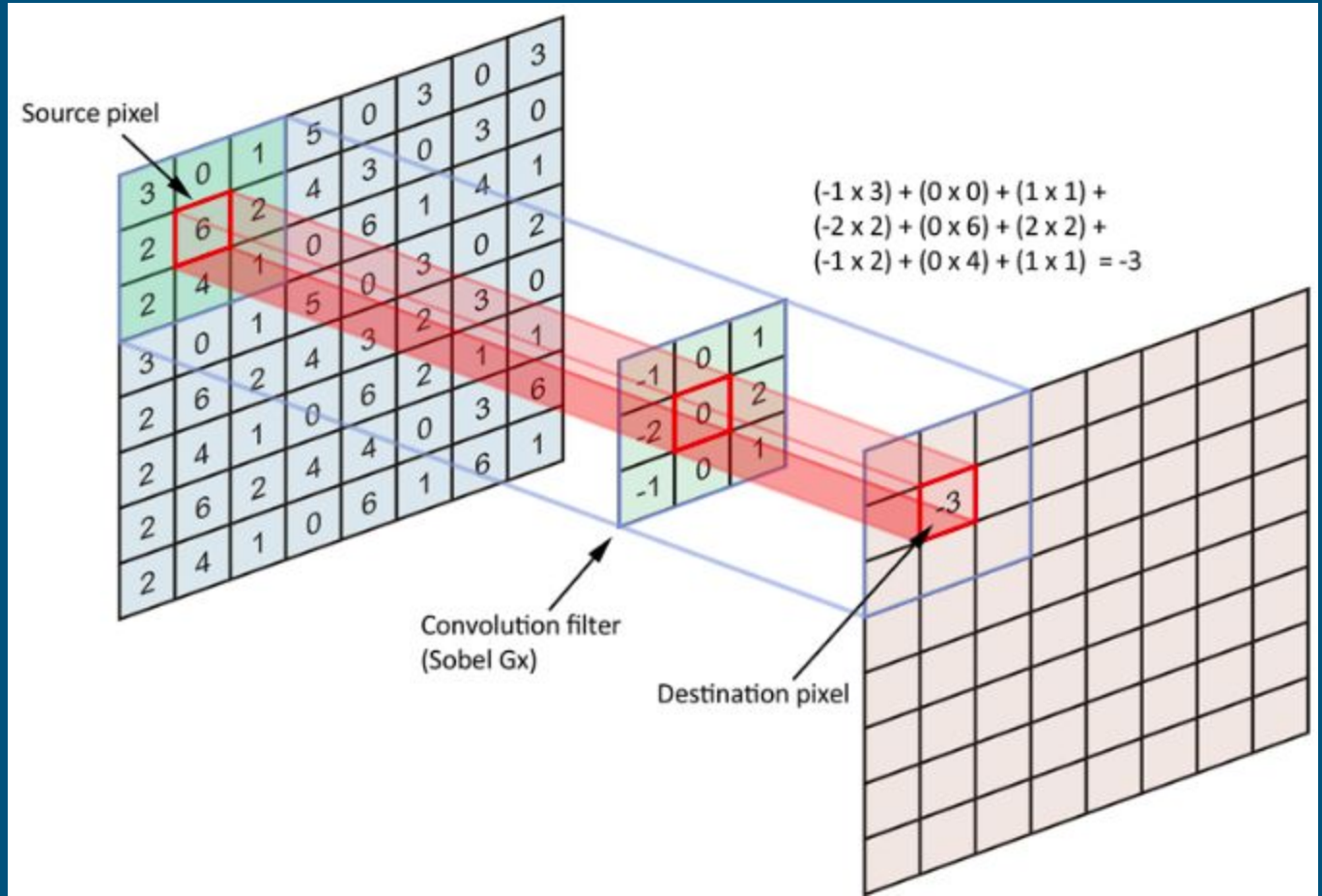
A - Architecture U-net



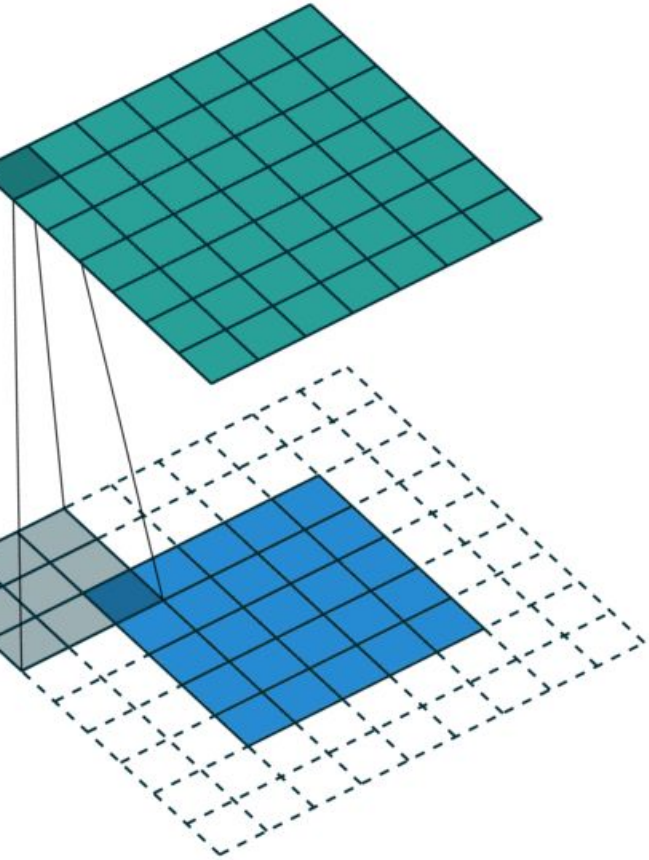
- **Convolutions + ReLu -- Downsampling (maxpooling) + dropout**
- **Déconvolutions + Relu -- Upsampling**

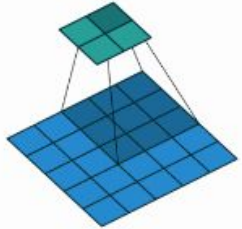

input image tile
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1
Dropout
1
16
16
128 x 128
128 x 128
128 x 128
32
32
64 x 64
64 x 64
64 x 64
64
64
32 x 32
32 x 32
128
16 x 16
16 x 16
256
8 x 8
8 x 8
256
8 x 8
128
16 x 16
16 x 16
128
32 x 32
32 x 32
64
32 x 32
64
64 x 64
64 x 64
32
64 x 64
32
128 x 128
128 x 128
16
128 x 128
16
128 x 128
32
128 x 128
1
128 x 128

B - Définition des termes - Convolution

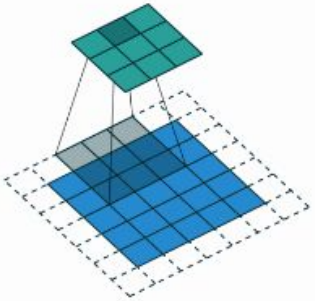


Padding et Strides

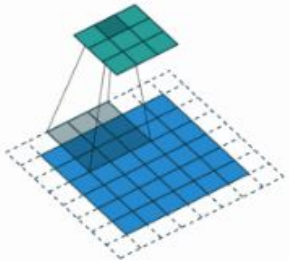




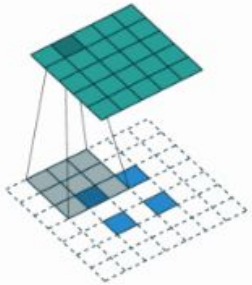
No padding, strides



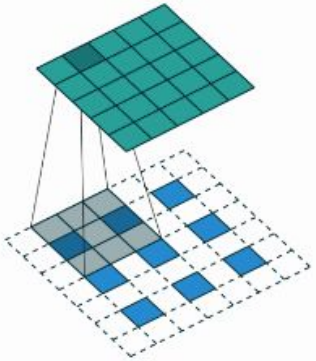
Padding, strides



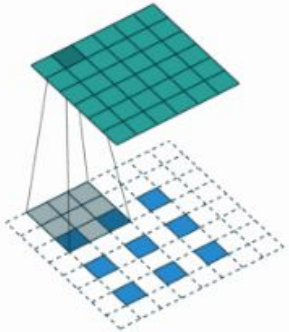
Padding, strides (odd)



No padding, strides, transposed

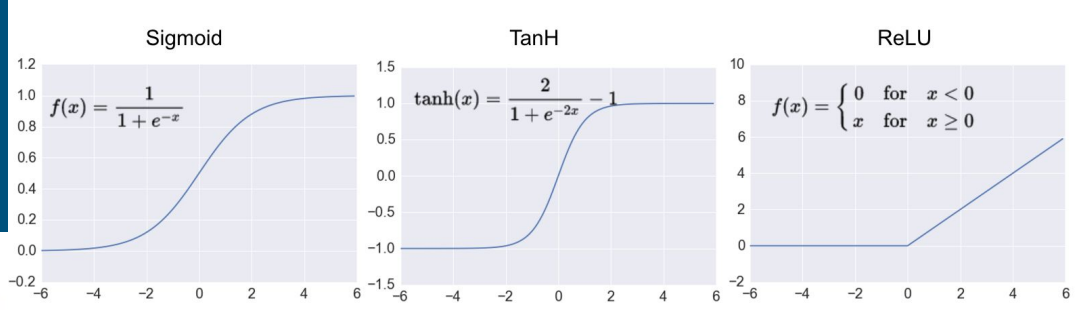


Padding, strides, transposed



Padding, strides, transposed (odd)

Activation : ReLU



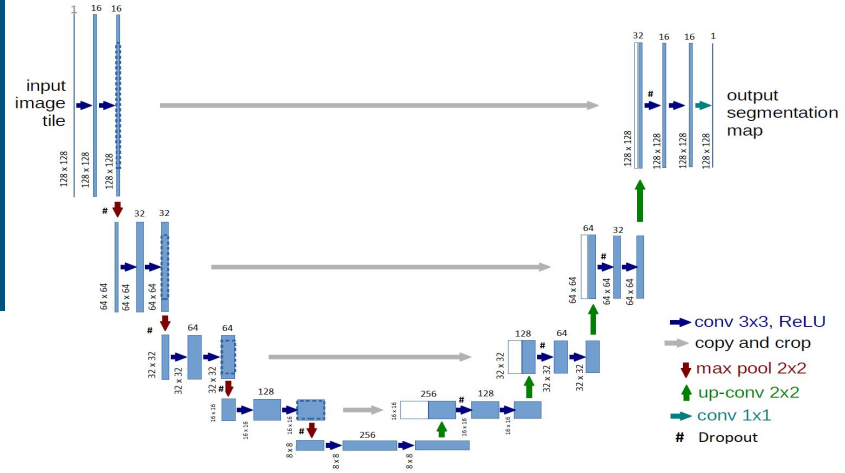
ReLU Layer

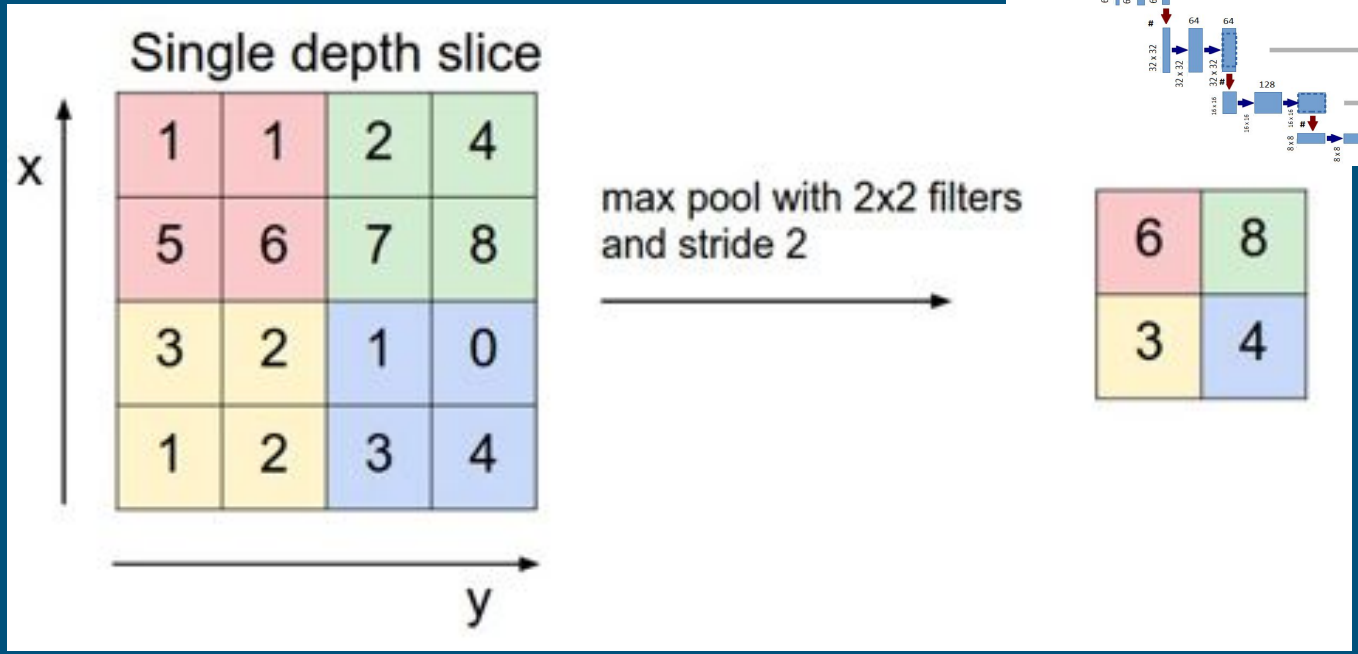
Filter 1 Feature Map

9	3	5	-8
-6	2	-3	1
1	3	4	1
3	-4	5	1

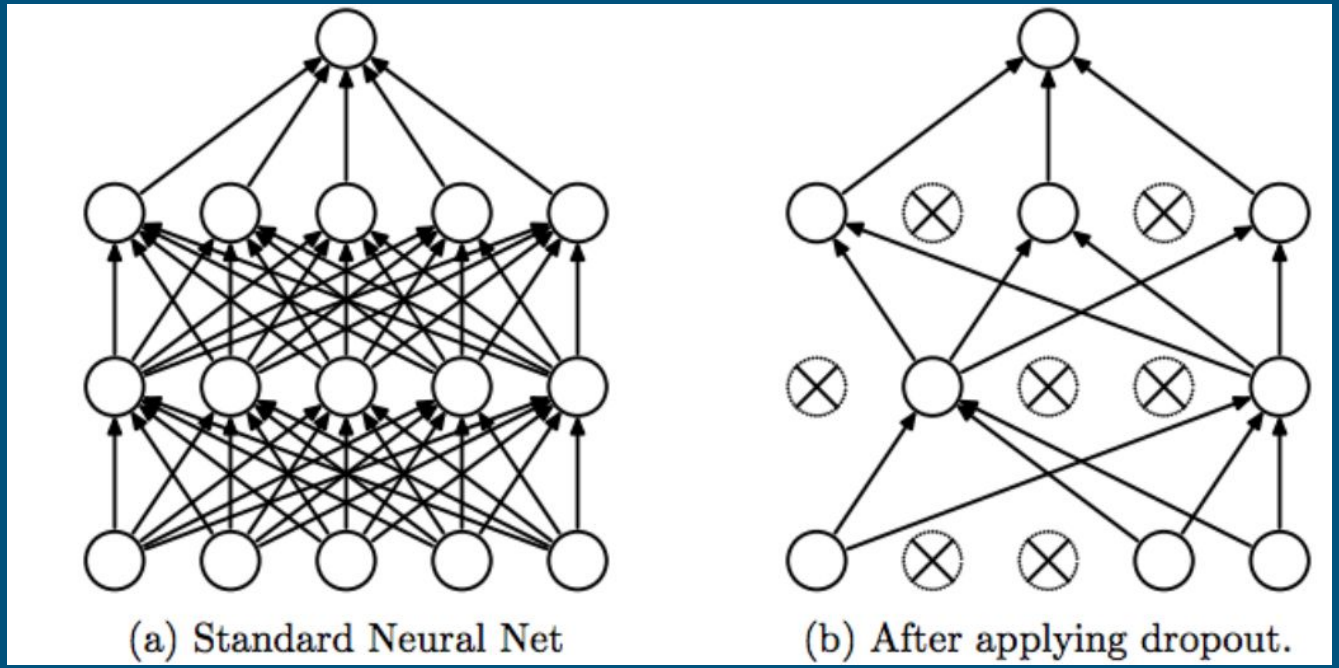
9	3	5	0
0	2	0	1
1	3	4	1
3	0	5	1

MaxPooling


input image tile
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1
Dropout


Single depth slice
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters and stride 2
6 8
3 4

Dropout



C - Exemples

http://scs.ryerson.ca/~aharley/vis/conv/

https://cs.stanford.edu/people/karpathy/convnetjs/

http://cs231n.github.io/convolutional-networks/

III - Training

1. Train/validation split
2. Image preprocessing
3. Image augmentation
4. Metric
5. Model params
6. Model fit

Train/validation split

Train 80%(3200 images) Validation 20%(800 images) ; Stratified

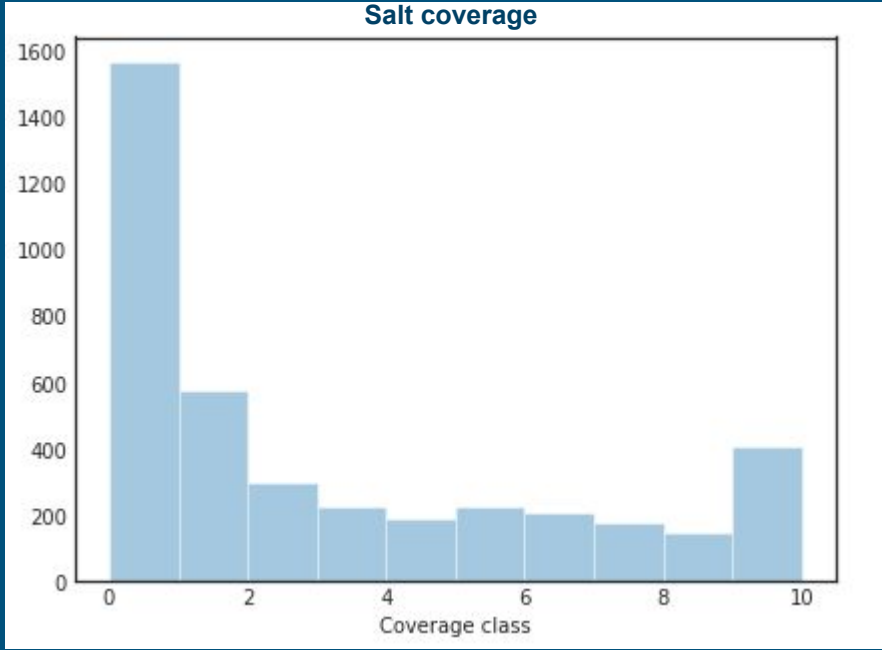


Image preprocessing

Scaling from 101 pixels to 128

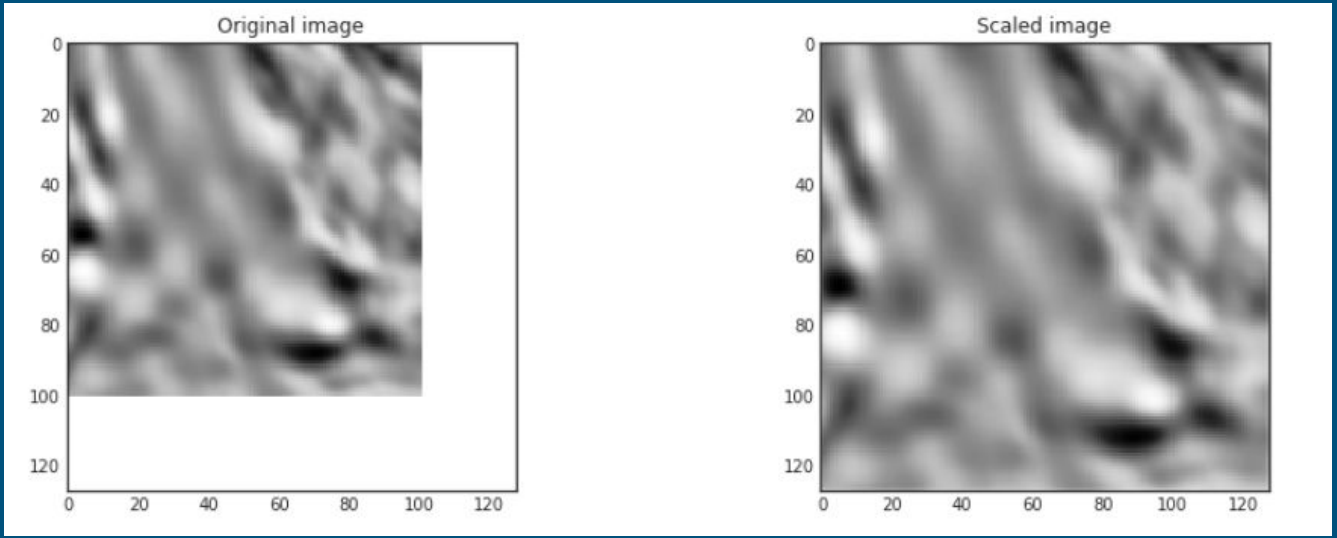


Image augmentation

Horizontal flip

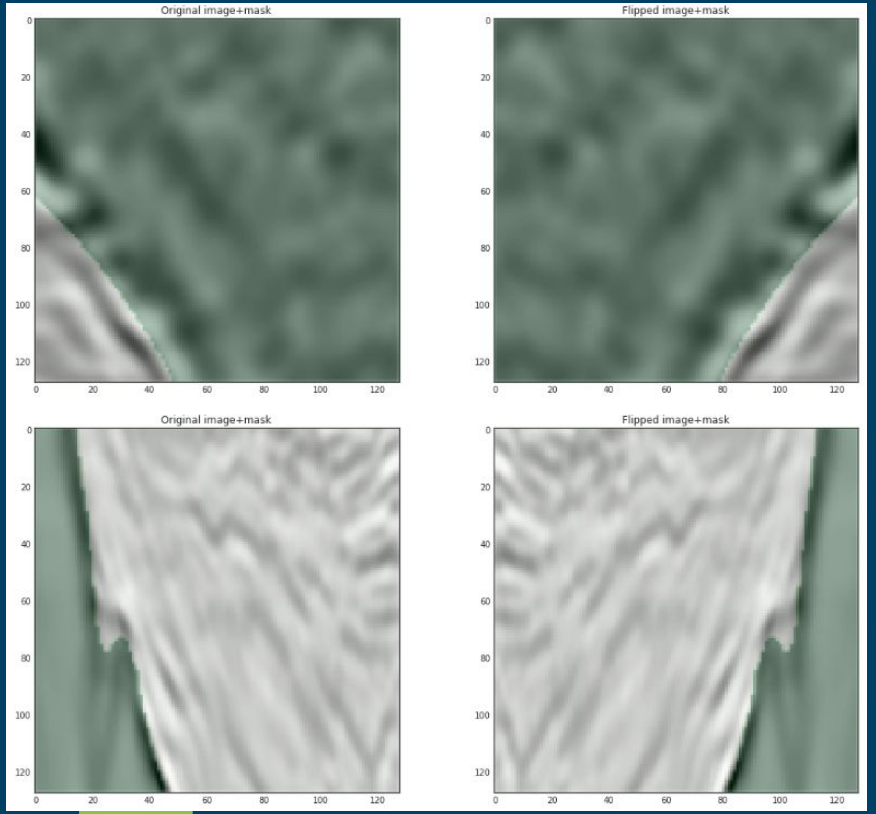


Image augmentation

Central crop (32x32) and resize

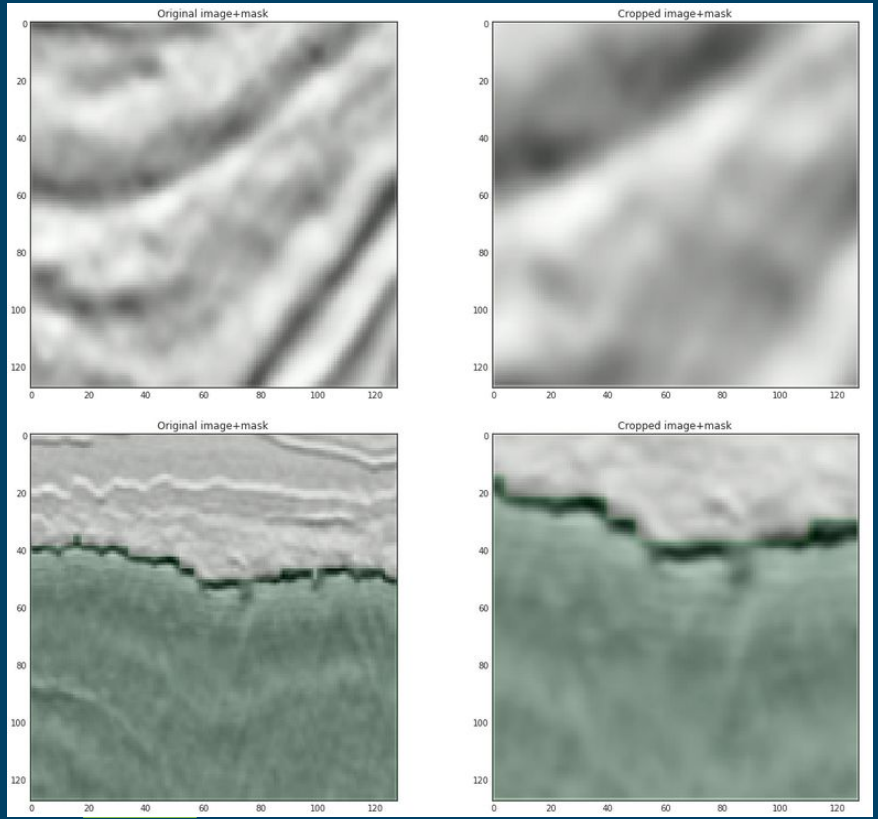
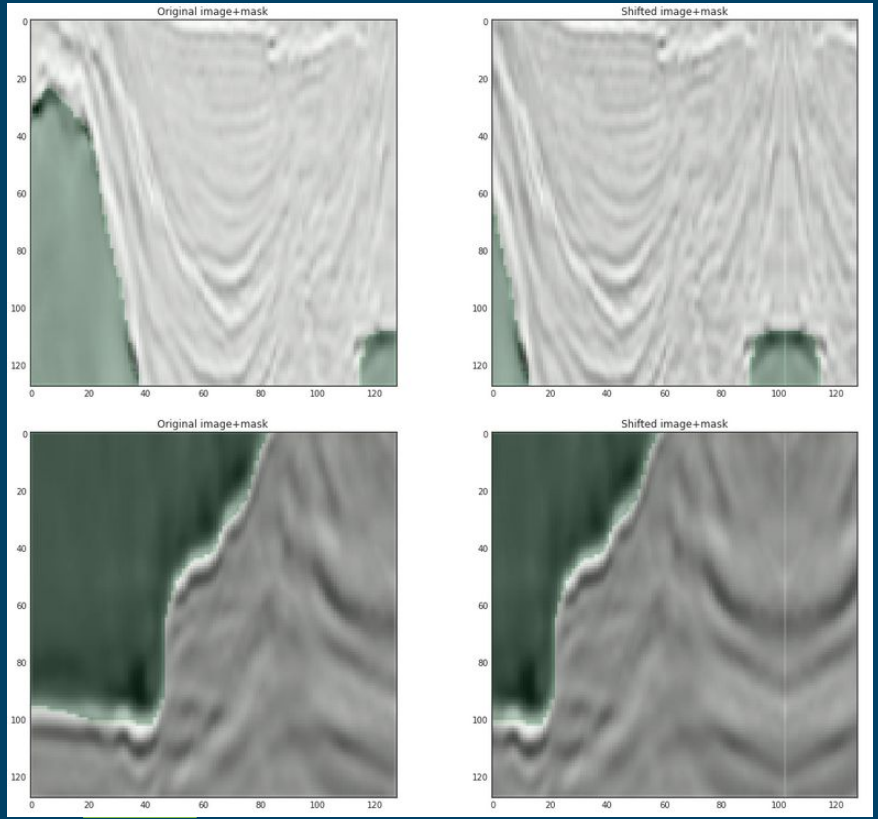


Image augmentation

Shift + flip

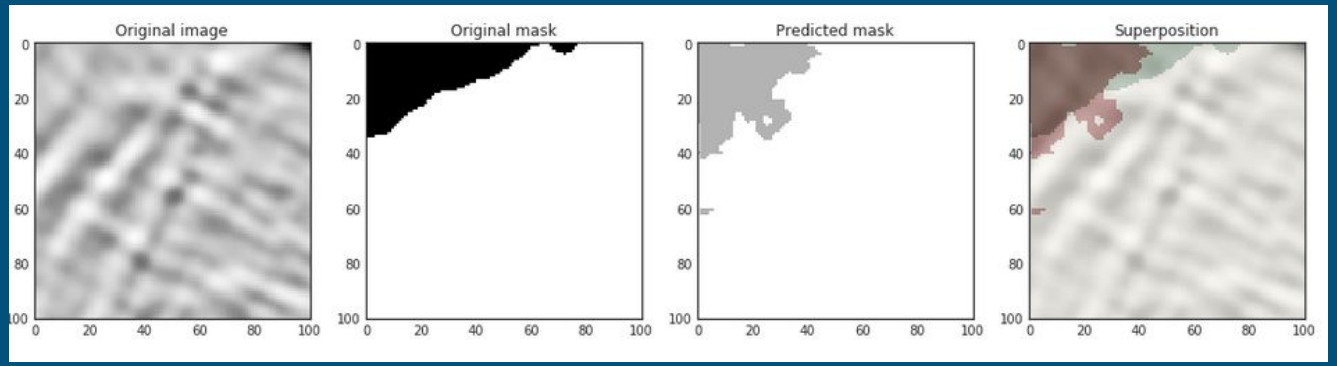


Metric

Intersection over Union (IoU)

$$IoU(A, B) = \frac{A \cap B}{A \cup B}.$$

$$\frac{TP(t)}{TP(t) + FP(t) + FN(t)}.$$

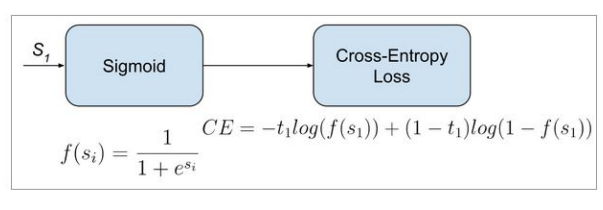


	Prediction	Mask
TP	1	1
FP	1	0
FN	0	1

Model Params

1. Layers: features count (image size) :
   1(128x128) -> 16(64x64) -> 32(32x32) -> 64(16x16) -> 128(8x8) ->
   64(16x16) -> 32(32x32) -> 16(64x64) -> 1(128x128)

2. Loss="binary_crossentropy" (https://gombru.github.io/2018/05/23/cross_entropy_loss/)



3. Optimizer="adam" The Adam optimization algorithm is an extension to stochastic gradient descent, adopted for deep learning.
   https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

4. Metrics=["accuracy"]

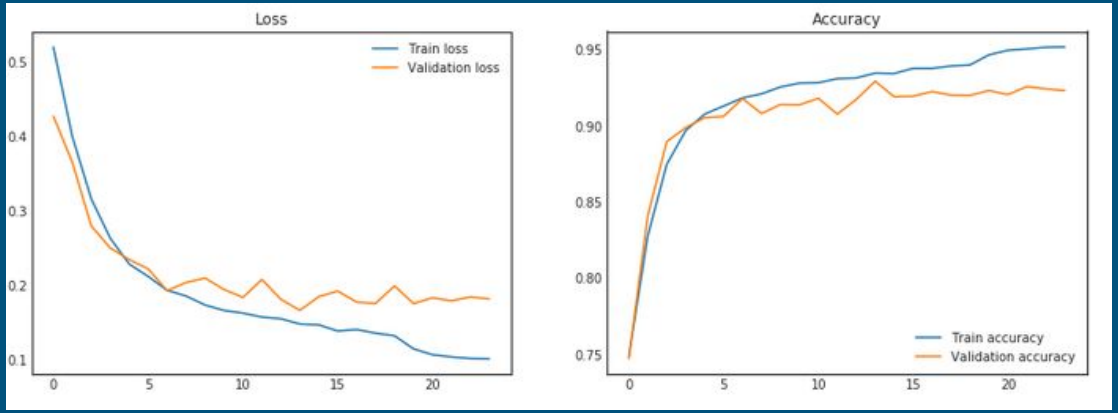
Model fit

+Horizontal flip
Total = 6400 images

epochs = 100;
batch_size = 32

learning_rate = 0.001
early_stopping_patience=10

ReduceLROnPlateau:
factor=0.1, patience=5, min_lr=0.00001



IoU = 0.73

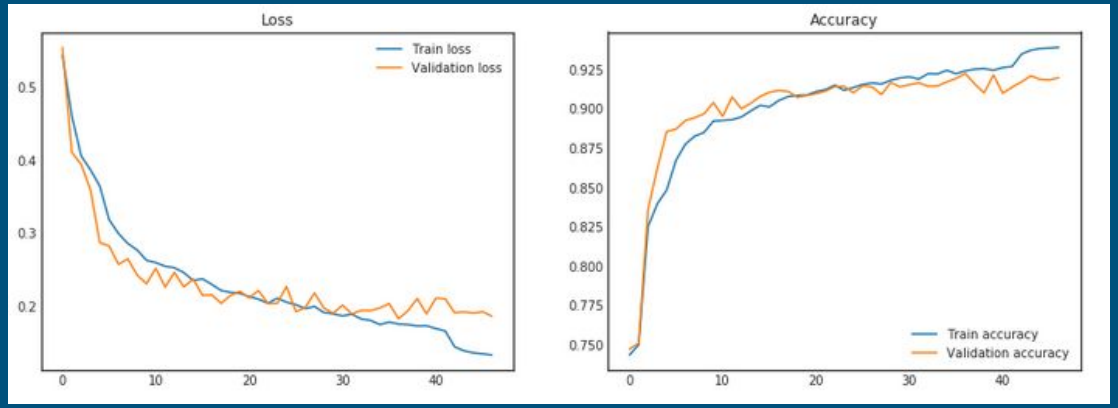
Model fit

+Horizontal flip
+Central crop
Total = 9600 images

epochs = 100;
batch_size = 32

learning_rate = 0.001
early_stopping_patience=10

ReduceLROnPlateau:
factor=0.1, patience=5, min_lr=0.00001



IoU = 0.70

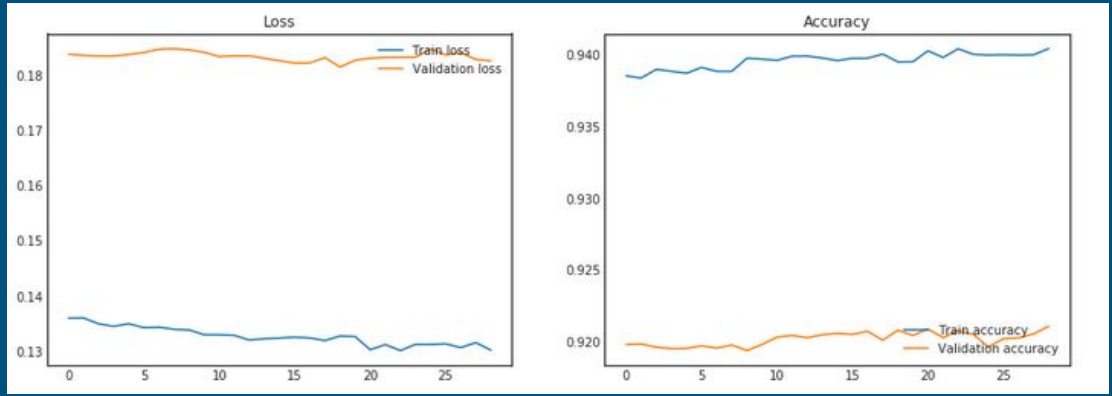
Model fit

+Horizontal flip
+Central crop
+Shift
Total = 12800 images

epochs = 100;
batch_size = 32

learning_rate = 0.001
early_stopping_patience=10

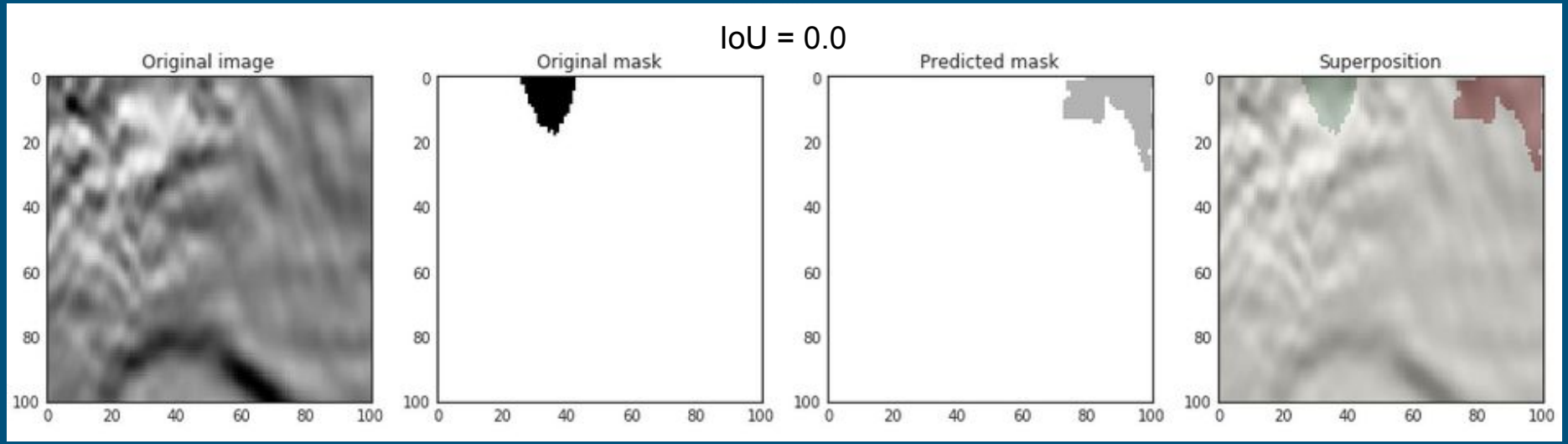
ReduceLROnPlateau:
factor=0.1, patience=5, min_lr=0.00001



IoU = 0.709

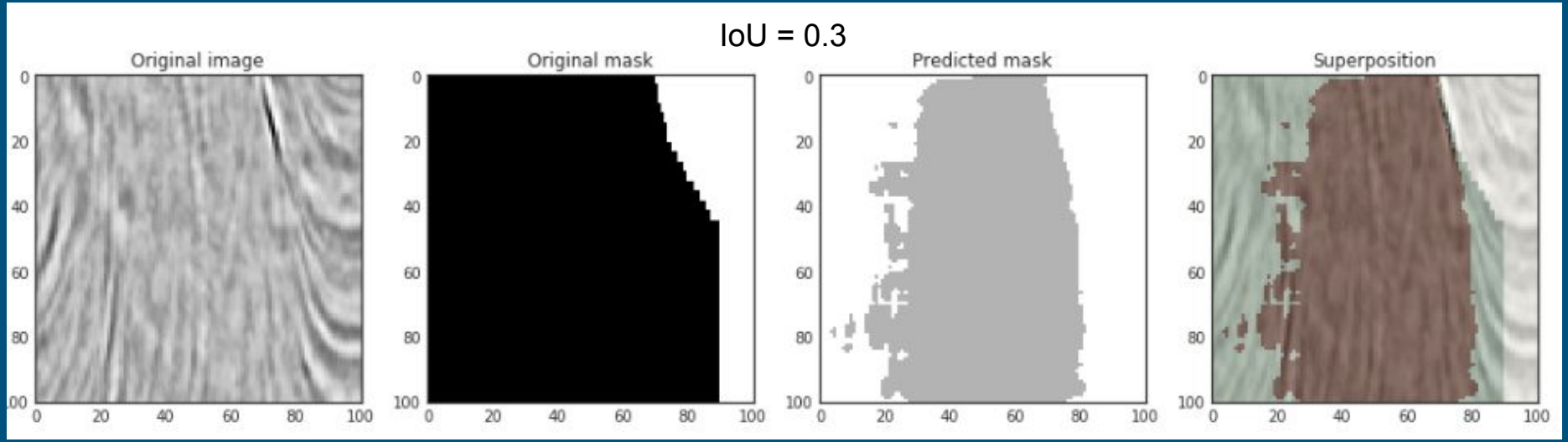
Predictions

$$\frac{TP(t)}{TP(t) + FP(t) + FN(t)}.$$



Predictions

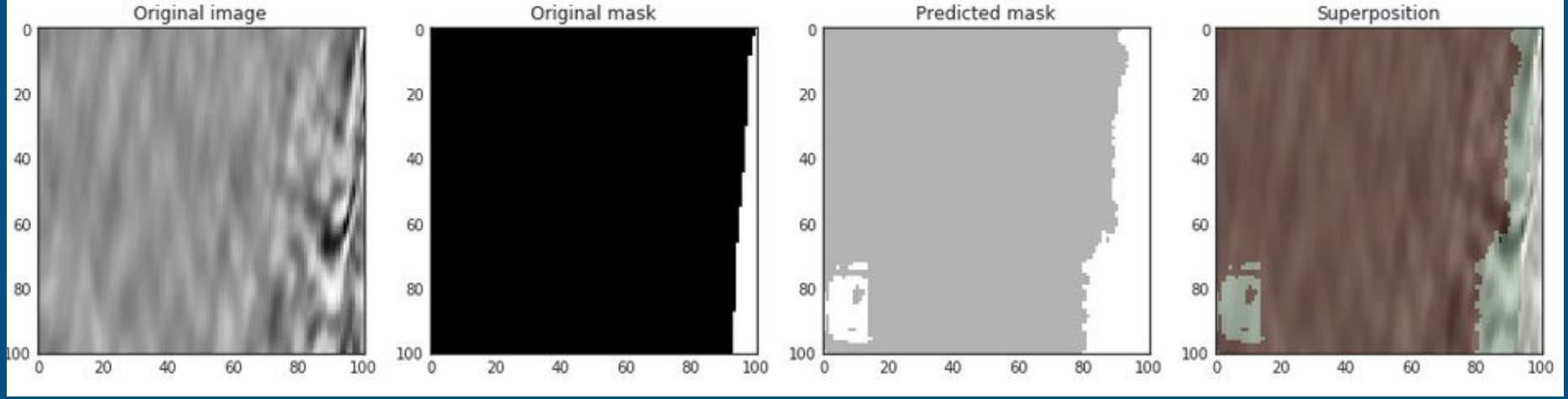
$$\frac{TP(t)}{TP(t) + FP(t) + FN(t)}.$$



Predictions

$$\frac{TP(t)}{TP(t) + FP(t) + FN(t)}.$$

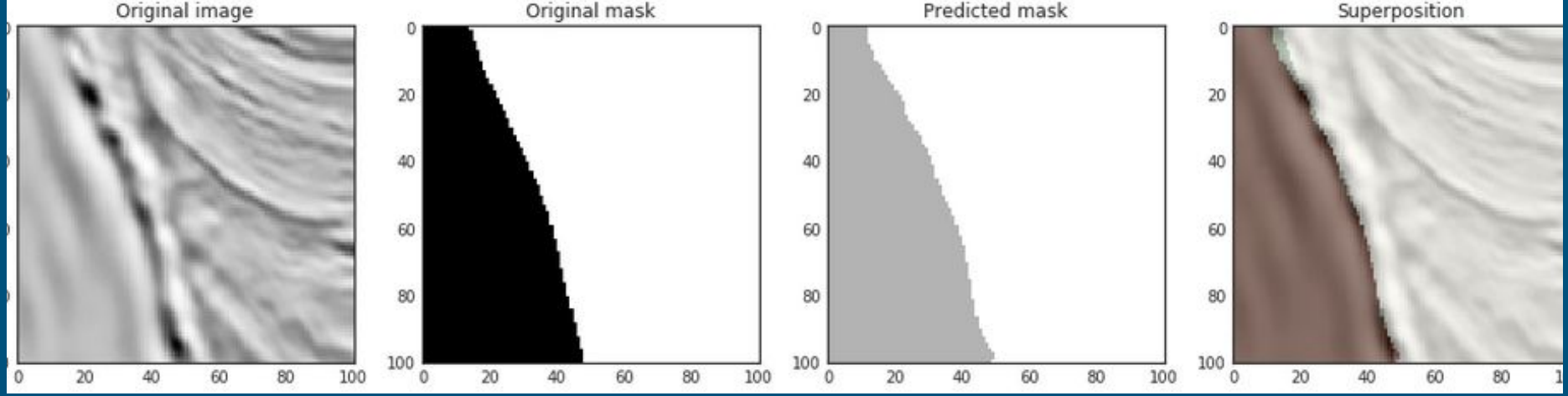
IoU = 0.8



Predictions

$$\frac{TP(t)}{TP(t) + FP(t) + FN(t)}.$$

IoU = 1.0



IV - Pistes d’amélioration

Modifier la fonction d’activation des neurones.

Changer le nombre de convolution successives

Utiliser un padding différent

Changer les valeurs du dropout.

Utiliser Averagepooling

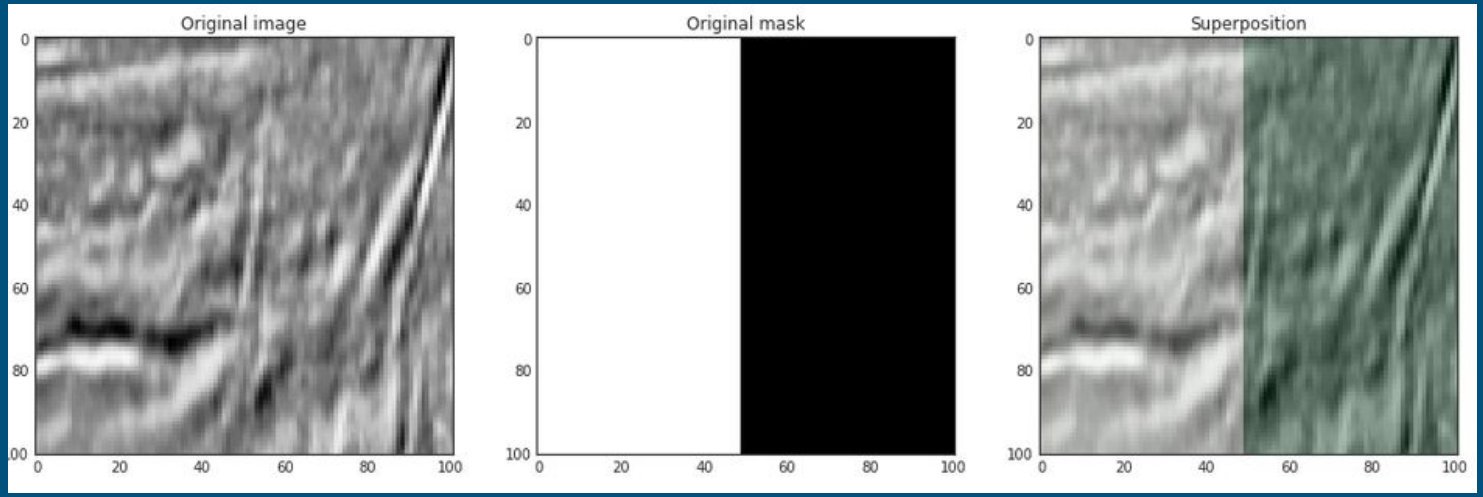
Changer le nombre de filtres / Kernels à chaque convolution

Changer l'architecture du réseau

Difficultés rencontrées

Beaucoup de paramètres + Temps de calcul longs

Masques pas toujours les plus pertinents



Merci

Q&R