

TP 4 : Traitement de l'Image

Rendu

Exercice 1 :

Formule utilisé pour transformer l'image ppm en image pgm : $0,3 \cdot cR + 0,6 \cdot cG + 0,1 \cdot cB$.

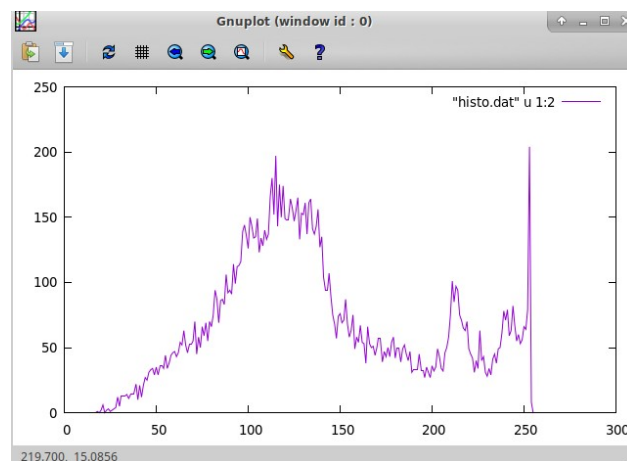


Image Original



Image en niveau de gris

Source : <https://lesvoyageusesduquebec.com/adopter-un-chien-en-voyage-ramener-un-chien-au-canada/>



Histogramme de l'image en niveau de gris

Exercice 2 :

Seuil utilisé : 120.



Image binaire

```
for (int i=0; i < nH; i++)  
    for (int j=0; j < nW; j++)  
    {  
        if ( ImgIn[i*nW+j] < S) ImgOut[i*nW+j]=0; else ImgOut[i*nW+j]=255;  
    }
```

Seuil niveau de gris

Exercice 3 :

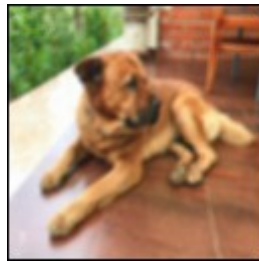


Image flouté

```
for (int i=1; i < nH-1; i++){  
    for(int j =1; j < nW -1; j++){  
        position = i*nW*3 + j*3;  
        posR = position;  
        posG = position+1;  
        posB = position+2;  
        ImgOut[posR] = (ImgIn[posR] + ImgIn[posR+nW*3] + ImgIn[posR-nW*3] + ImgIn[posR+nW*3+3] + ImgIn[posR+nW*3-3] + ImgIn[posR-nW*3+3] + ImgIn[posR-nW*3-3] + ImgIn[posR + 3] + ImgIn[posR - 3])/9;  
        ImgOut[posG] = (ImgIn[posG] + ImgIn[posG+nW*3] + ImgIn[posG-nW*3] + ImgIn[posG+nW*3+3] + ImgIn[posG+nW*3-3] + ImgIn[posG-nW*3+3] + ImgIn[posG-nW*3-3] + ImgIn[posG + 3] + ImgIn[posG - 3])/9;  
        ImgOut[posB] = (ImgIn[posB] + ImgIn[posB+nW*3] + ImgIn[posB-nW*3] + ImgIn[posB+nW*3+3] + ImgIn[posB+nW*3-3] + ImgIn[posB-nW*3+3] + ImgIn[posB-nW*3-3] + ImgIn[posB + 3] + ImgIn[posB - 3])/9;  
    }  
}
```

Floutage

Exercice 4 :



Fond flou

```

for (int i=1; i < nH-1; i++){
  for (int j =1; j < nW -1; j++){
    position = i*nW*3 + j*3;
    posR = position;
    posG = position+1;
    posB = position+2;
    if (ImgIn2[1*nW*3] == 255){
      ImgOut[posR] = (ImgIn[posR] + ImgIn[posR+nW*3] + ImgIn[posR-nW*3] + ImgIn[posR+nW*3*3] + ImgIn[posR-nW*3*3] + ImgIn[posR-nW*3+3] + ImgIn[posR - 3])/9;
      ImgOut[posG] = (ImgIn[posG] + ImgIn[posG+nW*3] + ImgIn[posG-nW*3] + ImgIn[posG-nW*3*3] + ImgIn[posG+nW*3*3] + ImgIn[posG-nW*3+3] + ImgIn[posG - 3])/9;
      ImgOut[posB] = (ImgIn[posB] + ImgIn[posB+nW*3] + ImgIn[posB-nW*3] + ImgIn[posB-nW*3*3] + ImgIn[posB+nW*3*3] + ImgIn[posB+nW*3+3] + ImgIn[posB - 3])/9;
    }
    else{
      ImgOut[posR] = ImgIn[posR];
      ImgOut[posG] = ImgIn[posG];
      ImgOut[posB] = ImgIn[posB];
    }
  }
}

```

Floutage Fond

Exercice 5 :



Erosion + Dilatation

```
for (int i=1; i < nH-1; i++)
for (int j=1; j < nW-1; j++)
{
    if ( ImgIn[i*nW+j] == 0){
        if(ImgIn[(i-1)*nW+(j+1)] == 255 || ImgIn[(i-1)*nW+(j-1)] == 255 || ImgIn[(i+1)*nW+(j+1)] == 255 || ImgIn[(i+1)*nW+(j-1)] == 255 ||
        ImgOut[i*nW+j] = 255;
    }
}
}
```

Erosion

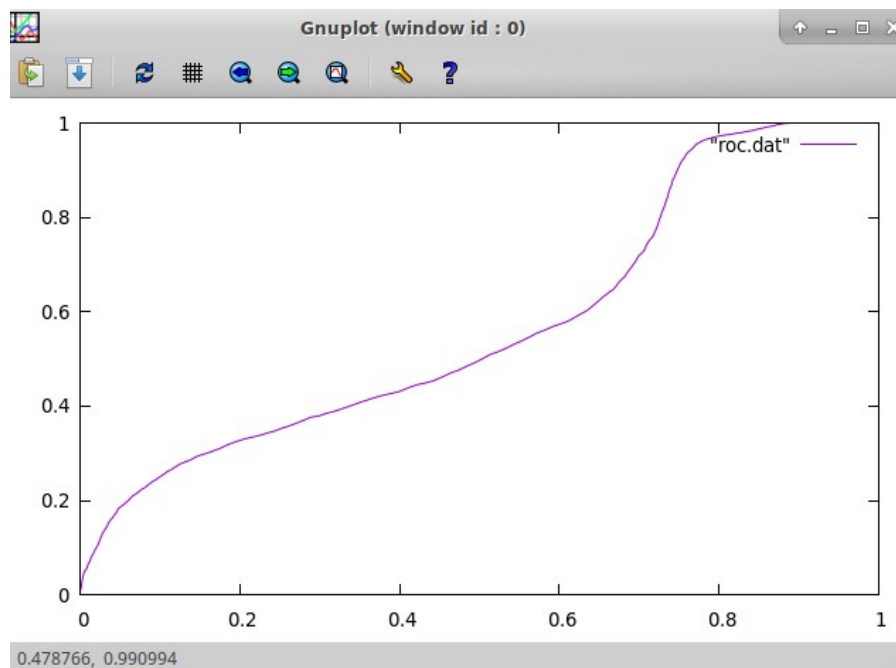
```
for (int i=1; i < nW-1; i++)  
    for (int j=1; j < nW-1; j++)  
        {  
            if ( ImgIn[i*nW+j] == 255){  
                if(ImgIn[i*nW+(j+1)] == 0 || ImgIn[(i-1)*nW+(j+1)] == 0 || ImgIn[(i-1)*nW+(j-1)] == 0 || ImgIn[(i+1)*nW+(j+1)] == 0 || ImgIn[(i+1)*nW+(j-1)] == 0 || ImgIn[(i+1)*nW+j] == 0 || Img  
                    ImgOut[i*nW+j] = 0;  
            }  
        }
```

Dilatation

Exercice 6 :



Carte de vérité



Courbe Roc

L'image choisit n'étant pas idéal, on peut voir que l'aspect de la courbe n'est pas idéal non plus.

47	89	227	4019	12049	0.00733234	0.946538
48	107	242	4007	12028	0.00881747	0.943045
49	116	267	3987	12014	0.00956307	0.937236
50	127	285	3972	12000	0.0104725	0.933051
51	148	299	3961	11976	0.0122072	0.929812
52	157	319	3944	11964	0.0129527	0.92517
53	169	343	3927	11945	0.0139508	0.919672
54	188	360	3912	11924	0.0155218	0.91573
55	205	377	3898	11904	0.0169296	0.911813
56	222	404	3878	11880	0.0183441	0.905652
57	239	421	3866	11858	0.019757	0.901796
58	259	439	3849	11837	0.021412	0.897621
59	278	464	3826	11816	0.0229866	0.891841
60	296	492	3804	11792	0.0244871	0.885475
61	313	522	3778	11771	0.025902	0.878605
62	331	547	3758	11748	0.0274029	0.872938
63	350	574	3734	11726	0.0289831	0.86676
64	379	599	3715	11691	0.0314002	0.86115
65	405	625	3695	11659	0.033571	0.855324
66	432	661	3663	11628	0.0358209	0.847132
67	458	687	3642	11597	0.0379925	0.841303
68	487	704	3628	11565	0.0404082	0.837488
69	513	731	3605	11535	0.0425797	0.831411
70	545	752	3587	11500	0.045247	0.826688
71	567	785	3561	11471	0.0471008	0.819374
72	610	812	3538	11424	0.0506897	0.813333
73	642	825	3528	11389	0.0533621	0.810476
74	680	845	3510	11349	0.0565301	0.80597
75	713	862	3495	11314	0.0592833	0.802157
76	752	889	3470	11273	0.0625364	0.796054
77	787	913	3452	11232	0.0654797	0.790836
78	836	933	3435	11180	0.0695739	0.786401
79	873	951	3425	11135	0.0727015	0.782678
80	918	976	3404	11086	0.0764745	0.777169
81	967	993	3388	11036	0.0805632	0.773339
82	1017	1018	3366	10983	0.08475	0.767792
83	1079	1050	3341	10914	0.0899691	0.760875

Valeur

- Première colonne : Valeur du seuil.
- Deuxième colonne : Vrai positif.
- Troisième colonne : Faux positif.
- Quatrième colonne : Vrai négatif.
- Cinquième colonne : Faux négatif.
- Sixième colonne : Sensibilité.
- Septième colonne : Spécificité.