

TD – Algorithme KNN - Identification automatique de panneaux

ELEMENTS DE CORRECTION :

```

## Mise en place de l'algorithme KNN

def Distance_uv(u,v):
    n = len(u)
    Dst = 0
    for i in range(n):
        di = u[i]-v[i]
        Dst += di**2
    Dst = Dst**(1/2)
    return Dst

def Distance(u,Lv):
    Ld = []
    for i in range(len(Lv)):
        v = Lv[i]
        Dst = Distance_uv(v,u)
        Res = [Dst,i]
        Ld.append(Res)
    return Ld

def Proches(u,Lv,k):
    Ld = Distance(u,Lv)
    Ld.sort()
    Res = Ld[:k]
    return Res

## Lecture des images

# Affichage

import matplotlib.pyplot as plt
plt.close('all')

def Affiche(image):
    plt.figure()
    plt.imshow(image)
    plt.axis('off')
    plt.show()
    plt.pause(0.00001)

# Lecture des images

def Lecture(Chemin):
    Image = plt.imread(Chemin)
    return Image

## Fonctions d'analyse des images

def Analyse(Image):
    Nl,Nc = Image.shape[0:2]
    N = Nl * Nc
    L_RGB = []
    for l in range(Nl):
        for c in range(Nc):
            R,G,B,H = Image[l,c]
            R = float(R)
            G = float(G)
            B = float(B)
            L_RGB += [R,G,B]
    return L_RGB

def Analyse_Globale(L_Chemin):
    Res = []
    N = len(L_Chemin)
    for i in range(len(L_Chemin)):
        Chemin = L_Chemin[i]
        print("Apprentissage image",i+1,"sur",N)
        Image = Lecture(Chemin)
        Analyse_im = Analyse(Image)
        Res.append(Analyse_im)
    return Res

```

```

## Création de la base des données

# Ouverture des images sources

Sources = "Sources\\"
Dossiers = [0,1,2,3,4,5,6,7]
Nb_Dossiers=len(Dossiers)
Nb_Images_Dossiers = [5,5,5,5,5,5,5]

Liste_Chemin = []
Liste_Dossier = []
Liste_Num = []
for d in Dossiers:
    Nb_Im = Nb_Images_Dossiers[d]
    for im in range(Nb_Im):
        Chemin = Sources + str(d) + "\\" + str(im) + ".png"
        Liste_Chemin.append(Chemin)
        Liste_Dossier.append(d)
        Liste_Num.append(im)

# Analyse des images

Donnees = Analyse_Globale(Liste_Chemin)

## Reconnaissance automatique

# Fermeture des images

plt.close('all')

# Ouverture et analyse de l'image recherchée
print("Quelle image souhaitez vous identifier ?")
print("01 ou 11 ou 12 ou 21 ou 22 ou 31 ou 41 ou 42 ou 51 ou 52 ou 61 ou 71 ?")
print("")
print("Ecrivez le numéro souhaité :")
selec=input()
Recherche = "Recherche\\"
Im_Cherchee_Chemin = Recherche + selec + ".png"
Im_Cherchee = Lecture(Im_Cherchee_Chemin)
Affiche(Im_Cherchee)
Im_Cherchee_Infos = Analyse(Im_Cherchee)

# Recherche des k plus proches voisins

k = 5
Resultat_Proches = Proches(Im_Cherchee_Infos,Donnees,k)
Resultat_Ind = [Resultat_Proches[i][1] for i in range(k)]
Resultat_Dossiers = [Liste_Dossier[ind] for ind in Resultat_Ind]
Resultat_Num = [Liste_Num[ind] for ind in Resultat_Ind]
print("Dossiers trouvés: ",Resultat_Dossiers)
print("Numéros des images: ",Resultat_Num)

# Selection du résultat

def Max_Occurences(L):
    Res = [0 for _ in range(Nb_Dossiers)]
    for t in L:
        Res[t] += 1
    Max = max(Res)
    for i in range(Nb_Dossiers):
        if Res[i]==Max:
            return i

Dossiers_final = Max_Occurences(Resultat_Dossiers)
print("Dossier: ",Dossiers_final)

# Affichage du résultat

Im_Trouvee_Chemin = Sources + str(Dossiers_final) + "\\0" + ".png"
Im_Trouvee = Lecture(Im_Trouvee_Chemin)
Affiche(Im_Trouvee)
print("Vous pouvez vérifier si les 2 figures sont les mêmes !")

```