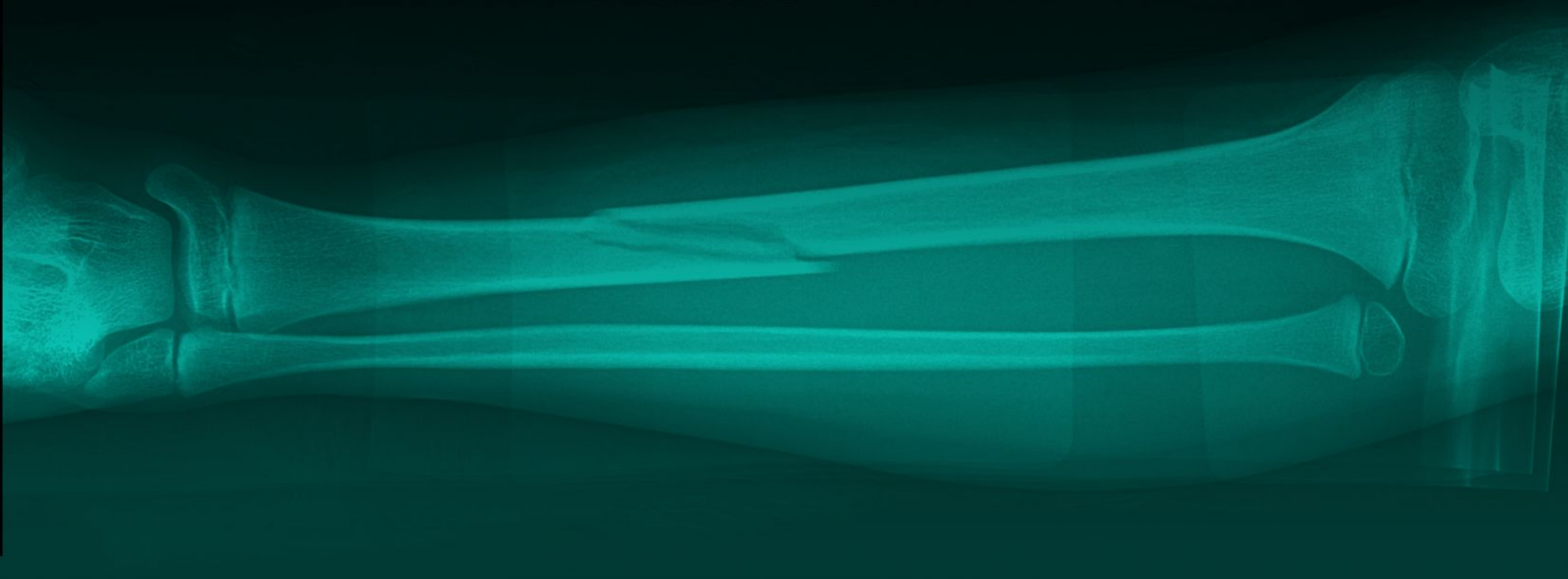
Détection de fractures osseuses

candidat #12184



HBI-120 de *Viken Detection*

Principe général

$$\theta > \varepsilon$$

Détection des bords

Avec l'algorithme Canny

cv2.Canny →

seuils: 40, 120

Détection des bords

Avec l'algorithme Canny

Détection des bords
Avec l'algorithme Canny

Détection des bords
Avec l'algorithme Canny

Non-bord
< seuil bas

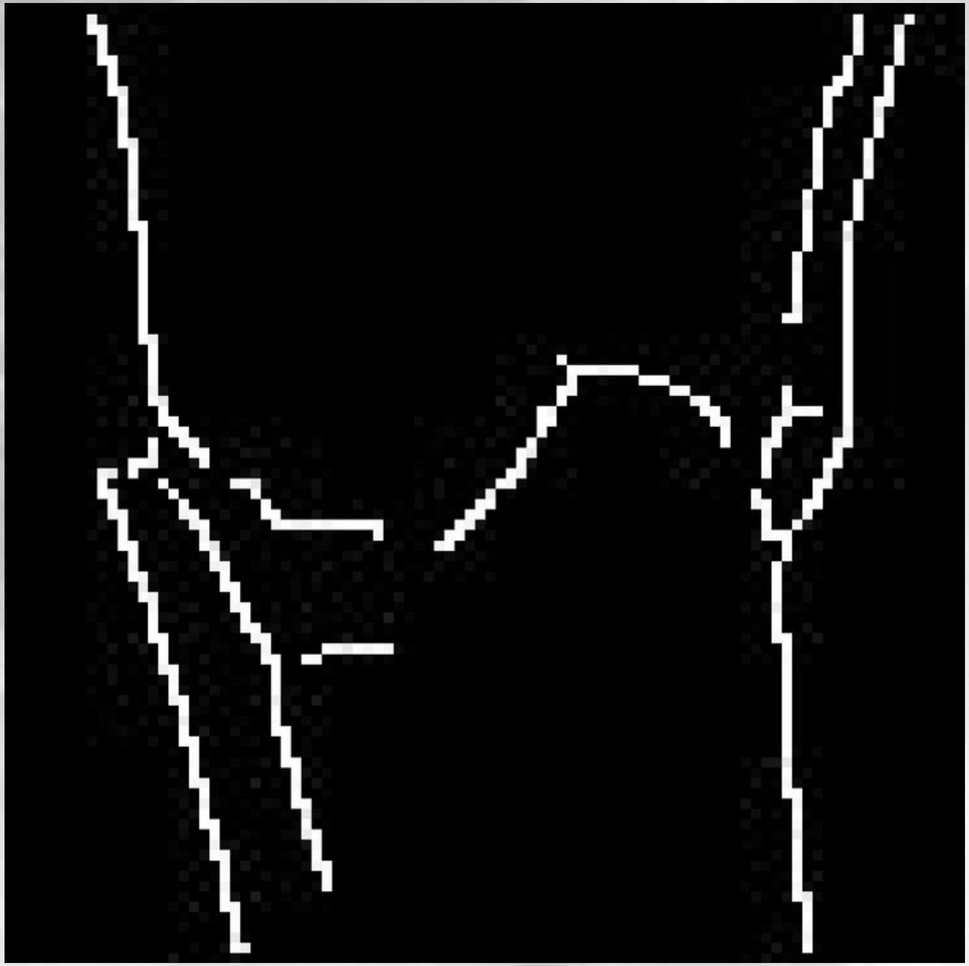
Détection des bords
Avec l'algorithme Canny

Bord faible
∈ [seuil bas, seuil haut[

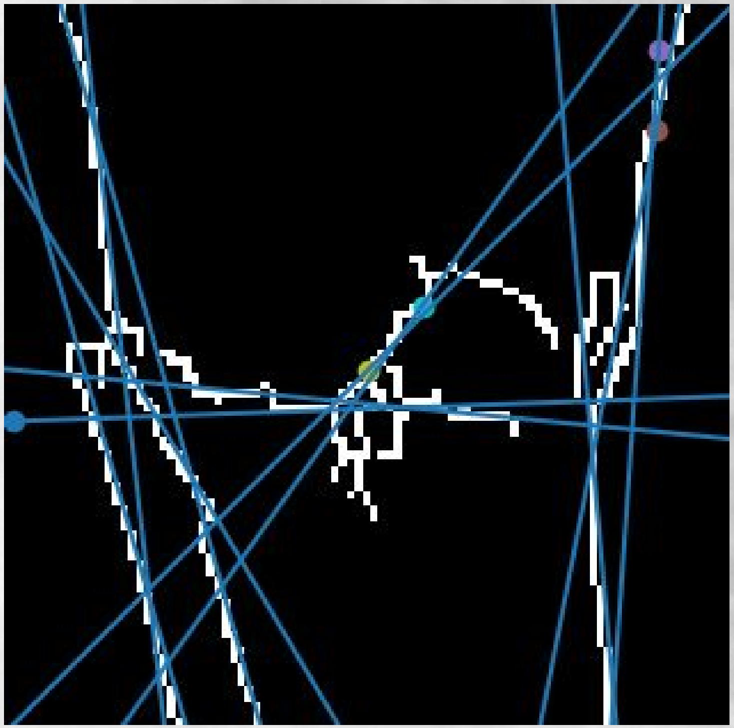
Bord fort
≥ seuil haut

Détection des bords

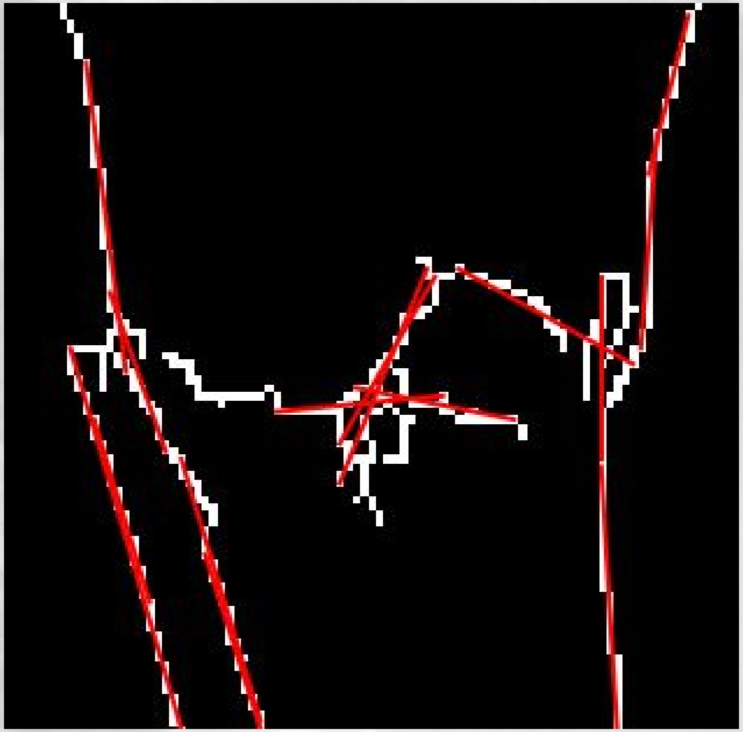
Avec l'algorithme Canny



Détection des traits
Avec la Transformée de Hough



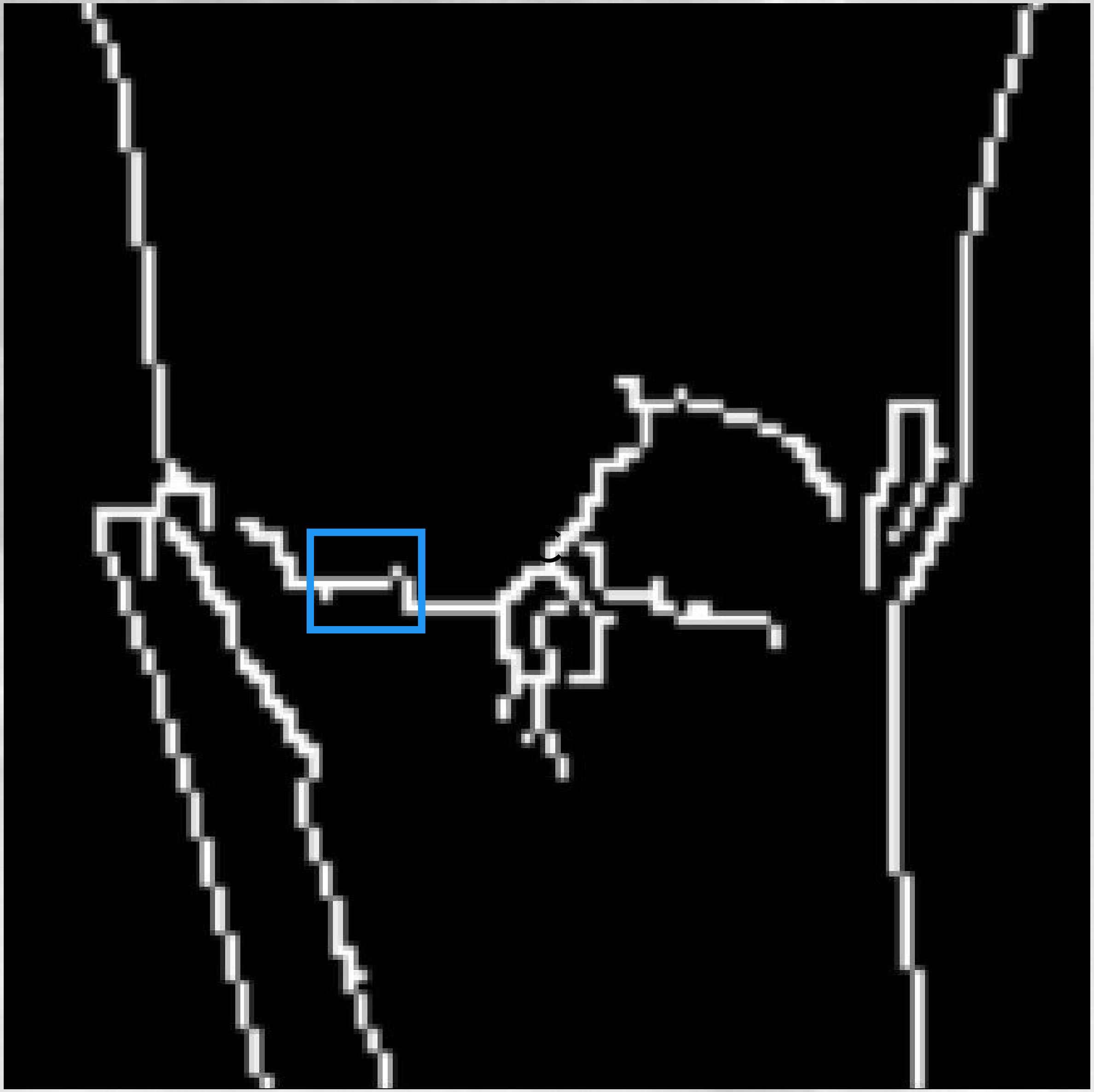
Classique
(détecte des droites)



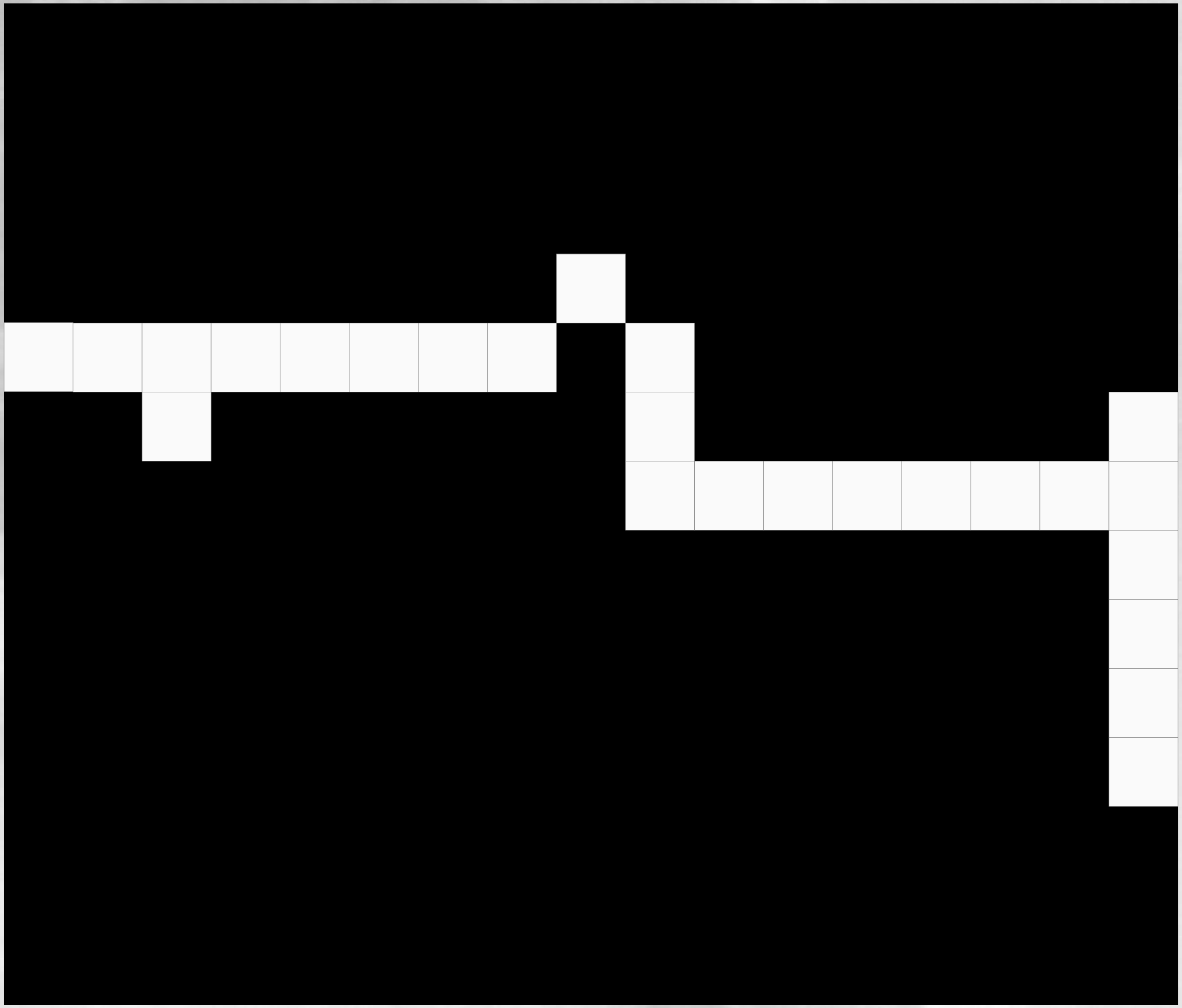
Probabiliste
(détecte des segments)

Détection des traits

Avec la Transformée de Hough

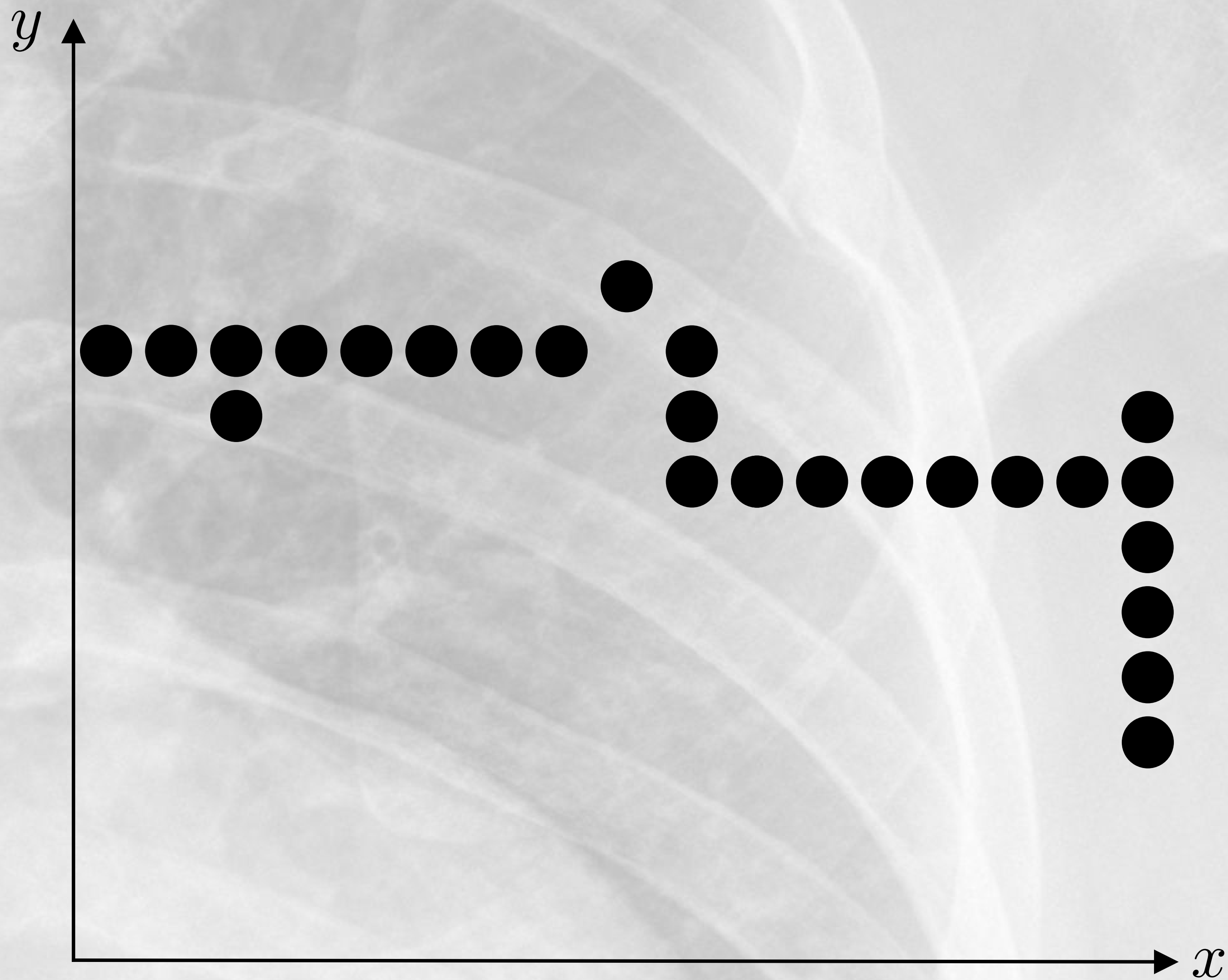


Détection des traits
Avec la Transformée de Hough



Détection des traits

Avec la Transformée de Hough



Détection des traits

Avec la Transformée de Hough

Détection des traits

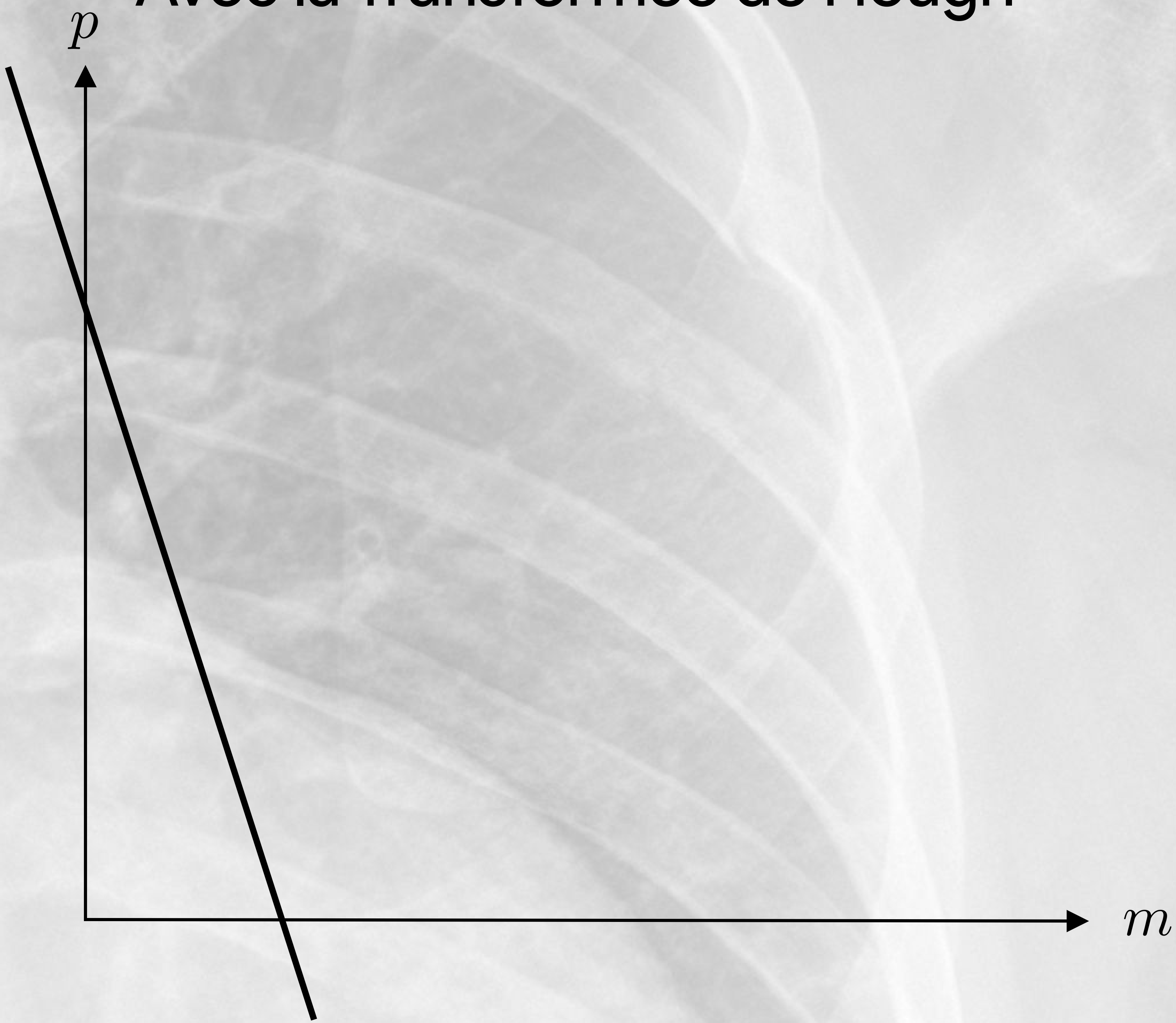
Avec la Transformée de Hough

$$y = mx + p$$

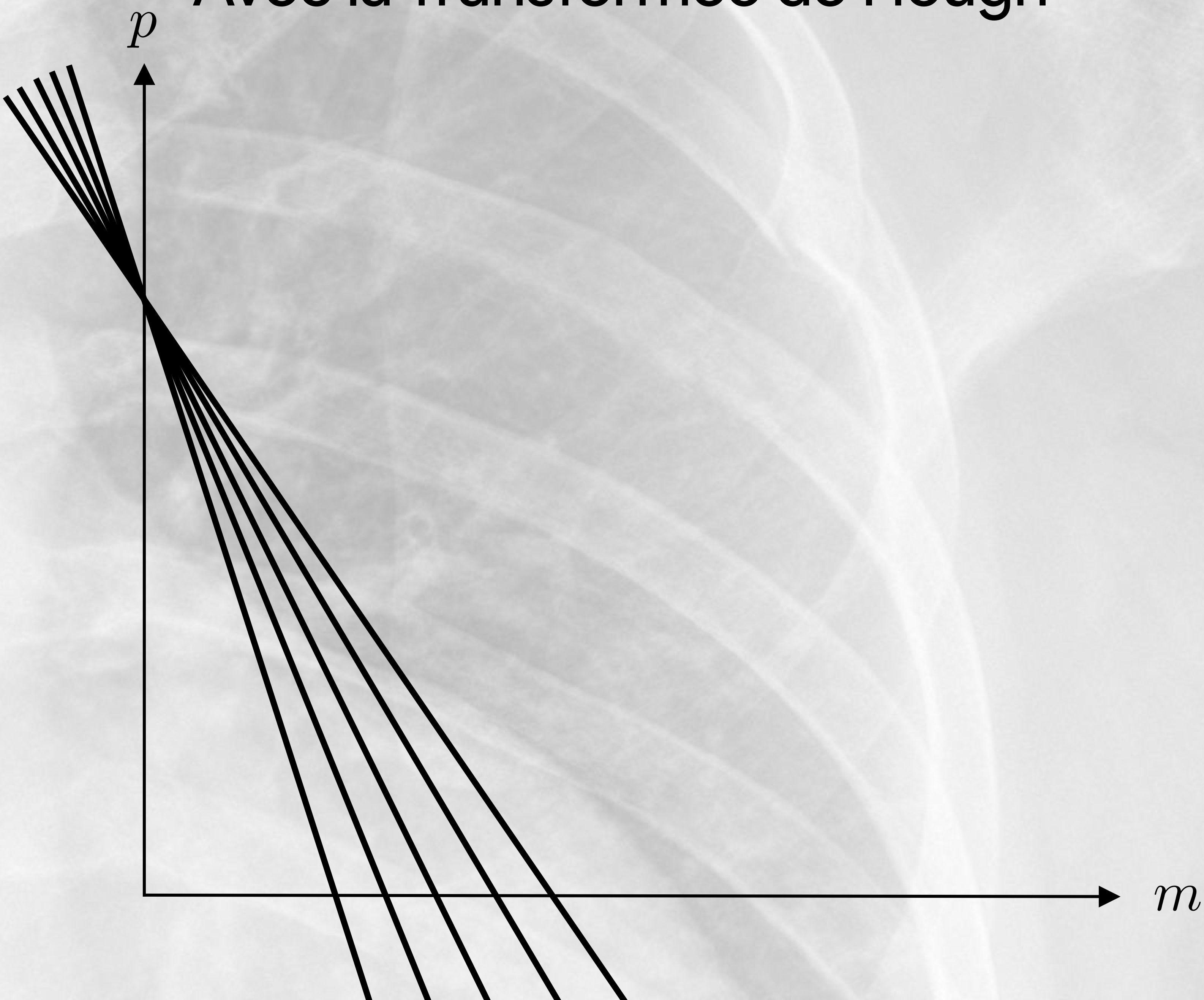
$$10 = m(4) + p \iff p = 10 - 4m$$

Détection des traits

Avec la Transformée de Hough

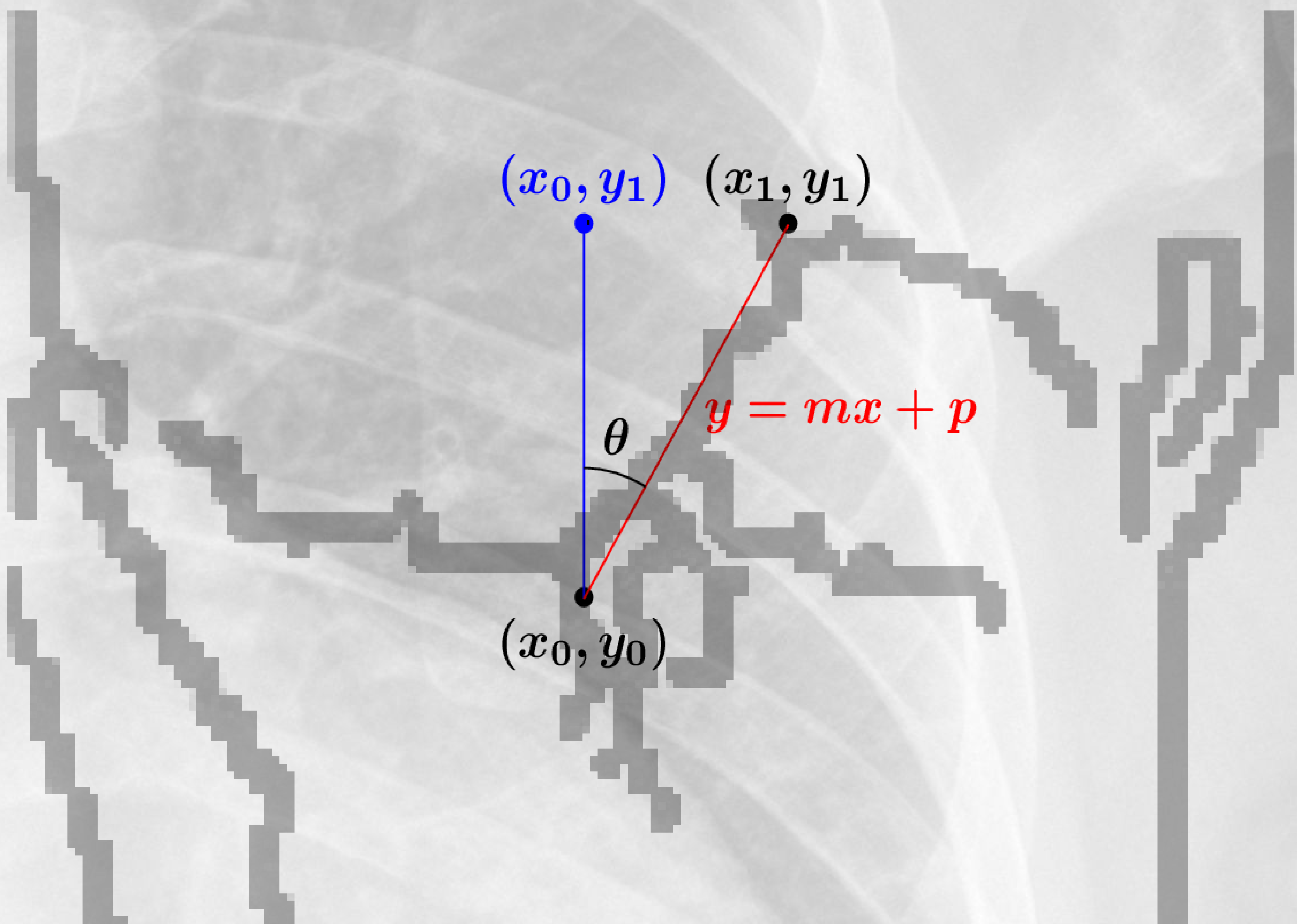


Détection des traits
Avec la Transformée de Hough



Calcul des angles

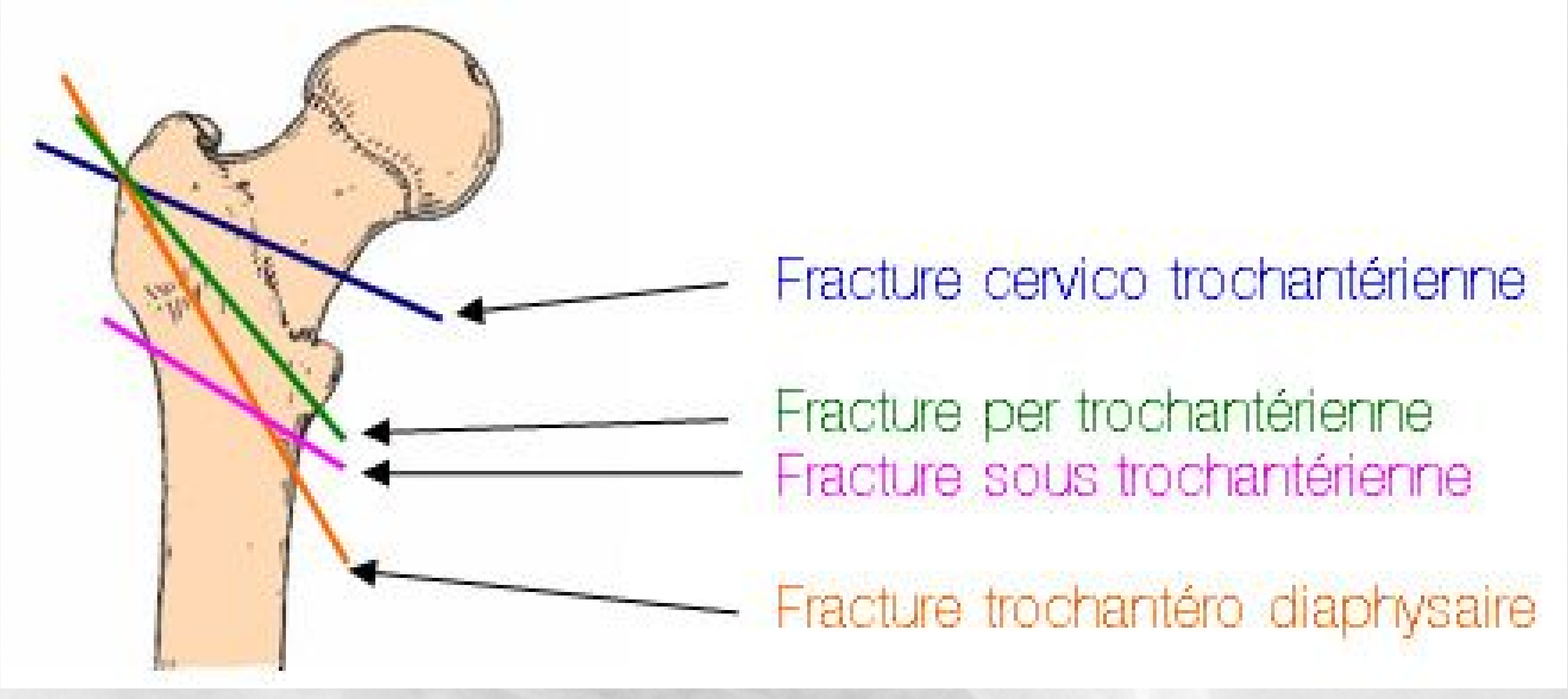
Avec de la trigonométrie



Critère de décision

$$\max \text{angles} > \varepsilon \iff \text{cassé}$$

Identification du type de fracture



Noms des différentes *lignes de fracture* du fémur

Détection des bords

Un problème de texture

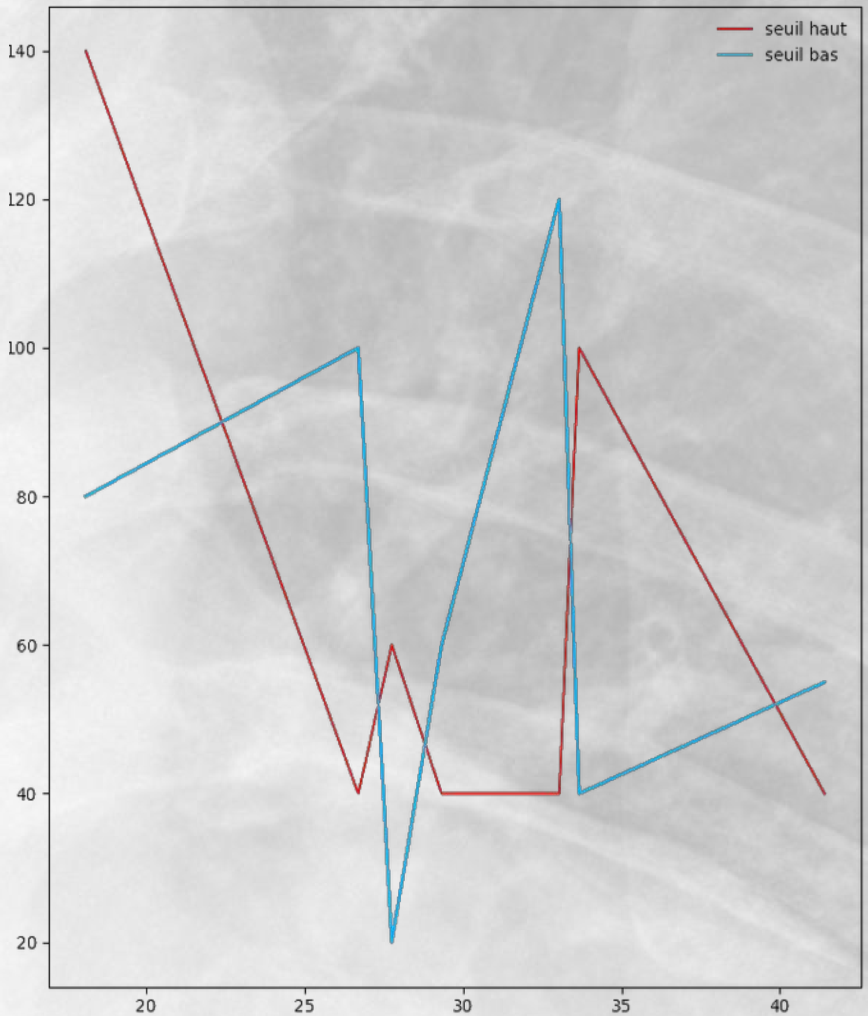
`cv2.Canny`

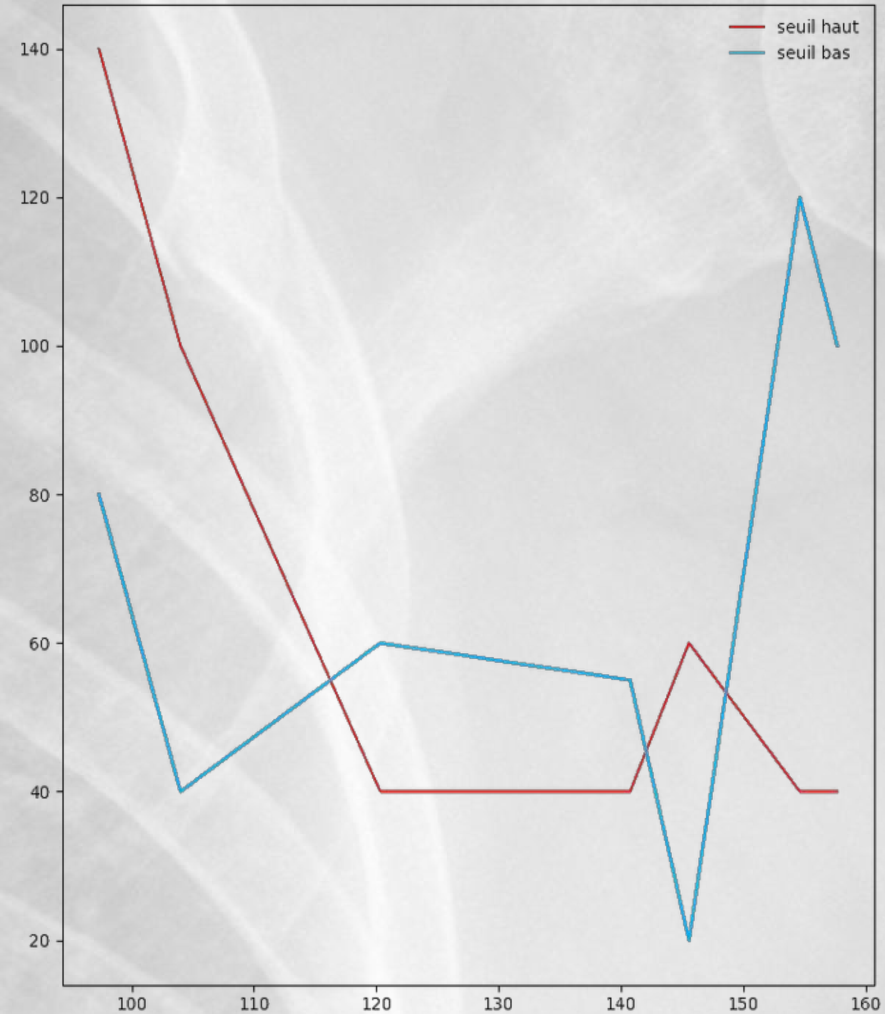
```
bas:  40
haut: 60
```

```
bas:  40
haut: 120
```

```
bas:  60
haut: 180
```

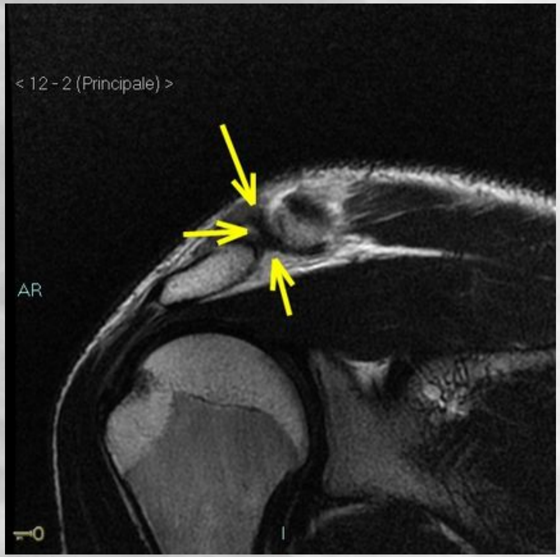
seuils(luminosité, contraste) ?





Seuils optimaux de détection de bords

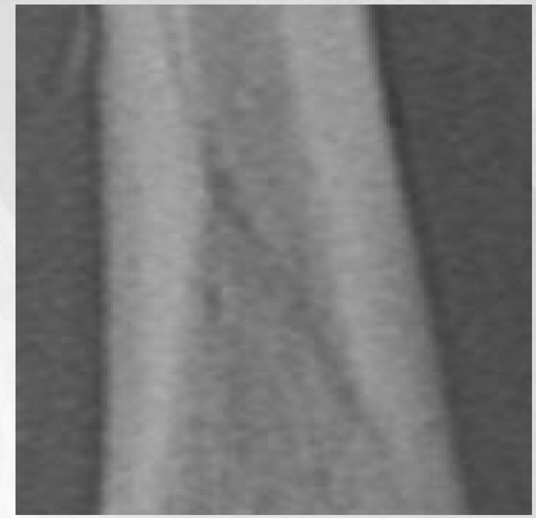
Recherche de sets de données



Recherche de sets de données

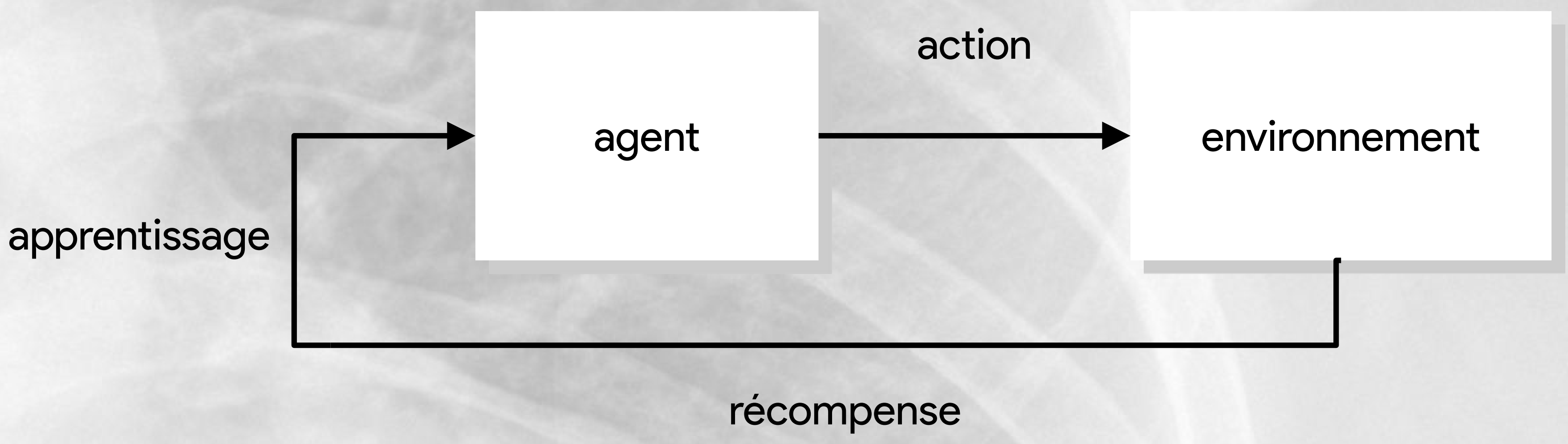
bas 40
haut 120

bas 27
haut 44

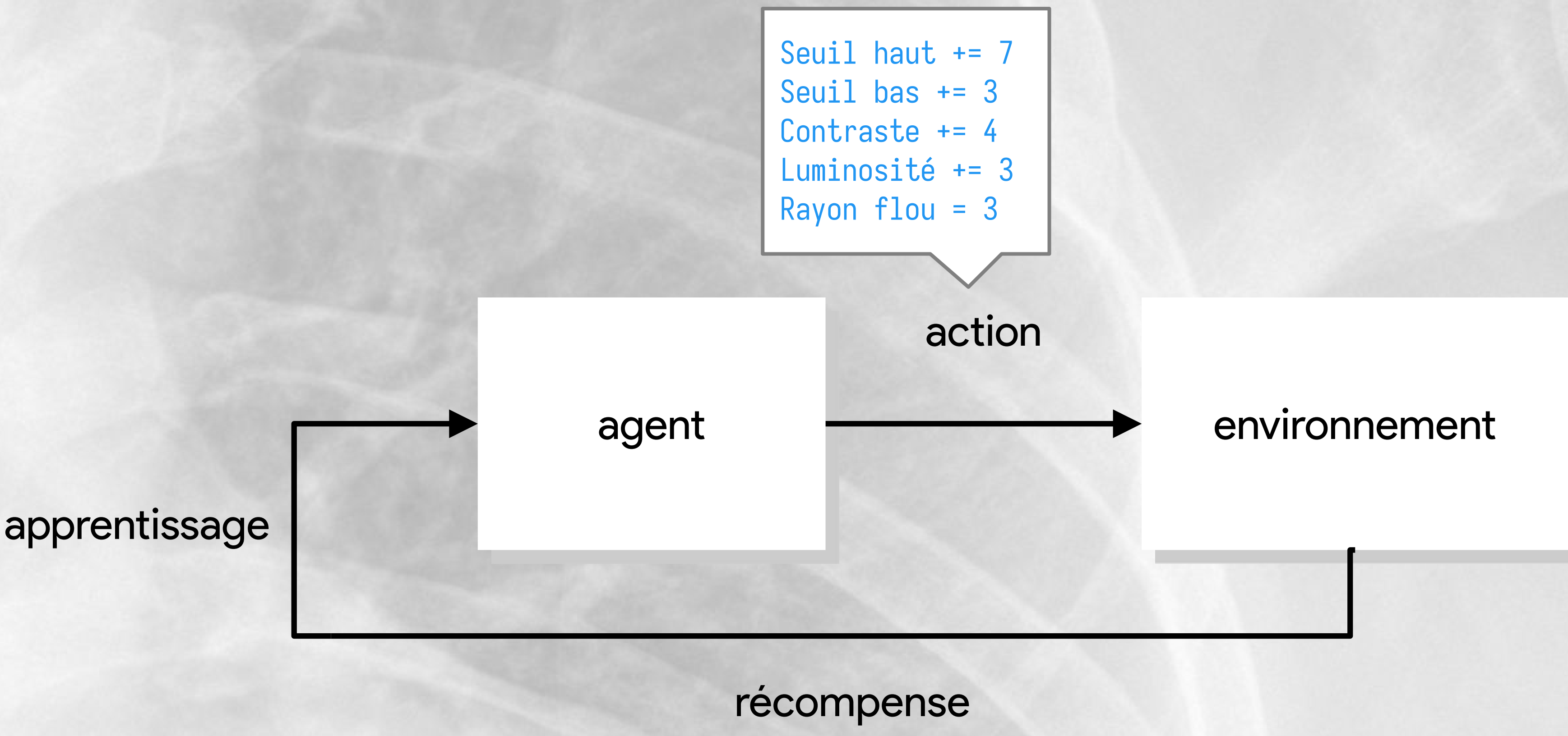


bas 20
haut 21

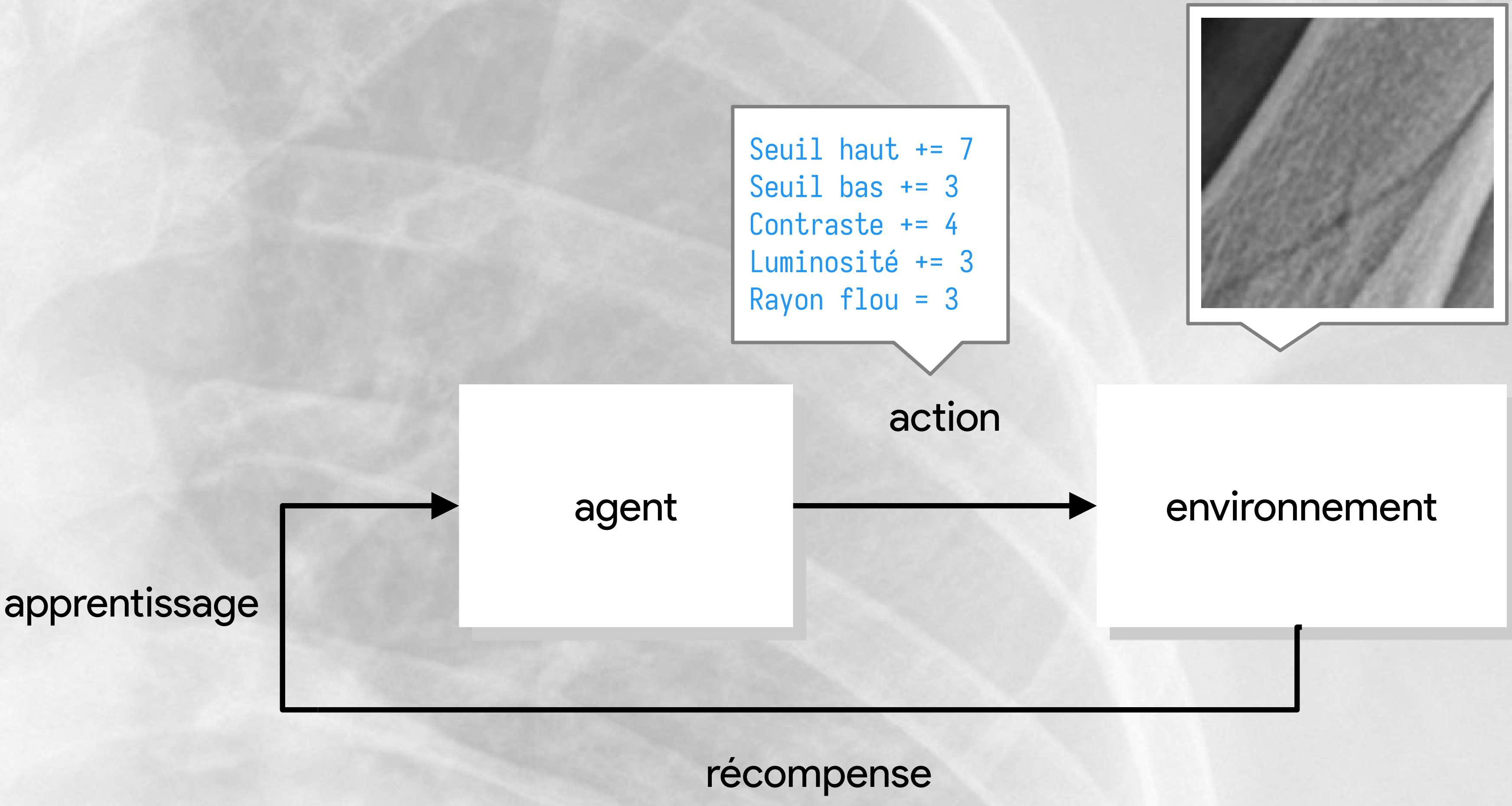
Apprentissage par renforcement



Apprentissage par renforcement



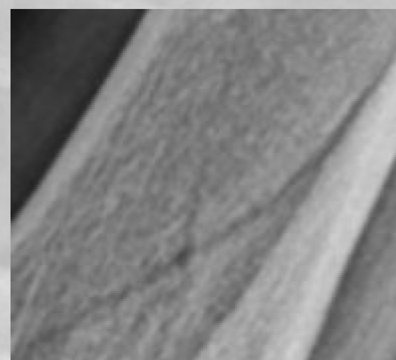
Apprentissage par renforcement



Apprentissage par renforcement

action

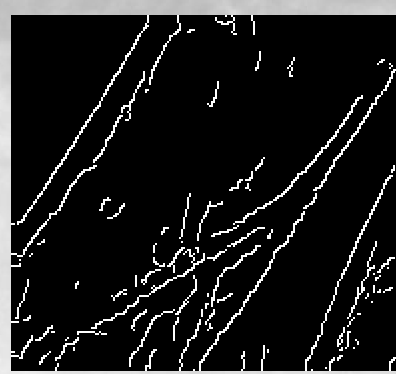
$\text{cont}, \text{lum}, r_{\text{flou}}, \text{seuil}_h, \text{seuil}_b$



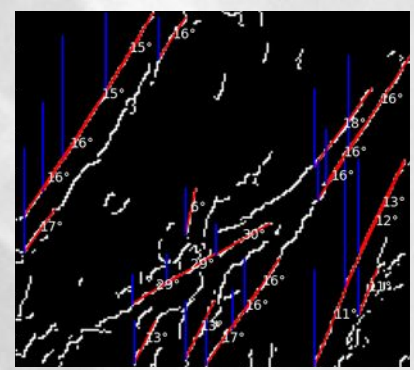
S

Environnement

$T = \underset{r_{\text{flou}}}{\text{Flou}}\left(S \cdot \text{cont} + \text{lum}\right)$

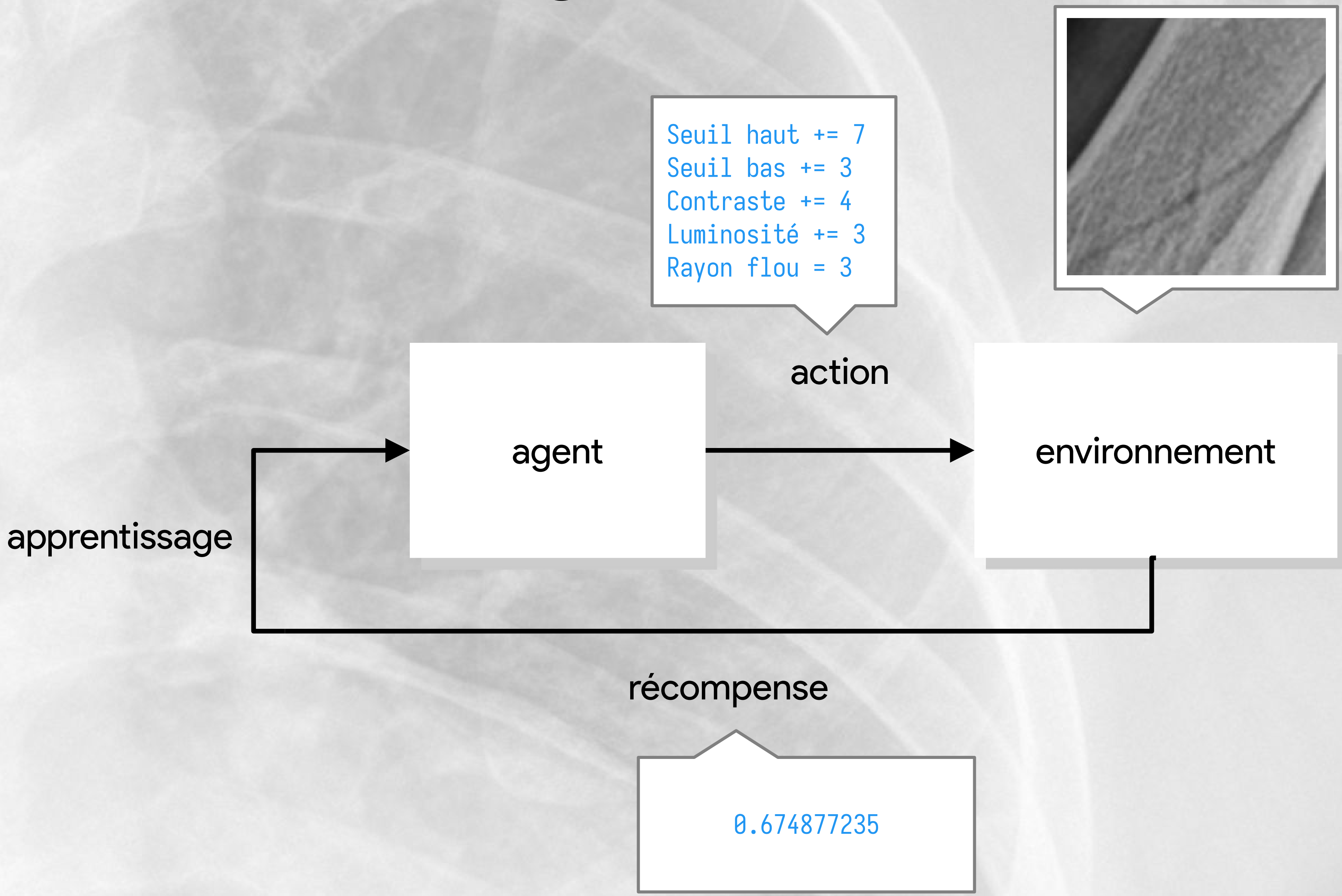


$B = \underset{\text{seuil}_h, \text{seuil}_b}{\text{Canny}} \; T$



$L = \{(i, f) \in \text{Hough}(B),\ \|i - f\| \geq 20\}$

Apprentissage par renforcement



Calcul de la récompense

$$\begin{cases} 1 - d(\mathrm{lum}\, B, 7, 15) & \text{si } d(\mathrm{lum}\, B, 7, 15) \neq 0 \\ 0.25 + 1 - d(|L|, 10, 25) & \text{sinon} \end{cases}$$

avec

$$d := (v, a, b) \mapsto \begin{cases} |v - a| & \text{si } v < a \\ |v - b| & \text{si } v > b \\ 0 & \text{sinon} \end{cases}$$

Apprentissage de l'agent

avec des *Q-Tables*

	État 1	État 2	État 3	…
Action 1	0.1244	0.3409	0.7574	0.7269
Action 2	0.8476	0.4427	0.3895	0.8374
Action 3	0.8479	0.7761	0.0762	0.7884
…	0.1121	0.4661	0.9433	0.1774

Apprentissage de l’agent

Le problème de dimension des *Q-Tables*

$$8^{200 \cdot 200}$$

$$20 \cdot 20 \cdot 6 \cdot 10 \cdot 5$$

	État 1	État 2	État 3	...
Action 1 `Seuil haut -= 10`	0.1244	0.3409	0.7574	0.7269
Action 2 `Seuil haut -= 9`	0.8476	0.4427	0.3895	0.8374
Action 3 `Seuil haut -= 8`	0.8479	0.7761	0.0762	0.7884
...	0.1121	0.4661	0.9433	0.1774

Apprentissage de l'agent

Le problème de dimension des *Q-Tables*

$$8^{200\cdot 200}\cdot 20\cdot 20\cdot 6\cdot 10\cdot 5\cdot 11 \quad \text{octets} \;\approx$$

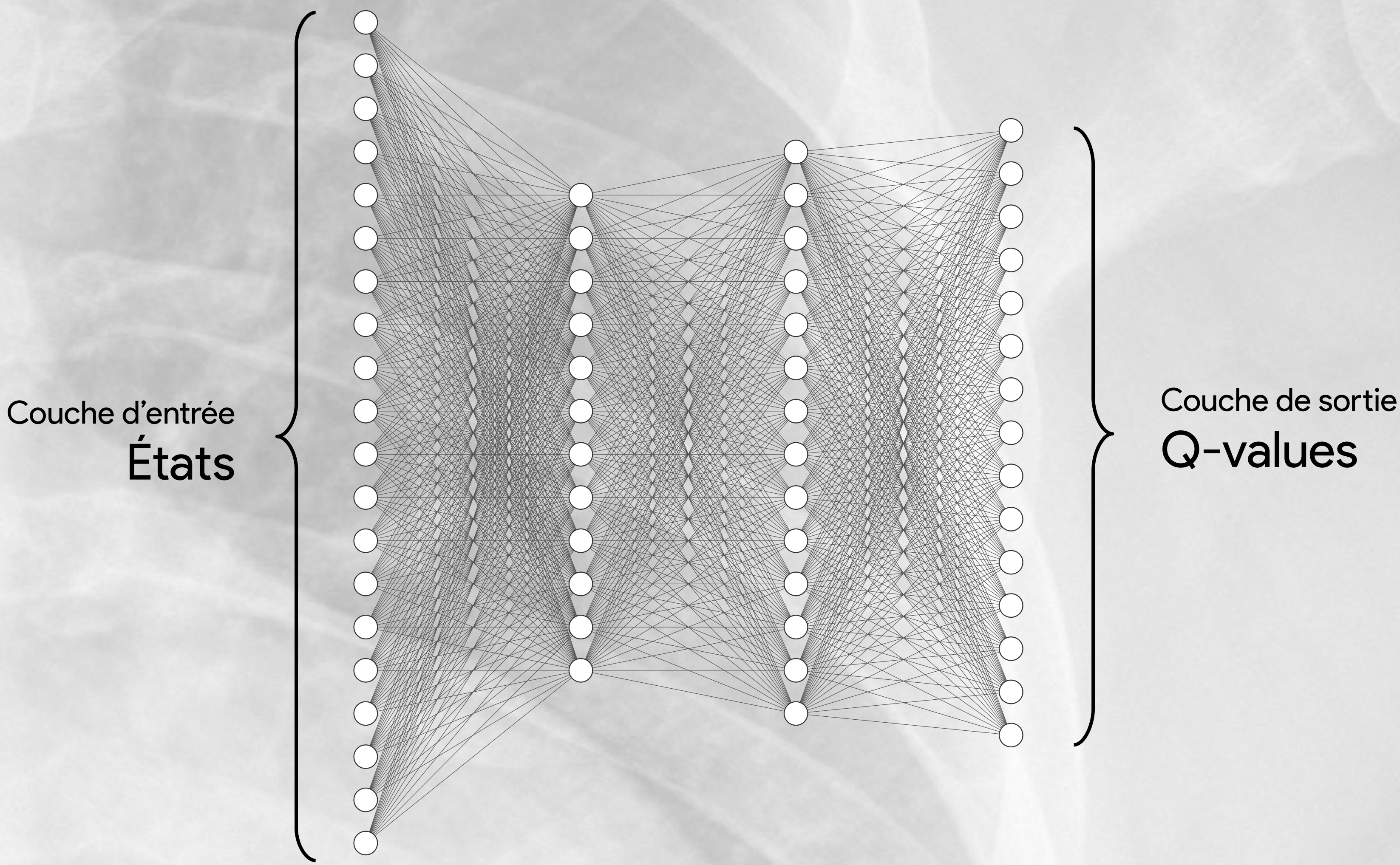
$$5.25\cdot 10^{36114}\ \text{Po}$$

Taille du (deep) web en 2014

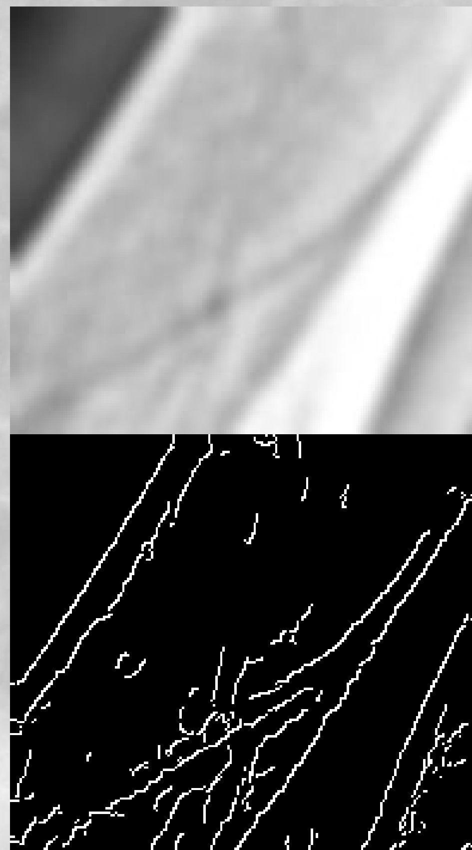
$$91\ \text{Po}$$

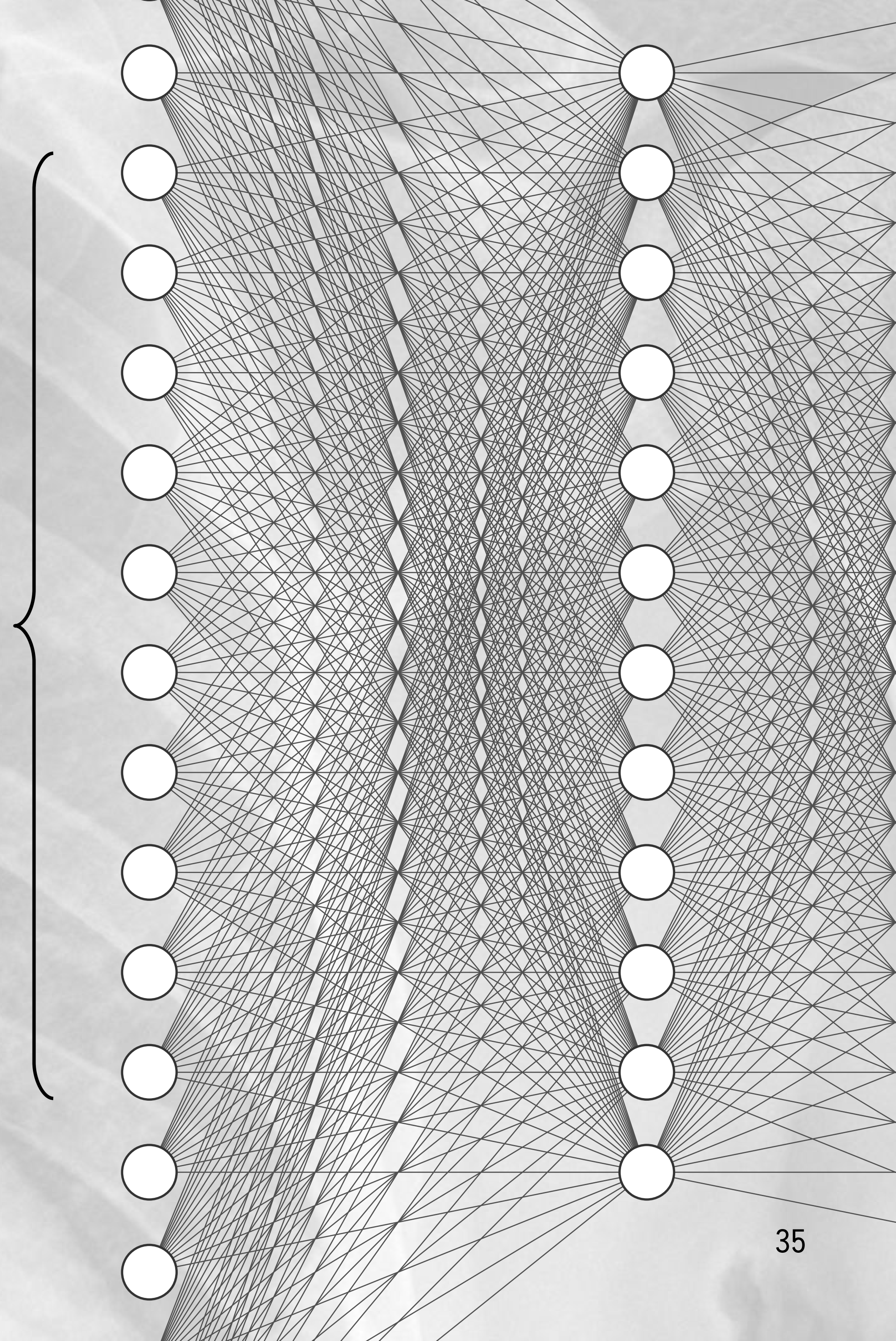
Apprentissage de l'agent

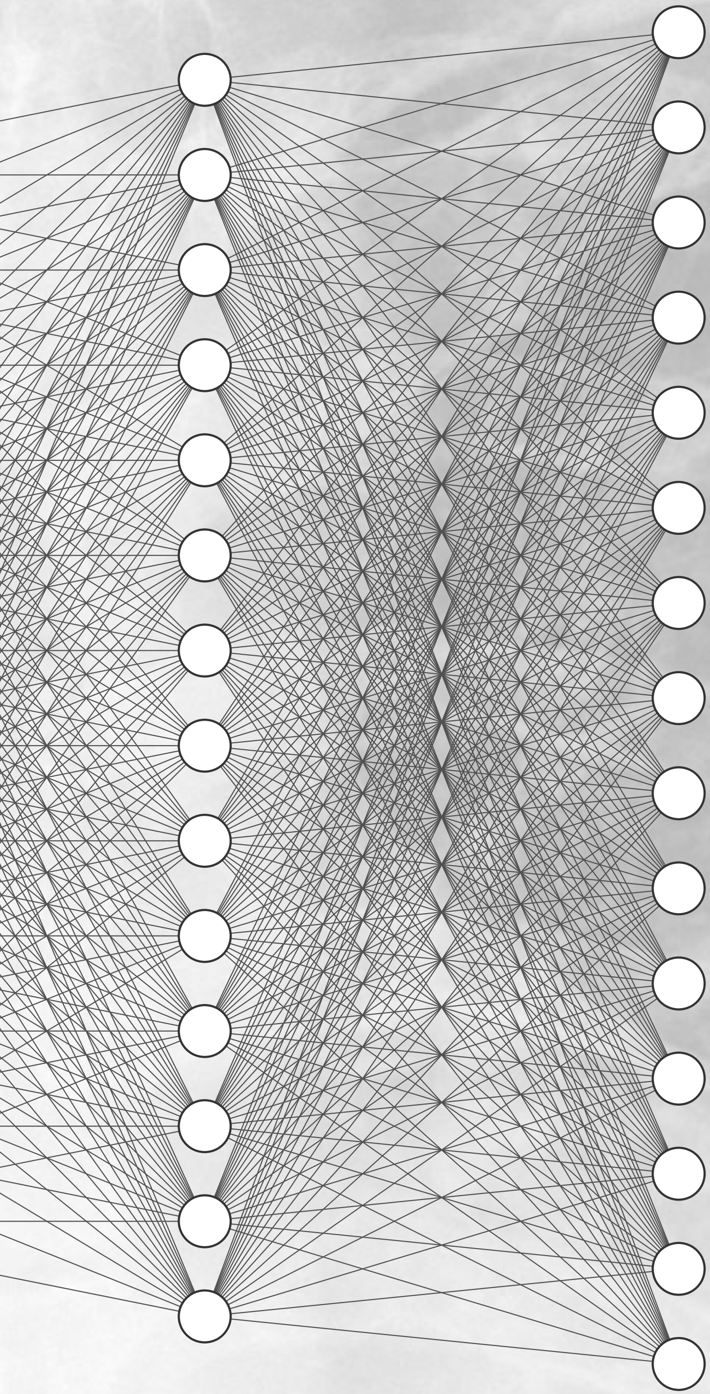
Avec des *Deep-Q Networks*



Un état







Incréments du seuil haut

$$\mathrm{seuil}_h \in \{-10, -9, \ldots, 8, 9, 10\}$$

Incréments du seuil bas

$$\mathrm{seuil}_b \in \{-10, -9, \ldots, 8, 9, 10\}$$

Incréments du contraste

$$\mathrm{cont} \in \{1, 1.1, 1.2, 1.3, 1.4, 1.5\}$$

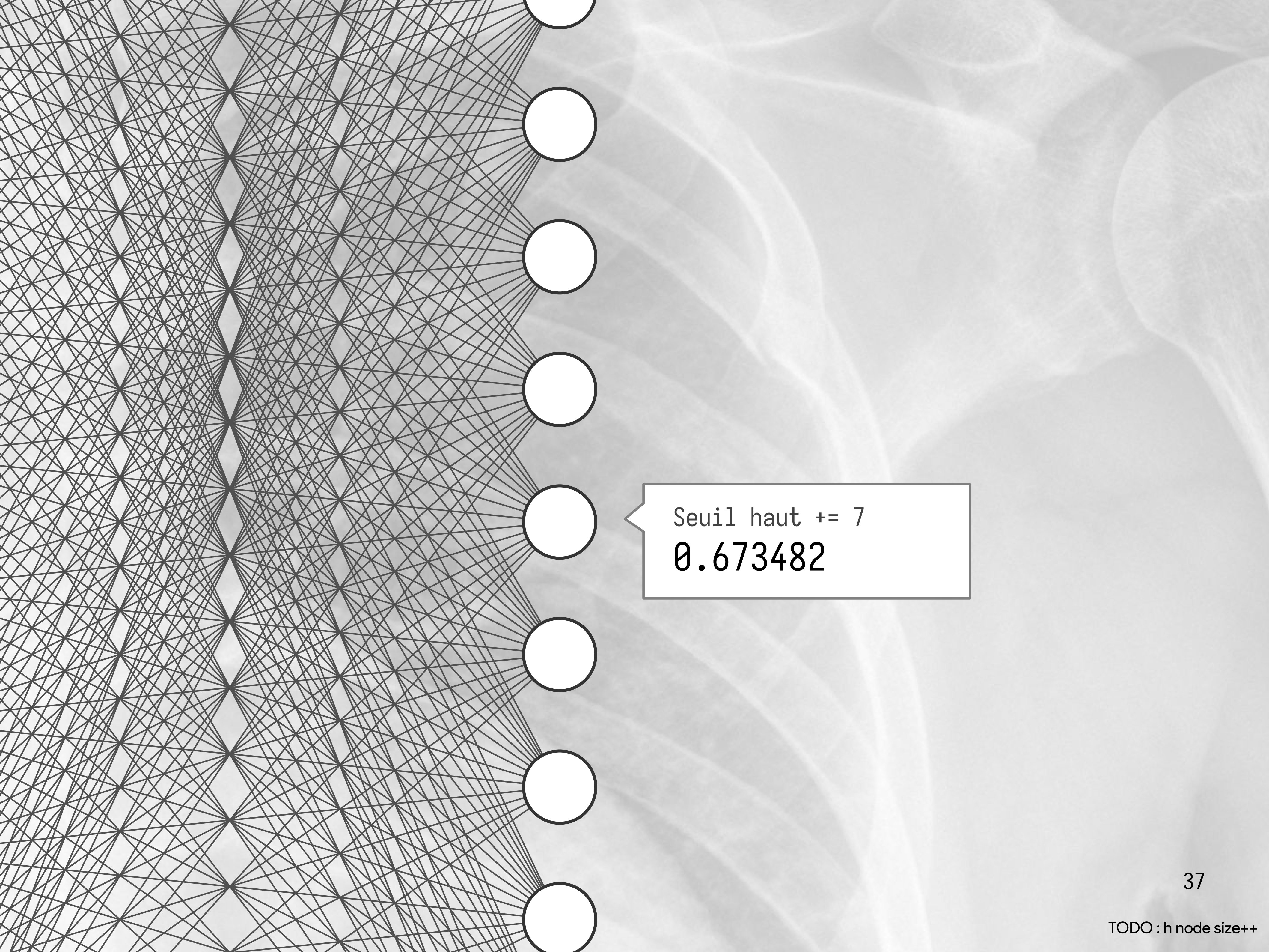
Incréments de la luminosité

$$\mathrm{lum} \in \{-5, -4, \ldots, +3, +4, +5\}$$

Réglage du rayon de flou

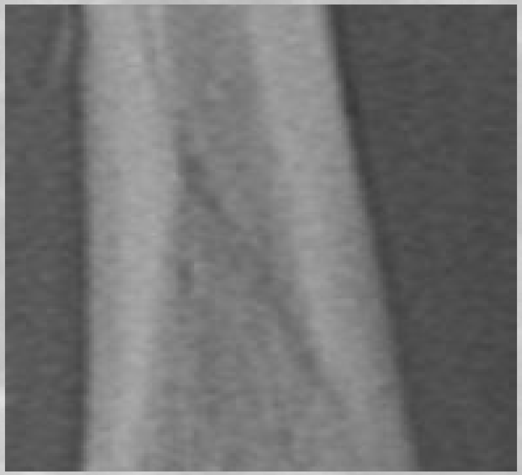
$$r_{\mathrm{flou}} \in \{0, 10, \ldots, 50\}$$

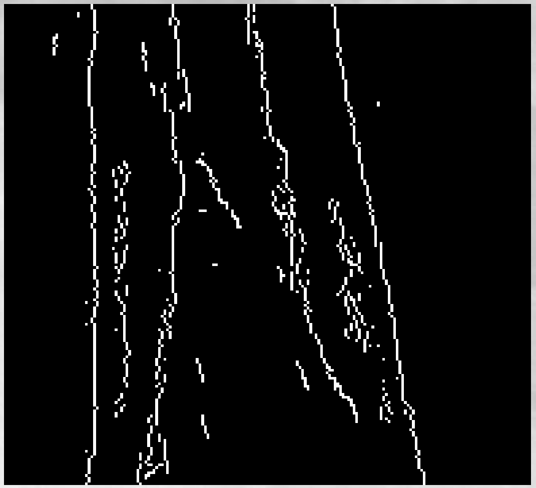
TODO : ajouter des + pour les incréments partout

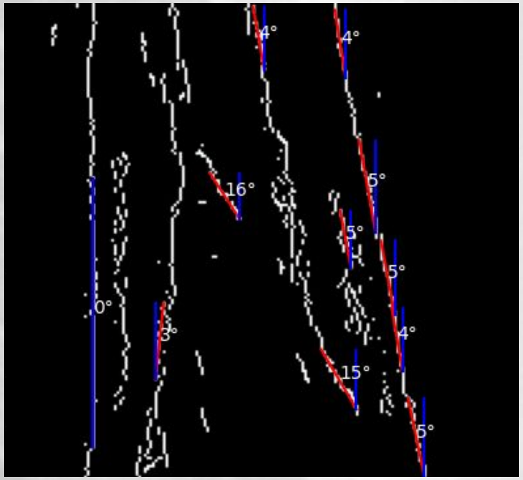


TODO : h node size++

Résultats







Résultats