Candidat #12184

Détection de fractures osseuses

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

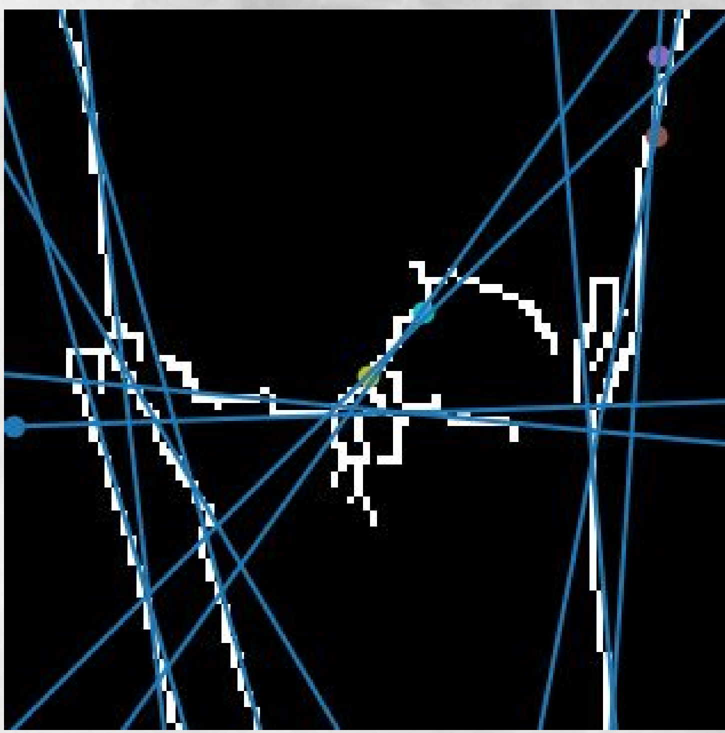
Détection des bords
Avec l'algorithme Canny

Bord fort
≥ seuil haut

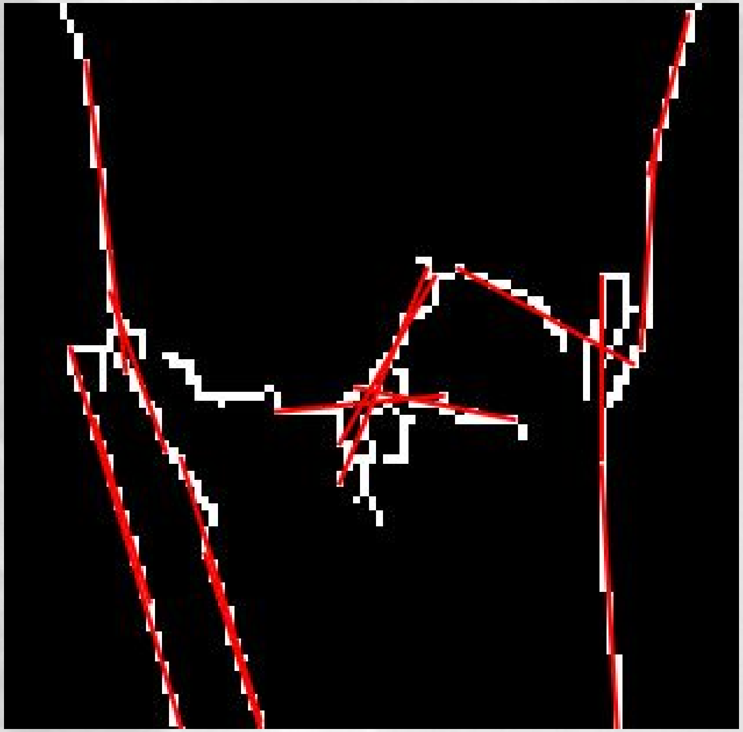
Bord faible
< seuil haut

Détection des bords
Avec l'algorithme Canny

Détection des traits
Avec la Transformée de Hough



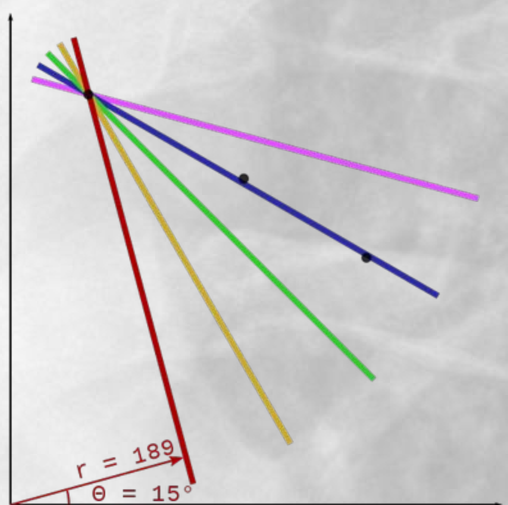
Classique
(détecte des droites)



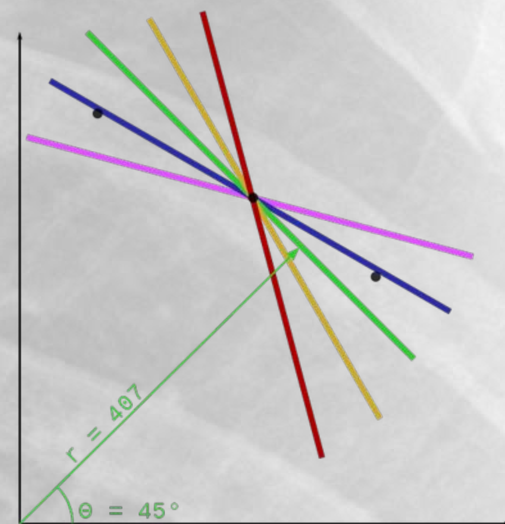
Probabiliste
(détecte des segments)

Détection des traits

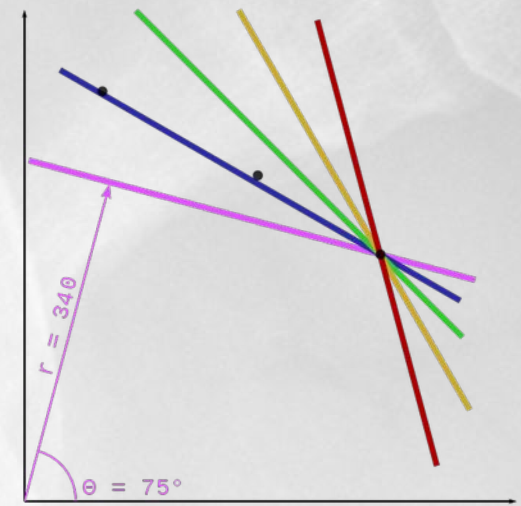
Avec la Transformée de Hough



θ	r
15	189.0
30	282.0
45	355.7
60	407.3
75	429.4



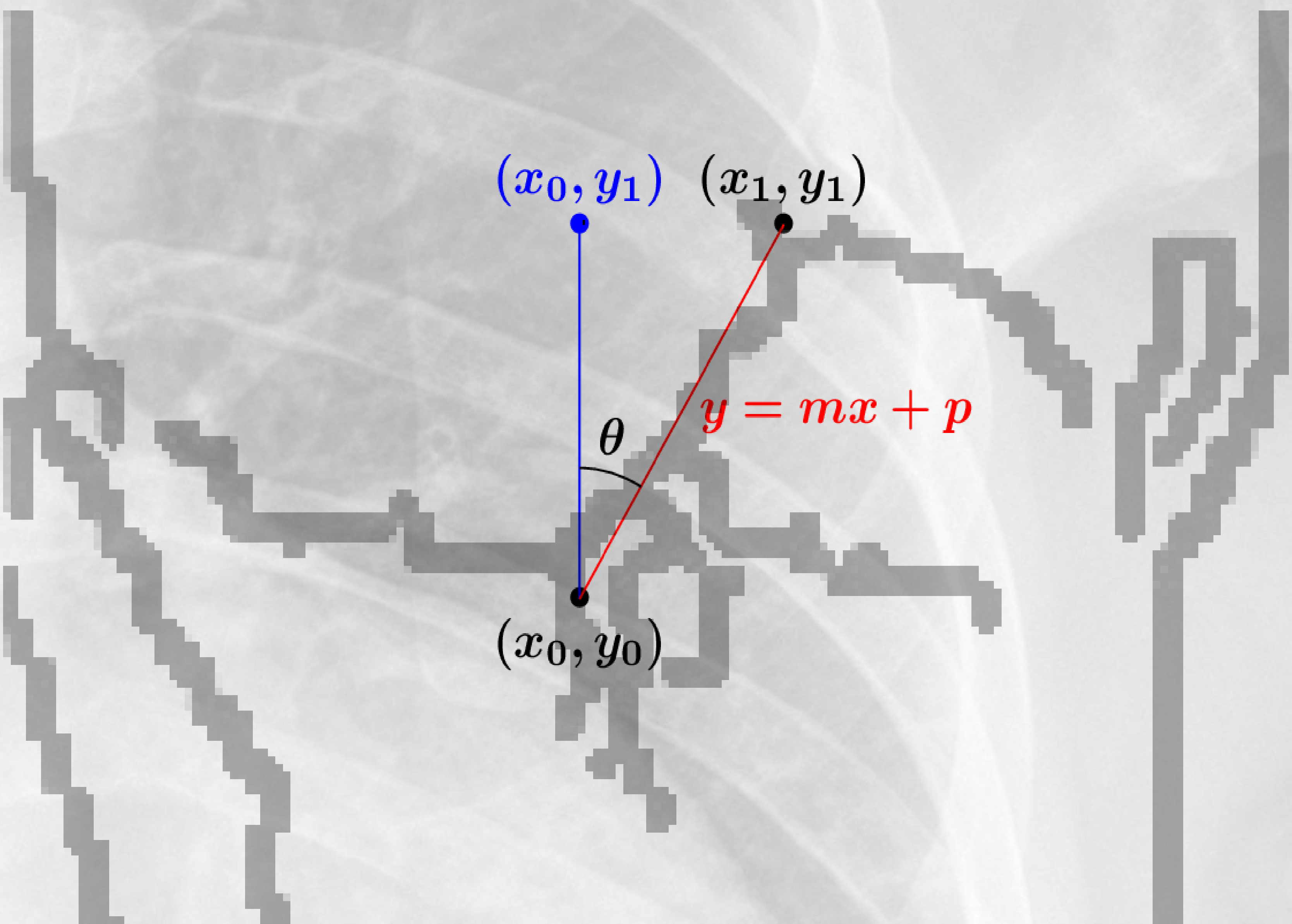
θ	r
15	318.5
30	376.8
45	407.3
60	409.8
75	385.3



θ	r
15	419.0
30	443.6
45	438.4
60	402.9
75	340.1

Calcul des angles

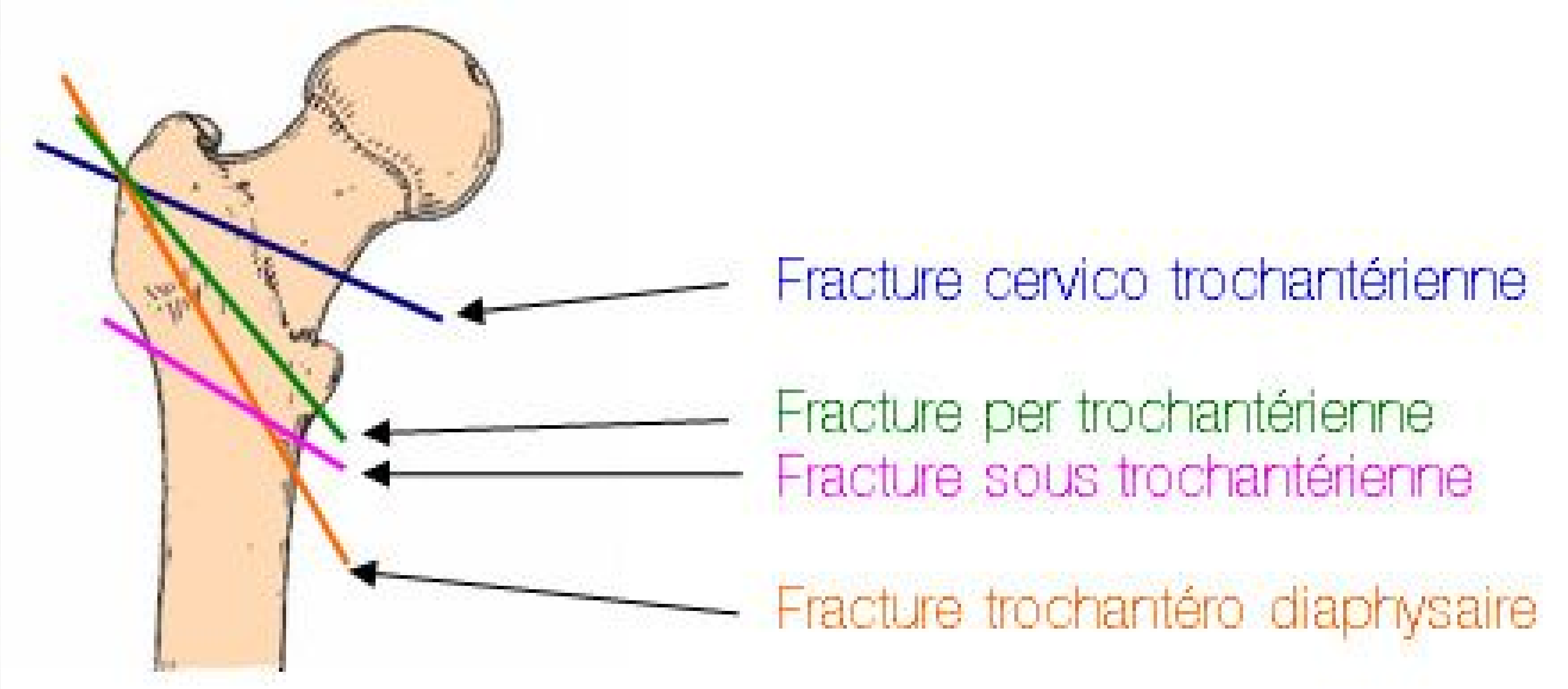
Avec de la trigonométrie



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

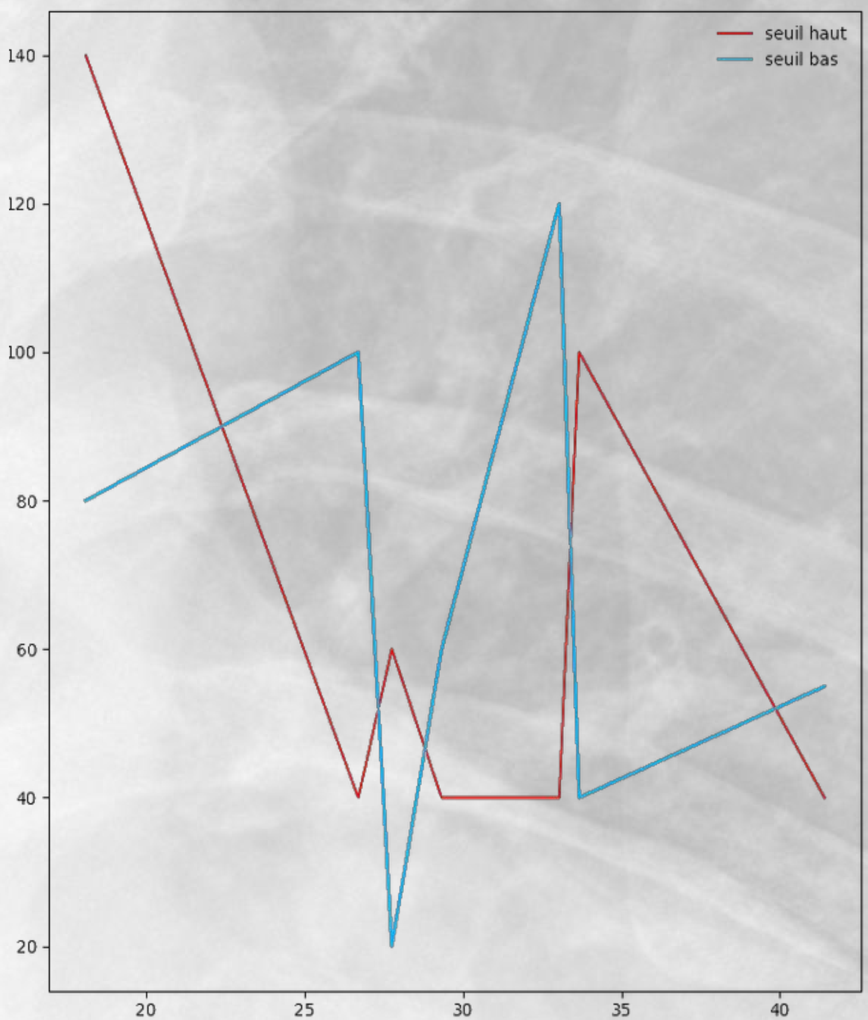
cv2.Canny

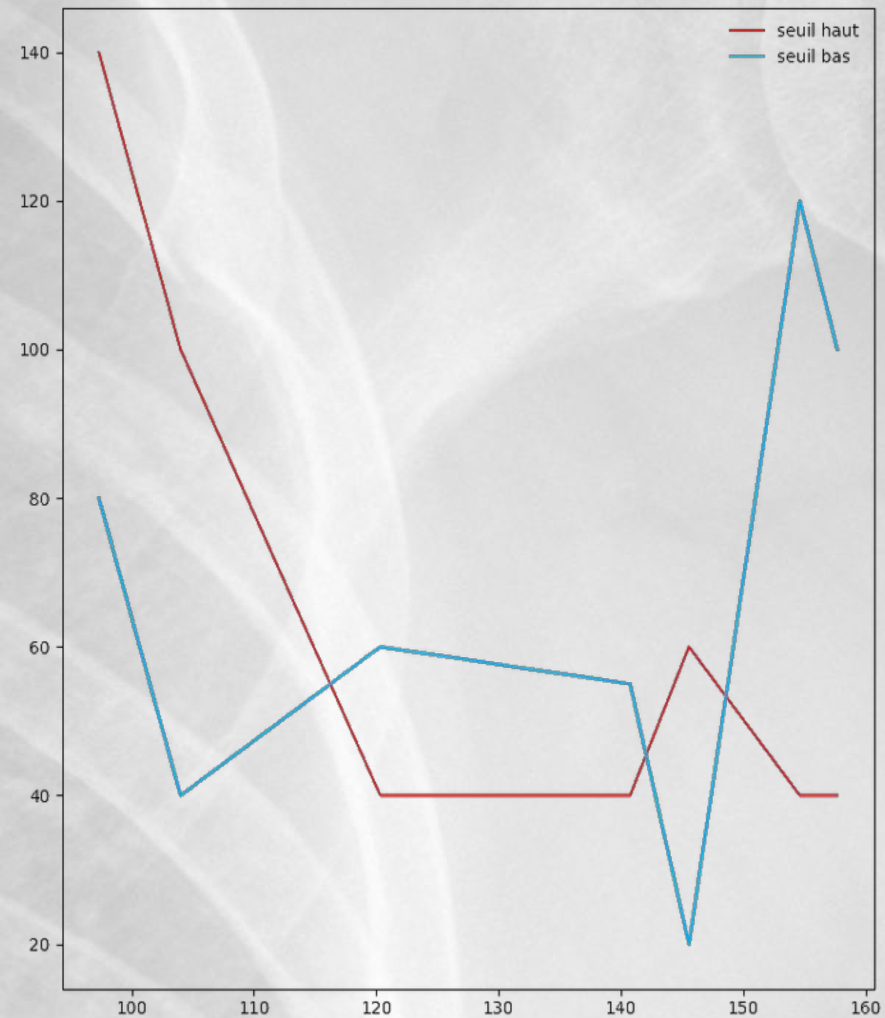
bas: 40
haut: 60

bas: 40
haut: 120

bas: 60
haut: 180

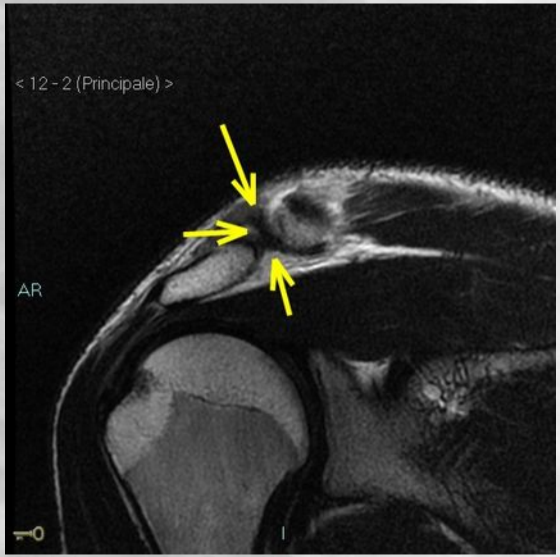
seuils(luminosité, contraste) ?



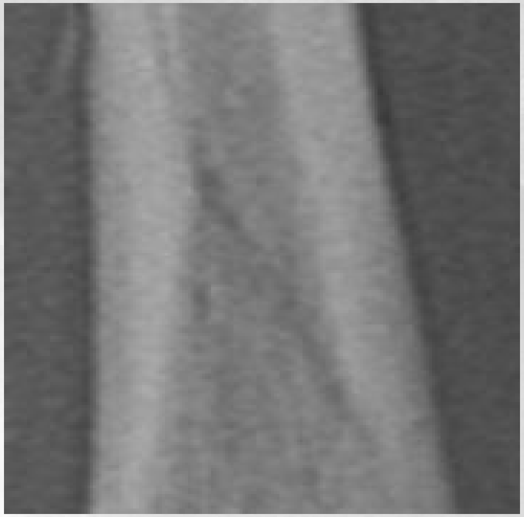


Seuils optimaux de détection de bords

Recherche de sets de données



Recherche de sets de données

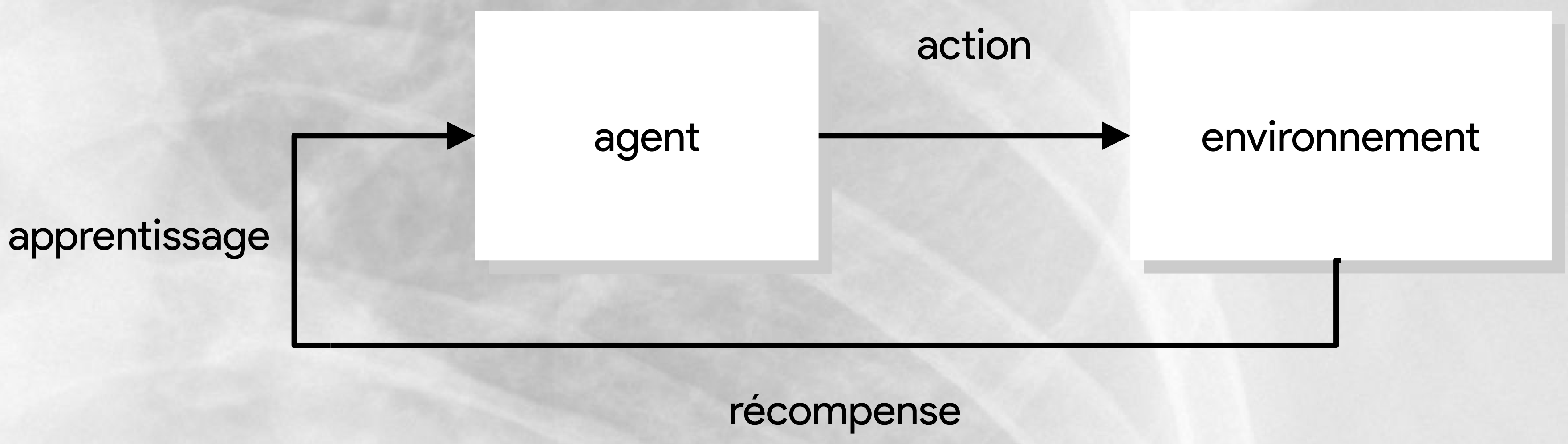


bas 40
haut 120

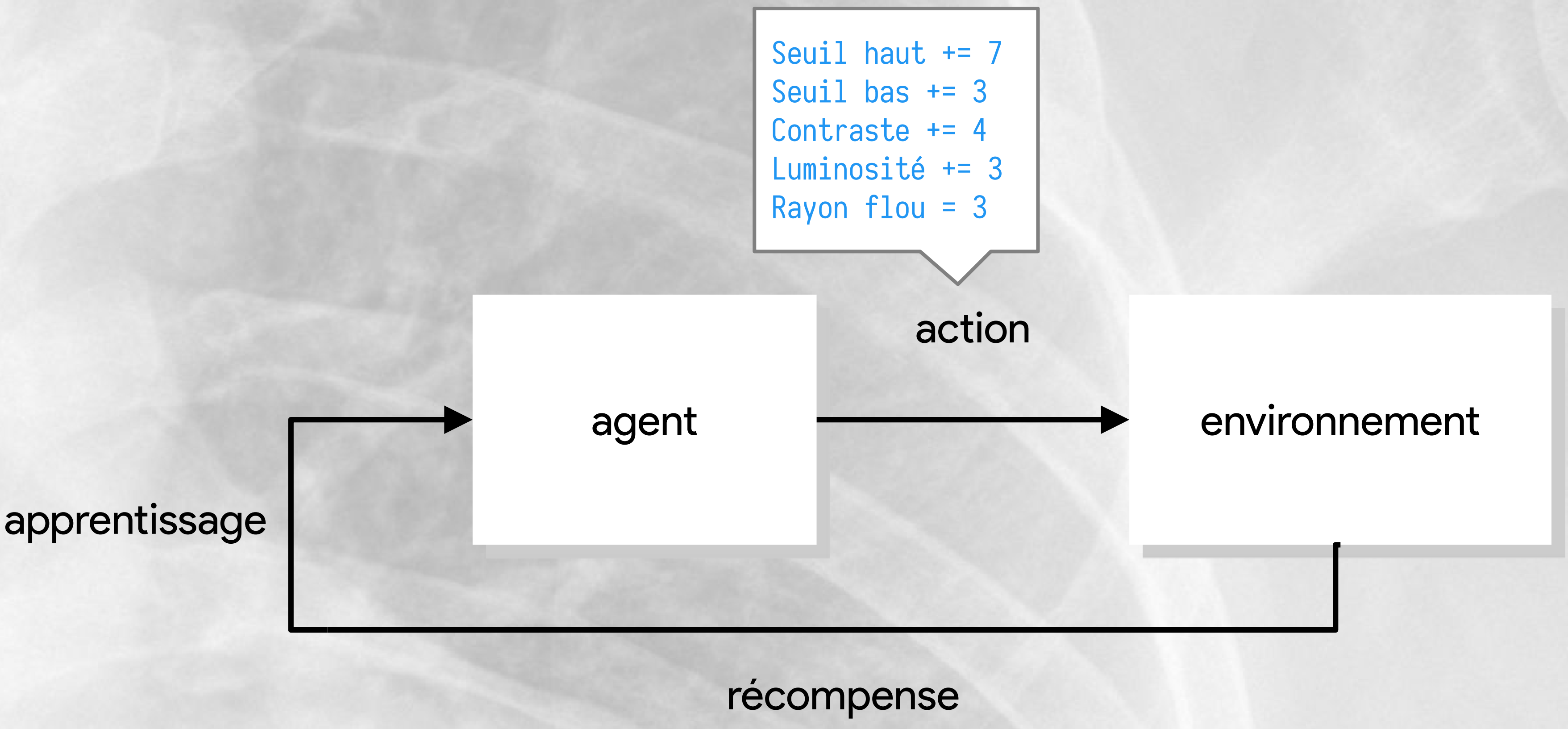
bas 27
haut 44

bas 20
haut 21

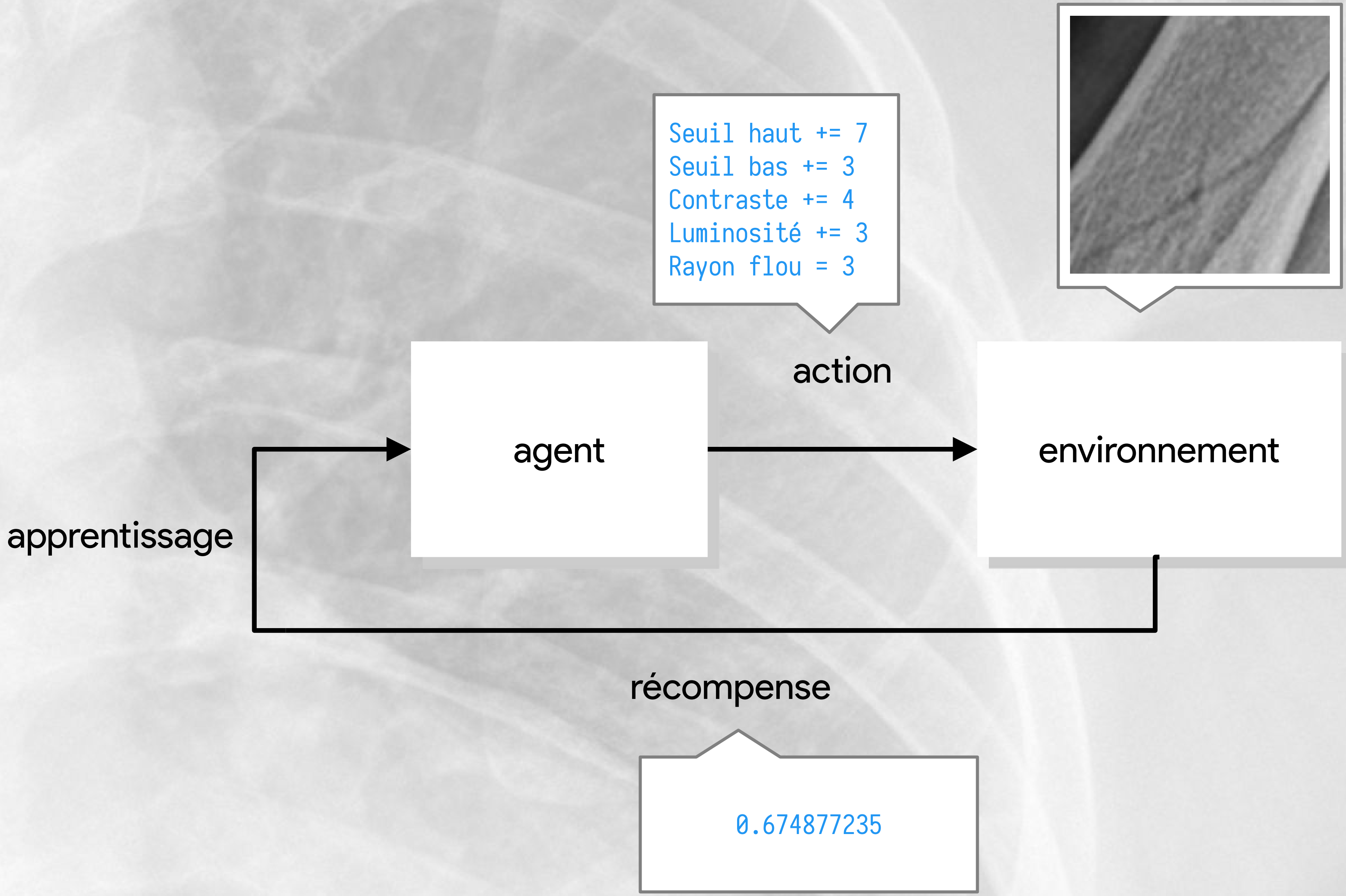
Apprentissage par renforcement



Apprentissage par renforcement



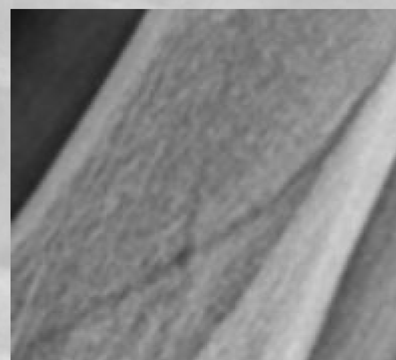
Apprentissage par renforcement



Apprentissage par renforcement

action

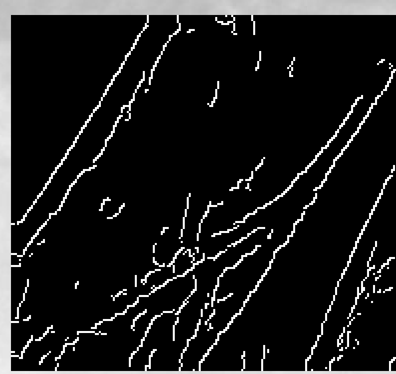
$\mathrm{cont}, \mathrm{lum}, r_{\mathrm{flou}}, \mathrm{seuil}_h, \mathrm{seuil}_b$



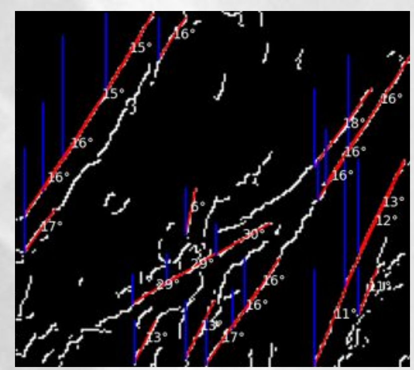
S

Environnement

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

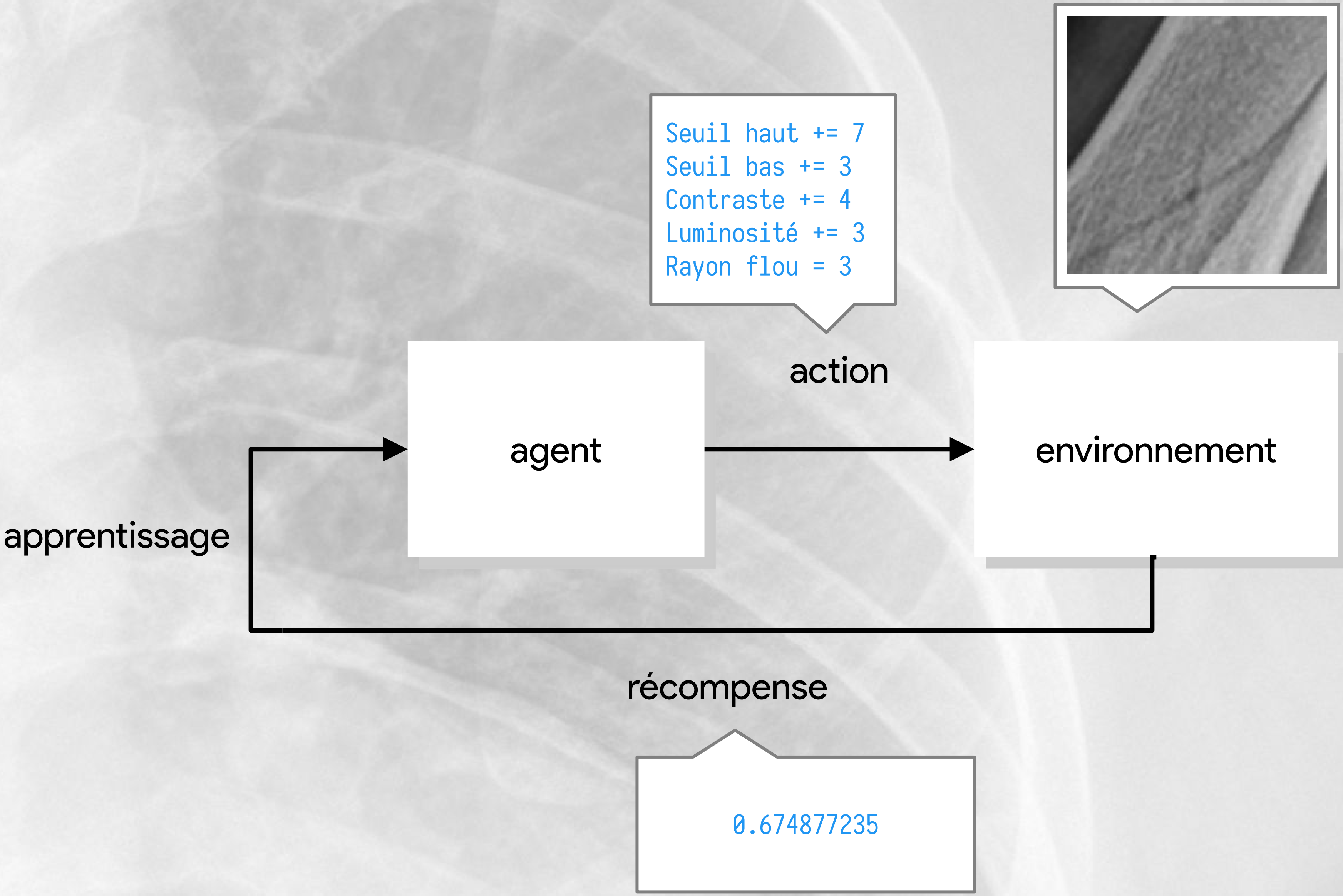


$$B = \underset{\mathrm{seuil}_h, \mathrm{seuil}_b}{\mathrm{Canny}} \; T$$



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

Apprentissage par renforcement



Calcul de la récompense

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

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

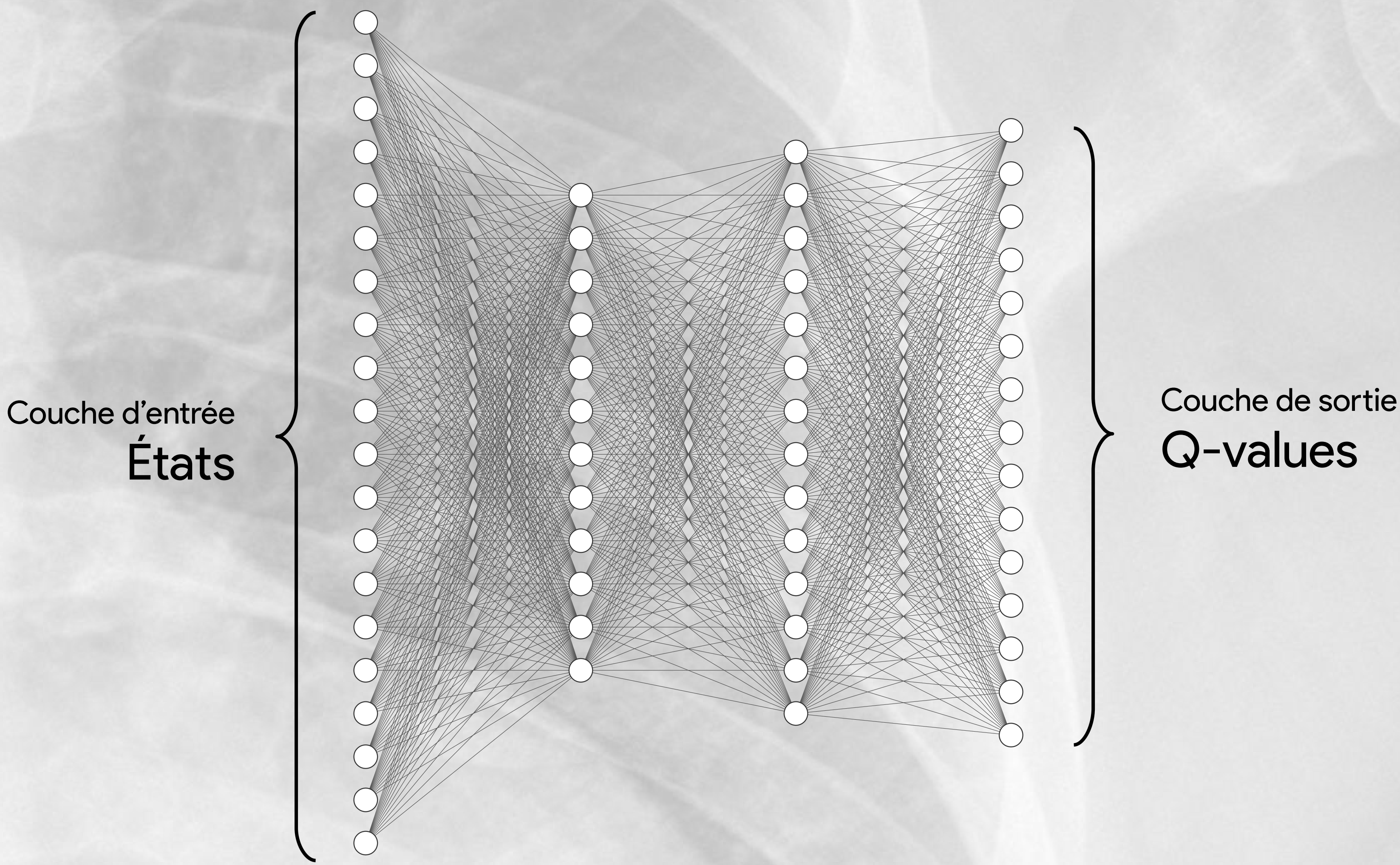
Le problème de dimension des *Q-Tables*

$$8^{200\cdot 200} \cdot 20 \cdot 20 \cdot 6 \cdot 10 \cdot 5 \approx$$

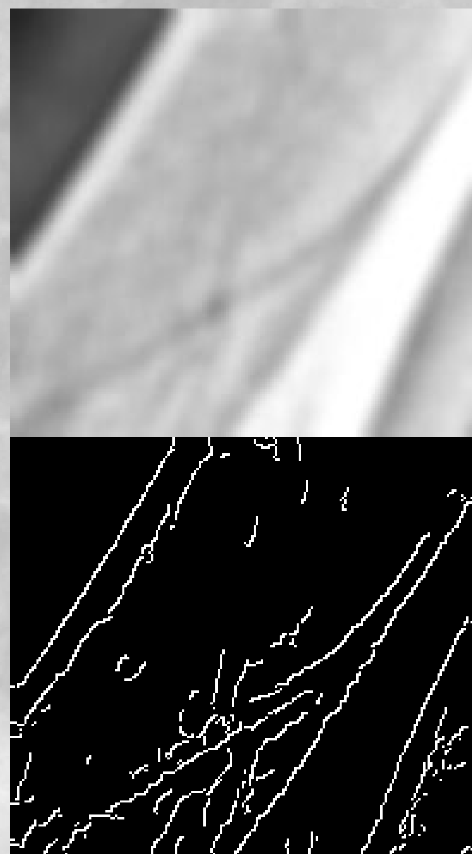
$$4.77 \cdot 10^{36128}$$

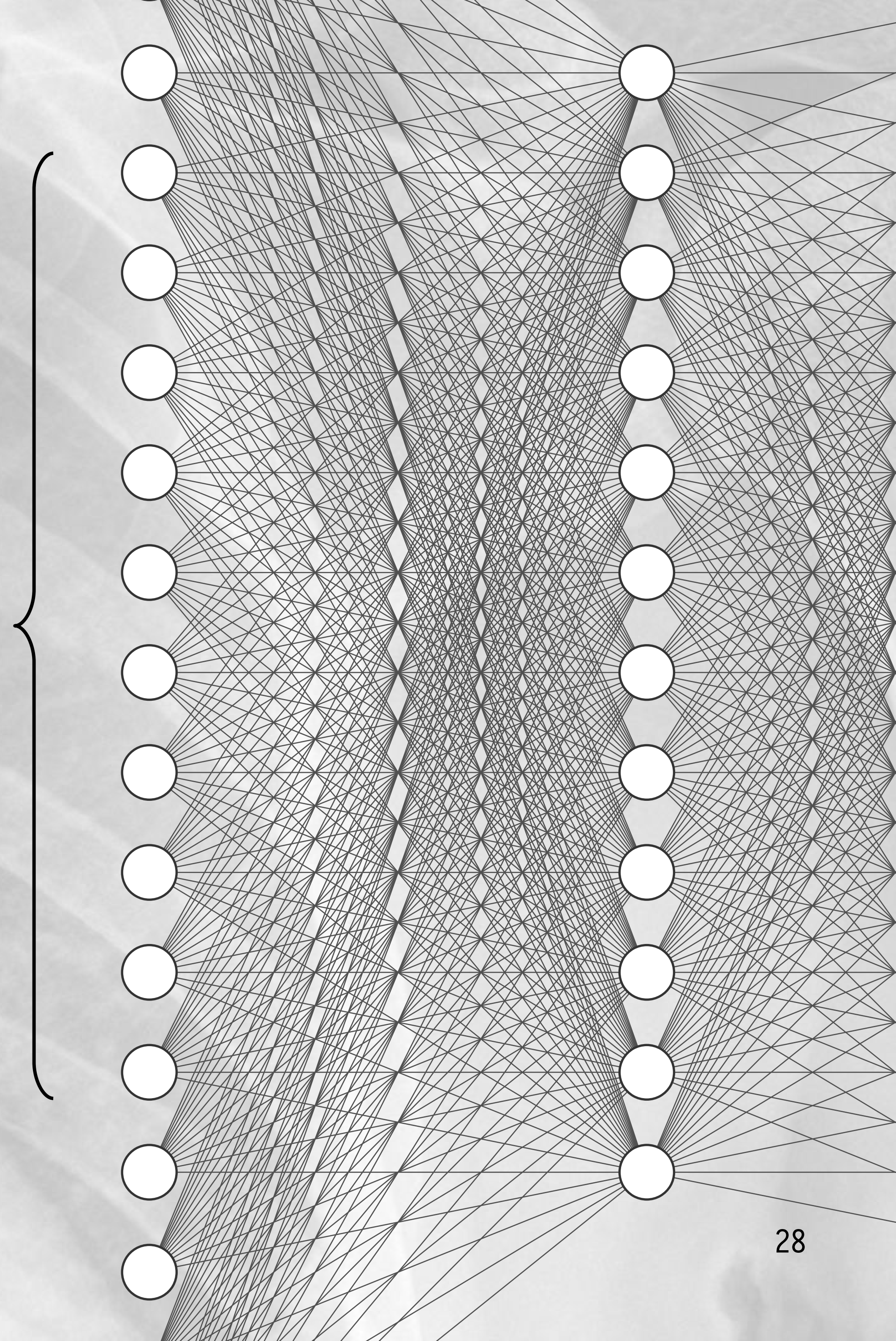
TODO : taille en pétaoctets

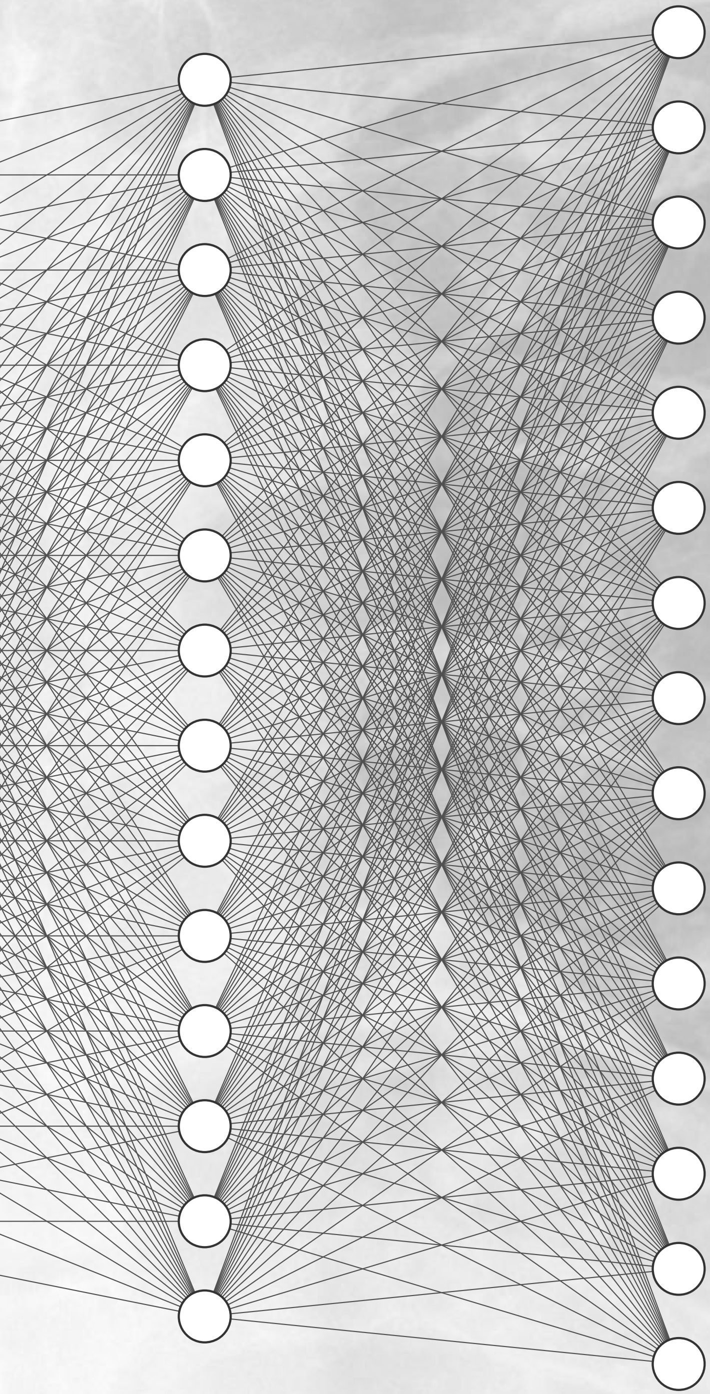
Apprentissage de l'agent
Avec des *Deep-Q Networks*



Un état







Incréments du seuil haut
$\text{seuil}_h \in \{-10, -9, \ldots, 8, 9, 10\}$

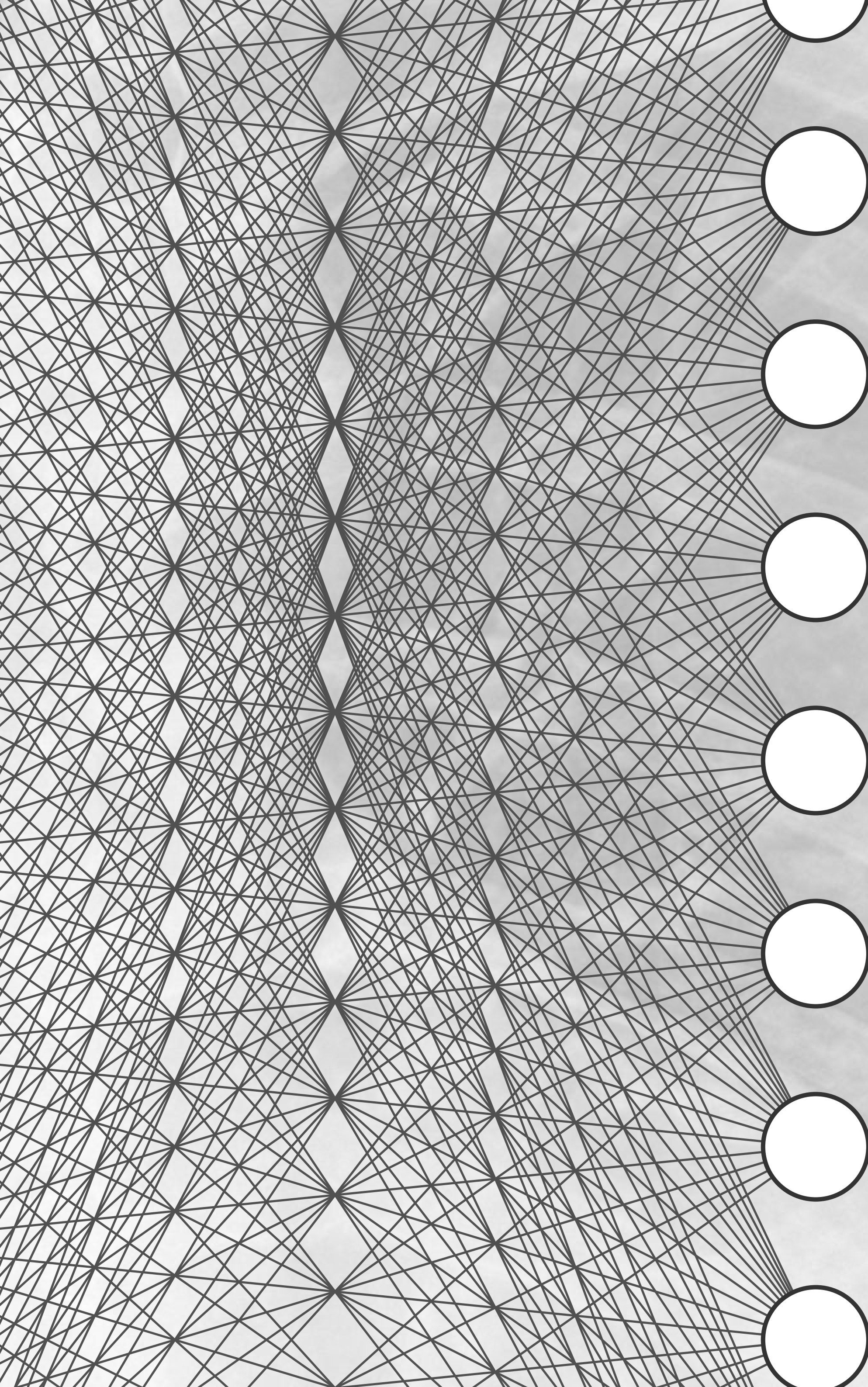
Incréments du seuil bas
$\text{seuil}_b \in \{-10, -9, \ldots, 8, 9, 10\}$

Incréments du contraste
$\text{cont} \in \{1, 1.1, 1.2, 1.3, 1.4, 1.5\}$

Incréments de la luminosité
$\text{lum} \in \{-5, -4, \ldots, +3, +4, +5\}$

Réglage du rayon de flou
$r_{\text{flou}} \in \{0, 10, \ldots, 50\}$

TODO : ajouter des + pour les incréments partout

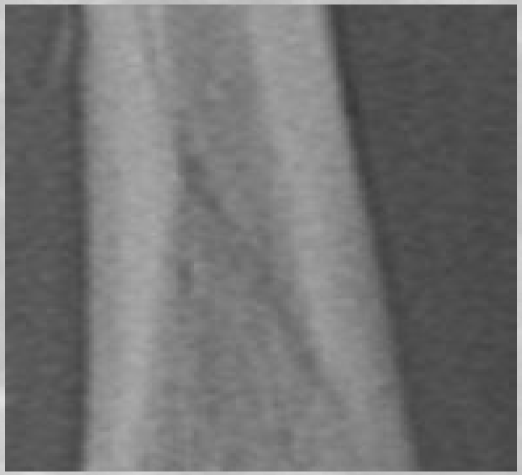


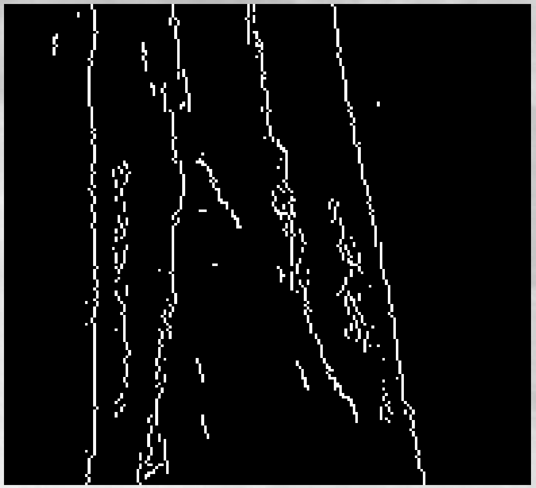
Seuil haut += 7

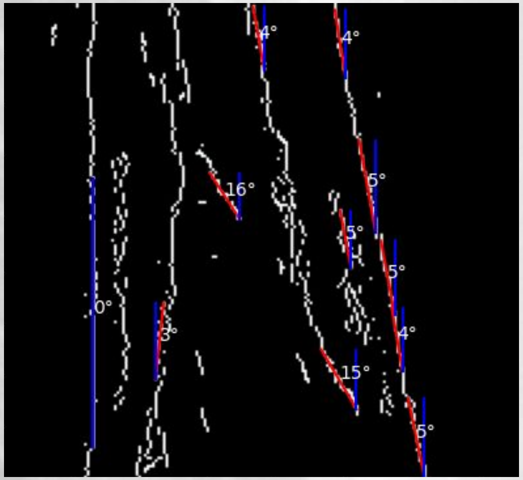
0.673482

TODO : h node size++

Résultats







Résultats