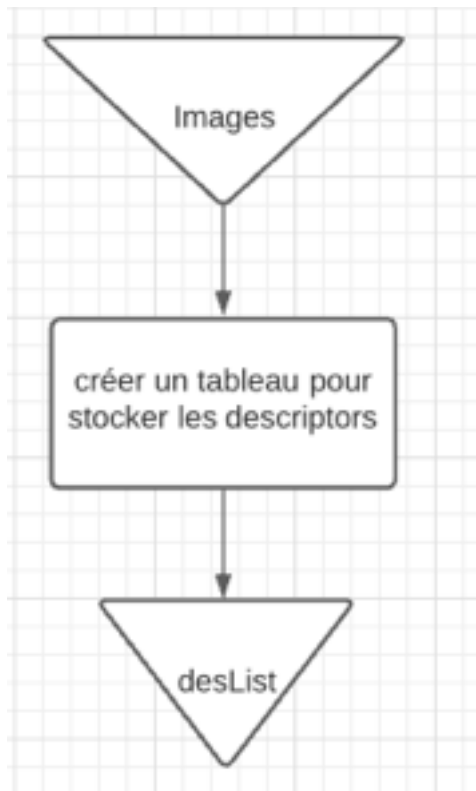


# Reconnaissance

FindDescriptor() :

```
#Trouve les descriptors/keypoints des images et les ajoute a desList
#un descriptor ou keypoints est un element permettant de caractériser une
image
def findDescriptor(images):
    desList = []
    for i in images:
        kp, des = orb.detectAndCompute(i, None)
        desList.append(des)
    return desList
```

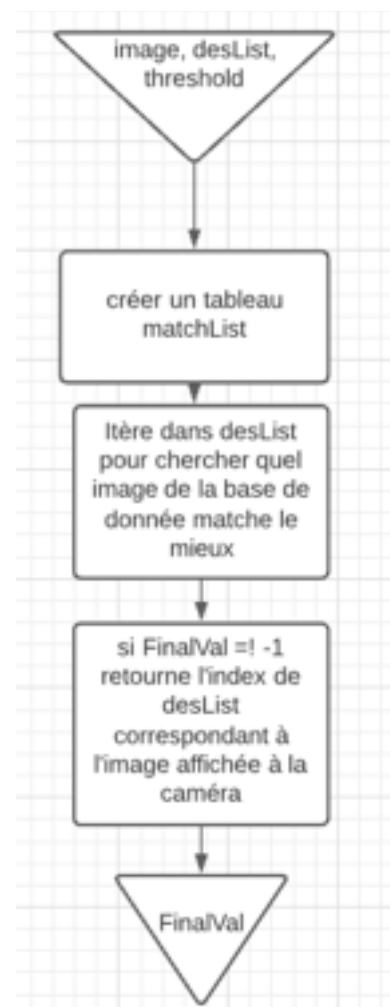


La fonction FindDescriptor sert à trouver et stocker les descriptifs de chaque image de la base de données dans un tableau afin de pouvoir comparer ensuite les images perçues à la caméra avec chacune des valeurs des index du tableau.

FindID() :

```
def findID(img, desList, thres = 3):
    kp2, des2 = orb.detectAndCompute(img, None)
    bf = cv2.BFMatcher(cv2.NORM_HAMMING)
    #brute force matcher ==> compare les images en fonctions de leur
    descriptor
    matchList = []
    finalVal = -1
    try:
        #compare le descriptor d'une image avec la liste de descriptors de
        la bdd
        for des in desList:
            matches = bf.knnMatch(des, des2, k=2)
            good = []
            for m, n in matches:
                if m.distance < 0.69 * n.distance:
                    good.append([m])
            matchList.append(len(good))
    except:
        pass
    # print(matchList)
    if len(matchList) != 0:
        if max(matchList) > thres:
            #retourne l'image avec le plus de similitudes
            finalVal = matchList.index(max(matchList))
    return finalVal
```

La fonction FindID cherche l'index du tableau de descripteurs mentionné juste au-dessus qui a le plus de similitudes le mieux avec l'image de la caméra.



Reco() :

```
def reco():  
  
    success, img2 = cap.read()  
    imgOriginal = img2.copy()  
    #copie l'image de la caméra  
    img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)  
    #la transforme en nuances de gris  
    img2 = cv2.GaussianBlur(img2, (7,7), 0)  
    #la floute  
  
    id = findID(img2, desList)  
    if id != -1:  
        #si la fonction findID retourne un resultat coherent, l'affiche sur  
        le flux camera  
        cv2.putText(imgOriginal, classNames[id], (50, 50),  
cv2.FONT_HERSHEY_COMPLEX, 2, (255, 0, 255), 2)  
        successimg.append(classNames[id])  
  
    cv2.imshow('img2', imgOriginal)
```

La fonction Reco() est la fonction principale de reconnaissance, elle fait appel à FindID() qui elle utilise le tableau de descripteurs de FindDescriptors(). Son principe est simple, elle lit l'image de la caméra, la copie, la transforme en nuances de gris puis la floute. Ensuite, si l'appel à la fonction FindID s'avère positif, c'est à dire un retour autre que -1, elle affiche à l'écran le résultat. Cette fonction marche bien pour toutes les cartes sauf le 6 et le 9, en effet trop de similitudes sont présentes entre ces deux cartes.

