MIT Lab.
Multimedia
Intelligent Technical
Laboratory

影像形狀的轉換

目標任務

使用形狀轉換公式對影像進行縮放、旋轉、仿射轉換、透射轉換

(1) 影像縮放

- 縮放倍率：1/2

(2) 影像旋轉

- 中心點：影像中心
- 旋轉角度：逆時針45度
- 縮放：1

(3) 影像仿射轉換

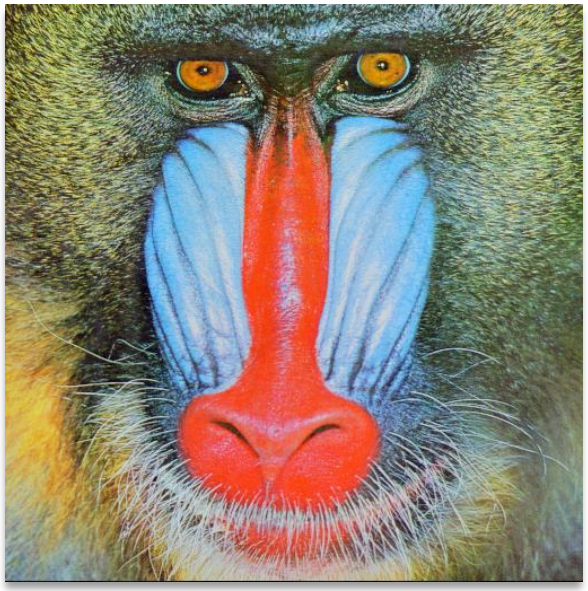
- pts1: [160, 165], [240, 390], [270, 125]
- pts2: [190, 140], [190, 375], [310, 140]

(4) 影像透射轉換

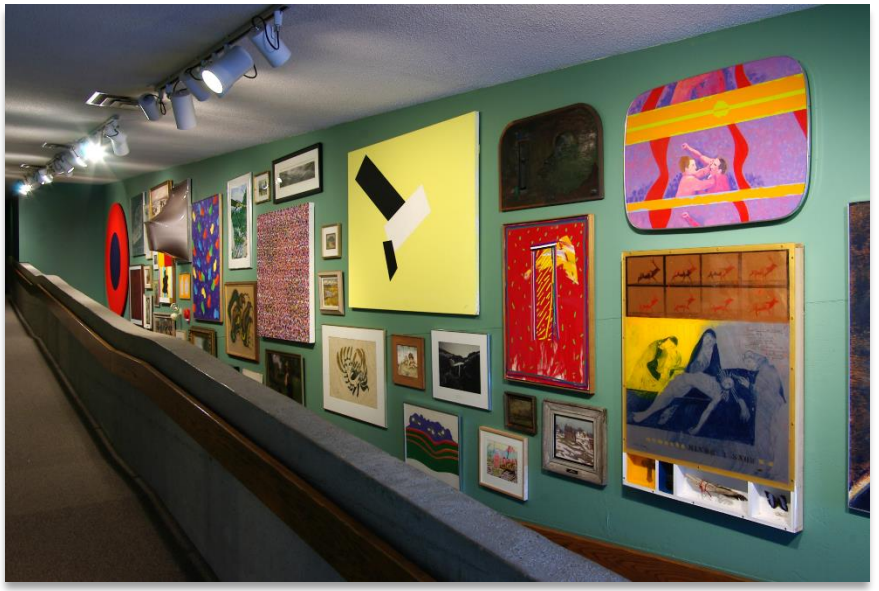
- pts1: [795, 350], [795, 690], [1090, 720], [1090, 250]
- pts2: [0, 0], [0, 500], [650, 500], [650, 0]

測試影像

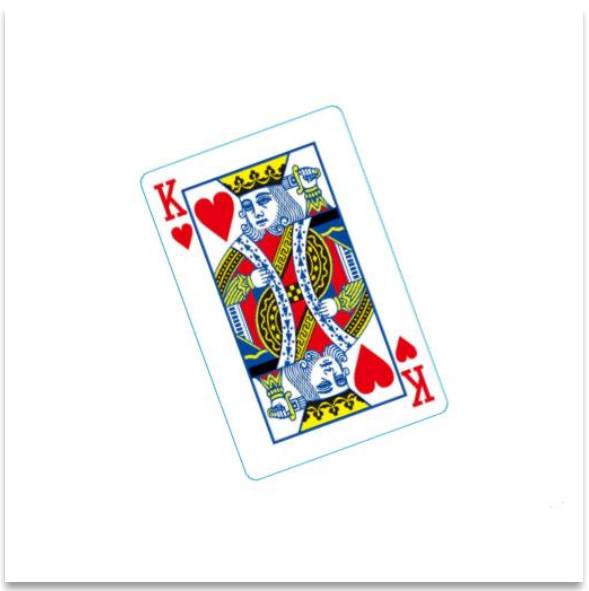
Baboon.bmp



Gallery.bmp



Poker.bmp



影像縮放

- **縮放影像用:**
  - ✓ ***cv2.resize(src, dsize)***

```
def Scale(src):
    H, W, C = src.shape
    N_Image = cv2.resize(src,            , interpolation=cv2.INTER_NEAREST)
    L_Image = cv2.resize(src,            , interpolation=cv2.INTER_LINEAR)
    C_Image = cv2.resize(src,            , interpolation=cv2.INTER_CUBIC)
    return N_Image, L_Image, C_Image
```

影像旋轉

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

- **建立旋轉矩陣:**
  - ✓ ***cv2.getRotationMatrix2D(center, angle, scale)***
- **使用旋轉矩陣:**
  - ✓ ***cv2.warpAffine(src, M, dsize)***

```
def Rotation(src):
    H, W, C = src.shape
    M_rotation = cv2.getRotationMatrix2D(                    )
    rotationImage = cv2.warpAffine(src, M_rotation, (H, W))
    return rotationImage
```

仿射變換

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

- 建立仿射矩陣:
  - ✓ *cv2.getAffineTransform(src, dst)*
- 使用仿射矩陣:
  - ✓ *cv2.warpAffine(src, M, dsize)*

```
def Affine(src):
    H, W, C = src.shape
    pts1 = np.array(                                    , dtype=np.float32)
    pts2 = np.array(                                    , dtype=np.float32)
    M_affine = cv2.getAffineTransform(pts1, pts2)
    affineImage = cv2.warpAffine(src, M_affine, (H, W))
    return affineImage
```

影像縮放

- 建立透視矩陣:
  - ✓ *cv2.getPerspectiveTransform(src, dst)*
- 使用透視矩陣:
  - ✓ *cv2.warpPerspective(src, M, dsize)*

```
def Perspective(src):
    pts1 = np.array(                                                    , dtype=np.float32)
    pts2 = np.array(                                                , dtype=np.float32)
    M = cv2.getPerspectiveTransform(pts1, pts2)
    perspectiveImage = cv2.warpPerspective(src, M, (650, 500))
    return perspectiveImage
```

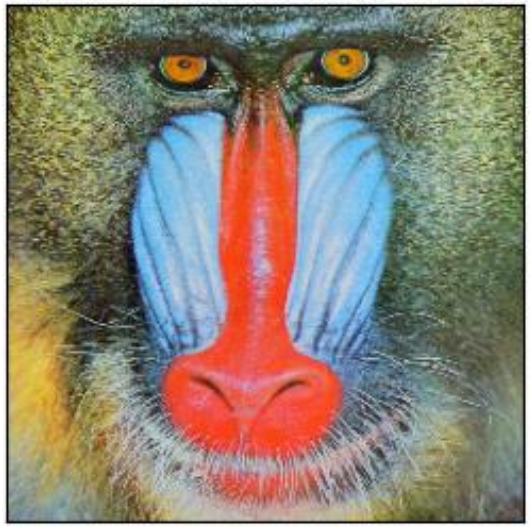
主程式

MIT Lab.
Multimedia
Intelligent Technical
Laboratory

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

image = cv2.imread('./Baboon.bmp')
N_Image, L_Image, C_Image = Scale(image)
image = cv2.imread('./Baboon.bmp')
rotationImage = Rotation(image)
image = cv2.imread('./Poker.bmp')
affineImage = Affine(image)
image = cv2.imread('./Gallery.bmp')
perspectiveImage = Perspective(image)

images = [N_Image, L_Image, C_Image, rotationImage, affineImage, perspectiveImage]
titles = ['NEAREST', 'LINEAR', 'CUBIC', 'ROTATION', 'AFFINE', 'PERSPECTIVE']

plt.figure()
for i in range(len(images)):
    plt.subplot(2, 3, i+1), plt.imshow(images[i][..., ::-1])
    plt.title(titles[i])
    plt.xticks([]), plt.yticks([])
plt.show()
```

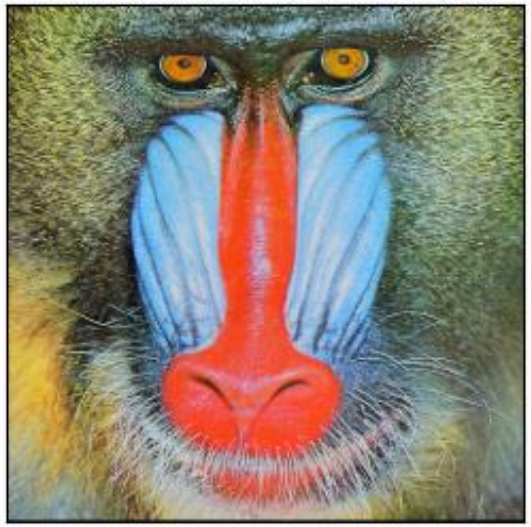
實作結果

NEAREST



LINEAR

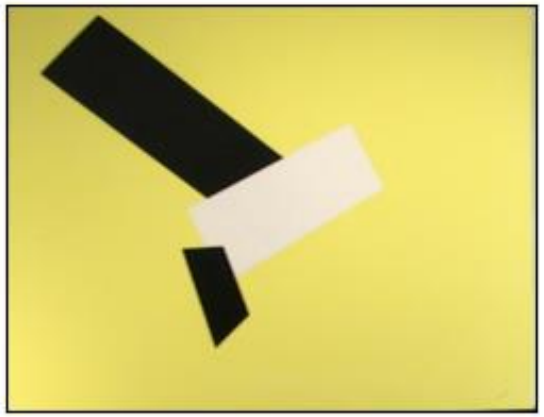
CUBIC



ROTATION

AFFINE

PERSPECTIVE



MIT Lab.
Multimedia
Intelligent Technical
Laboratory

Thanks for listening