

# Détection de fractures osseuses

# Recherche de sets de données

Format DICOM

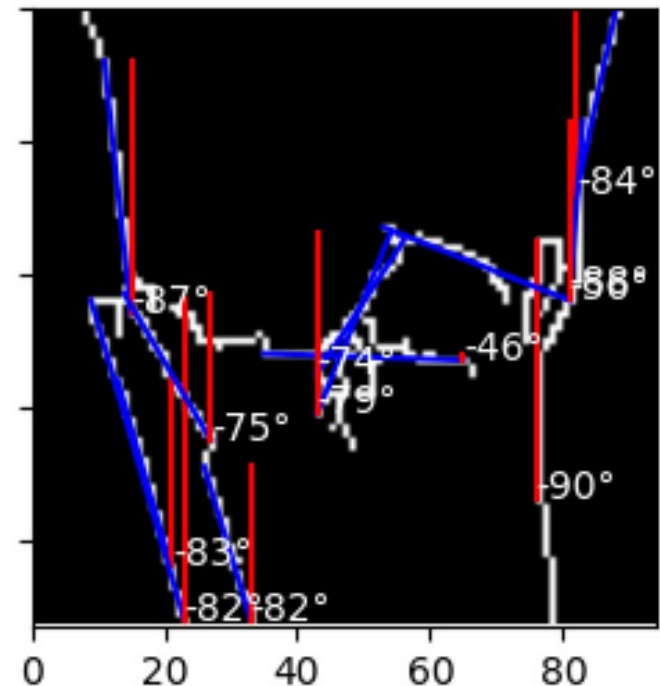
# Recherche de sets de données

## Images normales



# Stratégie sans *machine learning*

- Recadrage automatique
- Détection des bords
- Obtention de segments
  - Vectorisation
  - Détection de segments
- Calcul des angles



# Détection des bords

# Détection des bords

Un problème de texture



cv2.Canny



bas : 60  
haut : 40



bas : 40  
haut : 120



bas : 60  
haut : 180

# **Détection des bords**

## Recherche des seuils optimaux

seuils(image)



seuils(luminosité, contraste) ?

seuils(

luminosité = moyenne,

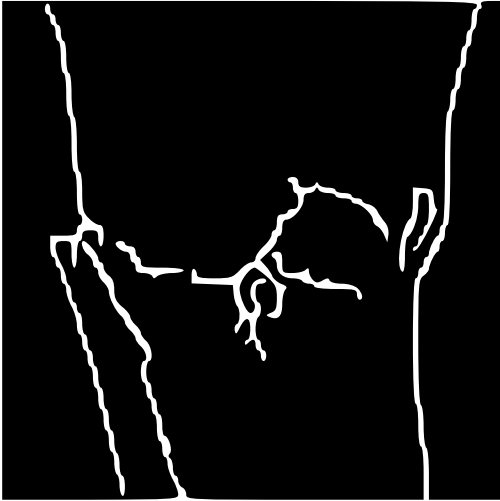
contraste =  $\sigma$ ,

) ?

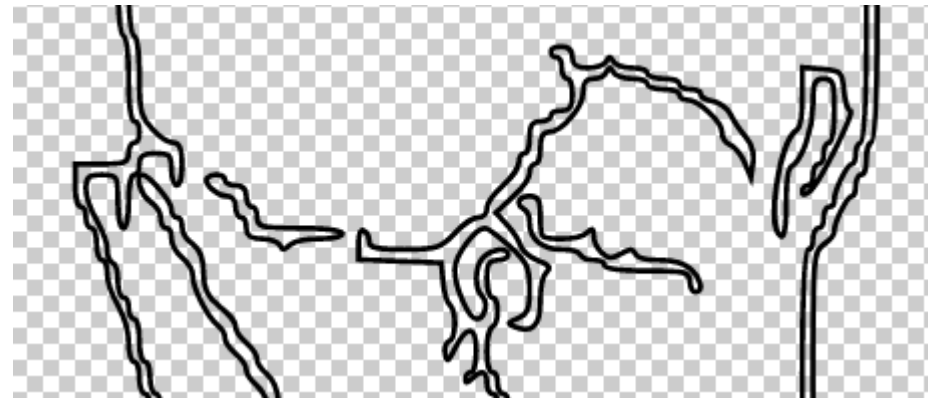
# Vectorisation

# Vectorisation

## *Avec potrace*

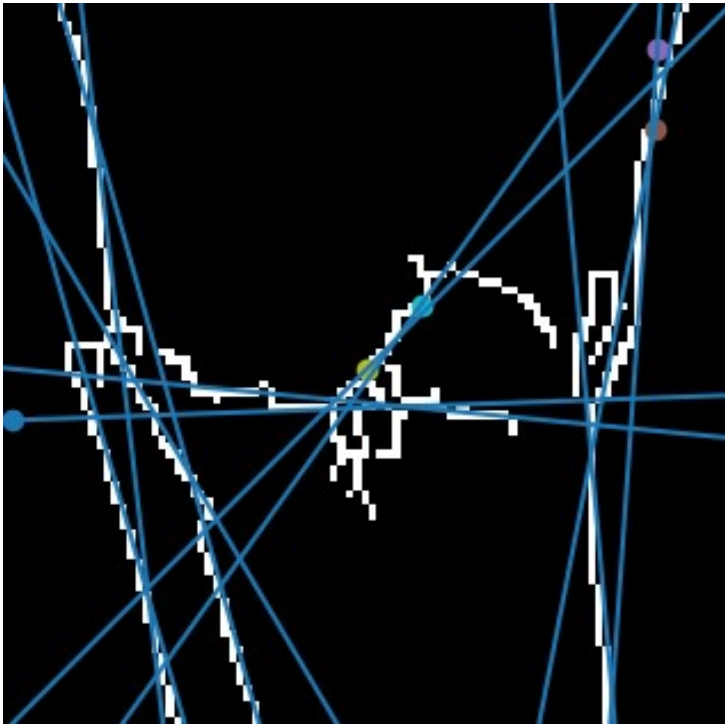


```
<g  
  transform="..."  
  fill="#000000transparent"  
  stroke="black"  
  stroke-width="10px"  
>
```

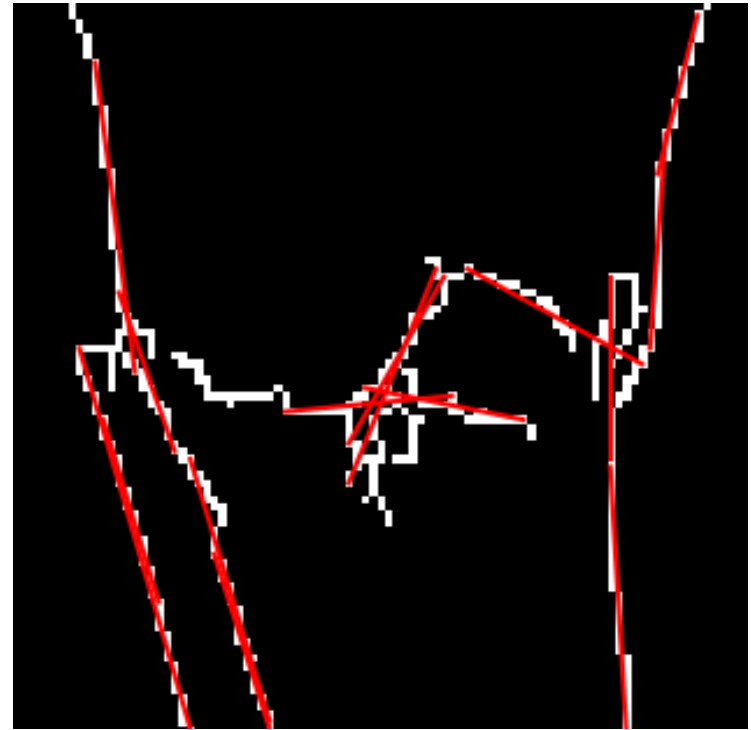


# Détection des traits

## Avec la Transformée de Hough



Classique  
(détection des droites)



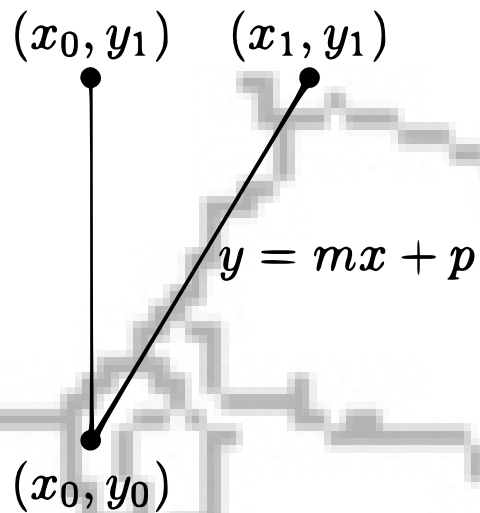
Probabiliste  
(détection des segments)

Calcul des angles

# Calcul des angles

## Avec de la trigonométrie

$$\theta = \arccos \frac{\text{adjacent}}{\text{hypothénuse}} = \arccos \frac{|y_1 - y_0|}{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}}$$



Critère de décision

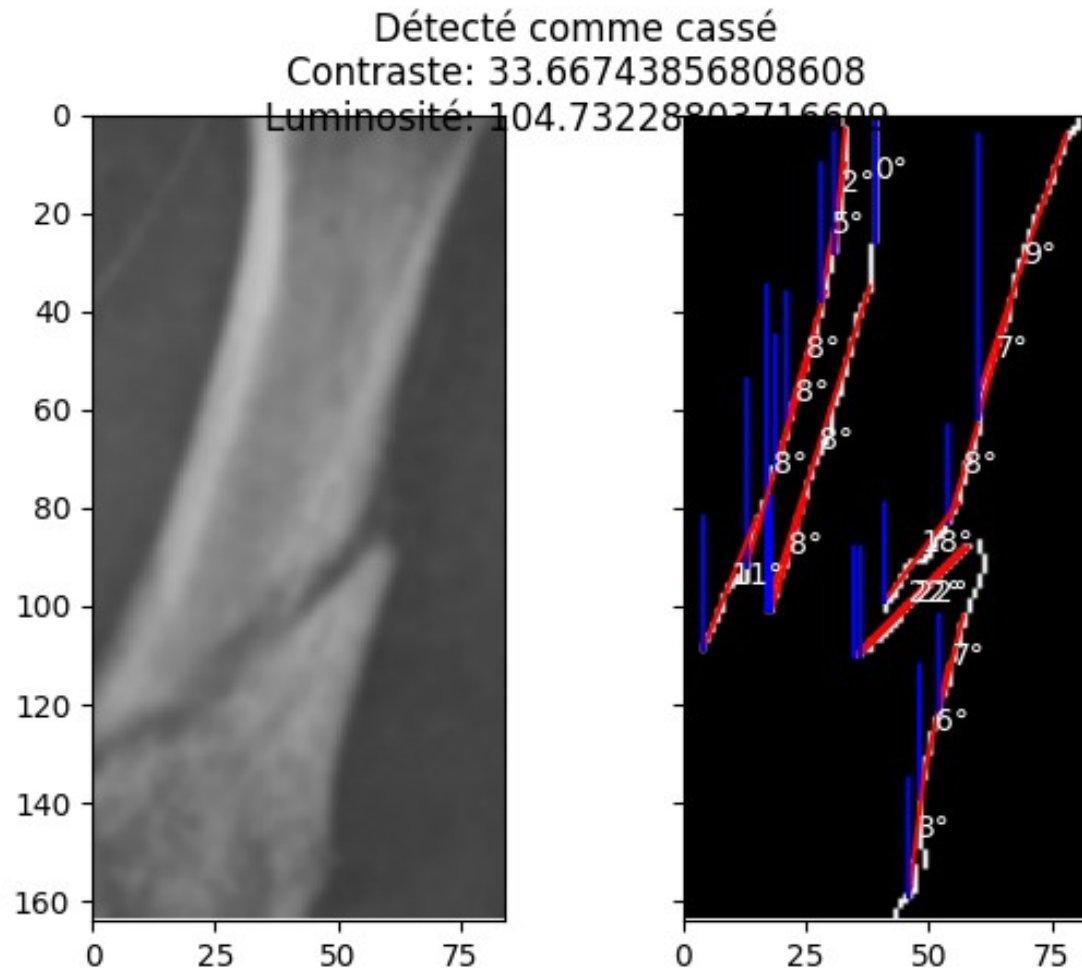


# Critère de décision

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

# Compensations

# Compensation De l'inclinaison



# Machine Learning

[github.com/ewen-lbh/bone-fracture-detection](https://github.com/ewen-lbh/bone-fracture-detection)