

Computer Vision

R10921104 王怡堯

(a) Dilation

Dilation = Erosion 裡的判斷式跟 kernel 01 調換

```
def Dilation(lena,mask):
    m=len(mask)
    n=len(mask[0])
    mm=len(lena)
    nn=len(lena[0])
    res=[[0]*nn for i in range(mm)]
    for i in range(m//2,mm-m//2):
        for j in range(n//2,nn-n//2):
            tmp=1
            check=0
            for k in range(m):
                for l in range(n):
                    if mask[k][l]==0:
                        if lena[i-(m//2-k)][j-(n//2-l)]==1:
                            check=1
                            break
                if check:
                    break
            if check:
                res[i][j]=1
            else:
                res[i][j]=0
    return res

mask1=[[1,0,0,0,1],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[1,0,0,0,1]]
a=Dilation(lena_arr//128,mask1)
img.fromarray(np.array(a,dtype='uint8')*255)
```



Computer Vision
R10921104 王怡堯

(b) Erosion

```
def Erosion(lena,mask):
    m=len(mask)
    n=len(mask[0])
    mm=len(lena)
    nn=len(lena[0])
    res=[[0]*nn for i in range(mm)]
    for i in range(m//2,mm-m//2):
        for j in range(n//2,nn-n//2):
            tmp=1
            check=0
            for k in range(m):
                for l in range(n):
                    if mask[k][l]==1:
                        if lena[i-(m//2-k)][j-(n//2-l)]==0:
                            check=1
                            break
                if check:
                    break
            if check:
                res[i][j]=0
            else:
                res[i][j]=1
    return res

mask1=[[0,1,1,1,0],[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1],[0,1,1,1,0]]
b=Erosion(lena_arr//128,mask1)
img.fromarray(np.array(b,dtype='uint8')*255)
```



(c) Opening

```
def opening(lena,mask):  
    return Dilation(Erosion(lena,mask),-np.array(mask)+1)
```



(d) Closing

```
def closing(lena,mask):  
    return Erosion(Dilation(lena,-np.array(mask)+1),mask)
```



Computer Vision

R10921104 王怡堯

(e) Hit-and-miss transform

```
def hit_and_miss(lena,mask1,mask2):  
    return np.array(Erosion(lena,mask1))*np.array(Erosion(-lena+1,mask2))
```

