利用影像處理技術提升老舊照片的整體辨識度

醫學影像期末專題報告

B0928013 吳佳恩
B0928015 艾思嘉
B0928031 鄭茹云

目錄

01

題目介紹

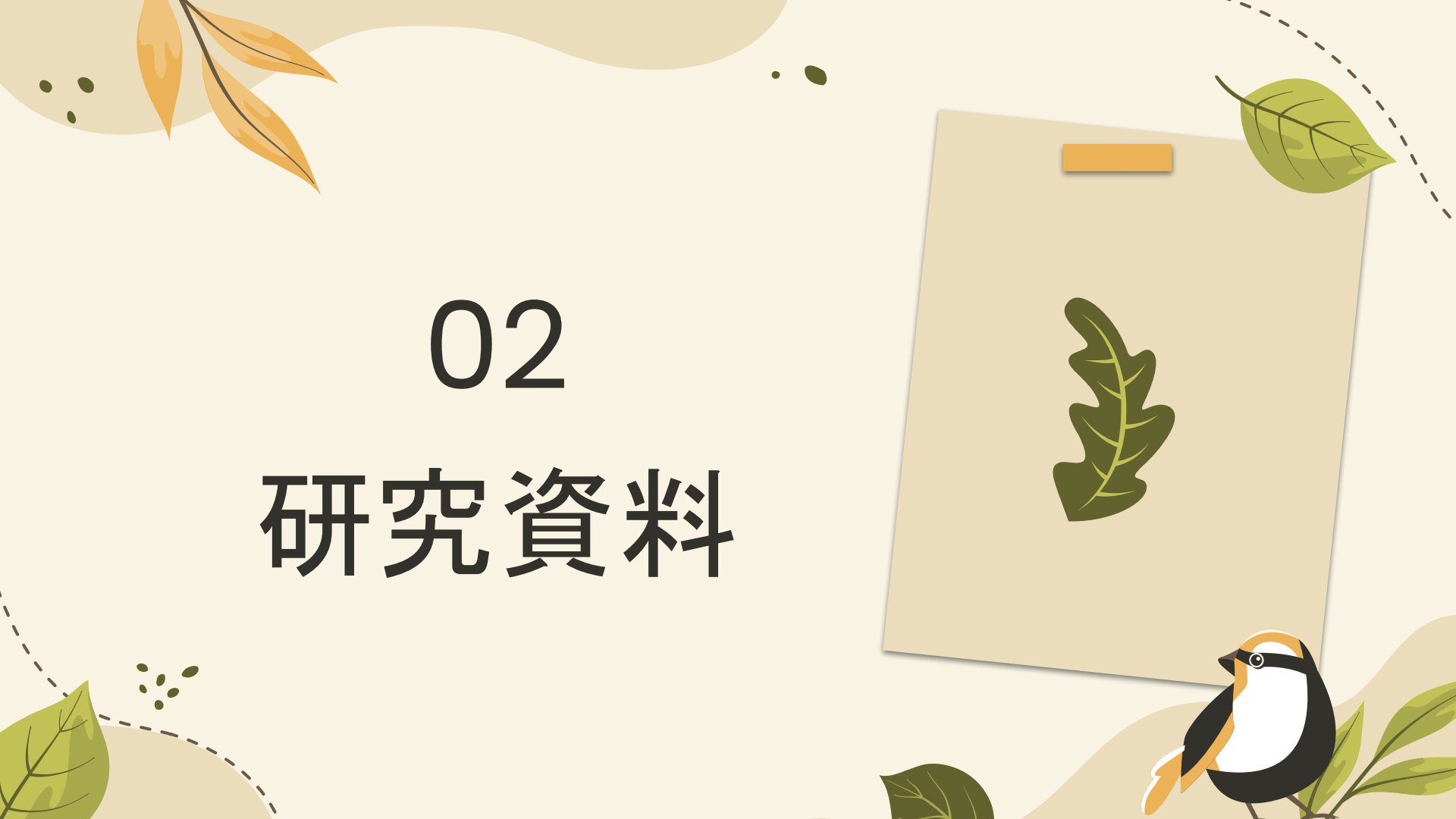
題目簡述

老舊照片保存了許多珍貴的歷史紀錄，但時常因當時的技術限制，照片辨識度往往不高。

- 右圖可以發現到印刷的痕跡呈現**網狀**，導致圖中五官不清楚
- 並且整體線條感不足，造成區塊與區塊間邊界感不足

因此，**我們將運用影像處理技術來嘗試修復這些照片**，讓人們更好地理解和欣賞這些珍貴的歷史記錄。

02
研究資料



研究資料

- 透過影像處理的方式來提高老舊照片的整體辨識度。
- 使用Python的OpenCV圖像處理庫進行製作
- 影像資料預計使用**台大圖書館的數位典藏館的台灣舊照片資料庫**

網址:
https://dl.lib.ntu.edu.tw/s/photo/page/Home

03

實作過程

研究步驟1 -resize(程式)

```
resized_image1 = cv2.resize(image, (width, height), interpolation=cv2.INTER_NEAREST)
resized_image2 = cv2.resize(image, (width, height), interpolation=cv2.INTER_LINEAR)
resized_image3 = cv2.resize(image, (width, height), interpolation=cv2.INTER_CUBIC)
resized_image4 = cv2.resize(image, (width, height), interpolation=cv2.INTER_AREA)
resized_image5 = cv2.resize(image, (width, height), interpolation=cv2.INTER_LANCZOS4)
```

研究步驟1 -resize(結果)

Before

After





研究步驟2 –去雜訊(程式)

1. Bilateral Filter

```
denoised = cv2.bilateralFilter(img8, 100, 75, 75)
```

2. Median Blur

```
denoised = cv2.medianBlur(img9, 10)
```

3. Frequency Domain Filtering

```
def apply_frequency_domain_filter(image, cutoff_radius):
    fft = np.fft.fft2(image)
    fft_shifted = np.fft.fftshift(fft)

    rows, cols, channels = image.shape
    crow, ccol = int(rows / 2), int(cols / 2)

    mask = np.ones((rows, cols, channels), dtype=np.uint8)
    mask[crow - cutoff_radius: crow + cutoff_radius, ccol - cutoff_radius: ccol + cutoff_radius, :] = 0

    fft_shifted_filtered = fft_shifted * mask


    fft_filtered = np.fft.ifftshift(fft_shifted_filtered)
    image_filtered = np.fft.ifft2(fft_filtered)
    image_filtered = np.abs(image_filtered)


    image_filtered = image_filtered.astype(np.uint8)
    image_filtered = cv2.normalize(image_filtered, None, 0, 255, cv2.NORM_MINMAX)

    return image_filtered


image =img1

cutoff_radius = 40
filtered_image = apply_frequency_domain_filter(image, cutoff_radius)
```

研究步驟2 –去雜訊(結果)

Original

bilateral Filter

研究步驟2 –去雜訊(結果)

Original

Median Blur

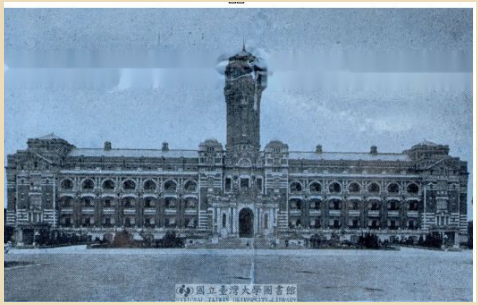
研究步驟2 –去雜訊(結果)

Original

Frequency Domain Filtering

研究步驟2 -去雜訊(結果)

Before



After

bilateral Filter

Median Blur

Frequency Domain Filtering

研究步驟3 -修補破損(程式)

```python
# 圓形方法1
import cv2
import numpy as np

image = cv2.imread('default.jpg')#

# 建立
mask = np.zeros(image.shape[:2], dtype=np.uint8)

def draw_mask(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:  # 按下左鍵時開始
        cv2.circle(mask, (x, y), 15, (255), -1)
    elif event == cv2.EVENT_RBUTTONDOWN:  # 按下右鍵時停止
        cv2.circle(mask, (x, y), 15, (0), -1)

cv2.namedWindow('Image')
cv2.setMouseCallback('Image', draw_mask)

while True:
    # 顯示
    cv2.imshow('Image', image)
    cv2.imshow('Mask', mask)

    key = cv2.waitKey(1)
    if key == 27:  # 按下 ESC 鍵退出迴圈
        break
    elif key == ord('r'):  # 按下 'r' 鍵重置
        mask = np.zeros(image.shape[:2], dtype=np.uint8)

cv2.destroyAllWindows()
```

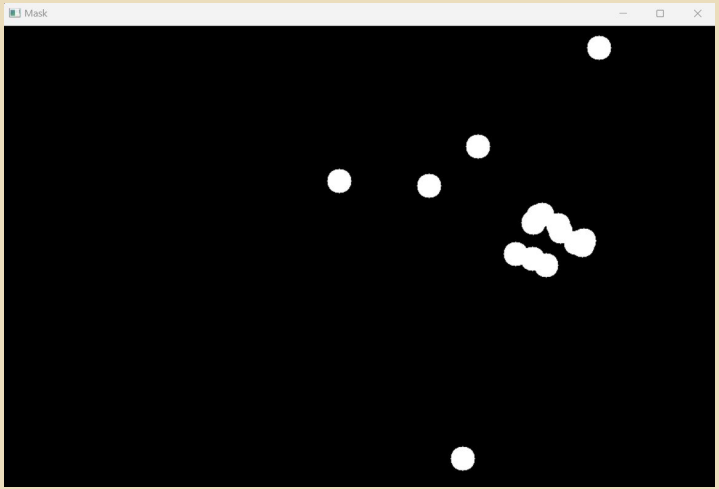
圓形

```python
#方形
import cv2
import numpy as np

image = cv2.imread('default (4).jpg')

# 建立空白遮罩
mask = np.zeros(image.shape[:2], dtype=np.uint8)

# 追蹤滑鼠
start_point = None
end_point = None
drawing = False

def draw_mask(event, x, y, flags, param):
    global start_point, end_point, drawing

    if event == cv2.EVENT_LBUTTONDOWN:  # 按下滑鼠左鍵開始
        start_point = (x, y)
        drawing = True
    elif event == cv2.EVENT_LBUTTONUP:  # 釋放停止
        end_point = (x, y)
        drawing = False
        cv2.rectangle(mask, start_point, end_point, (255), -1)

cv2.namedWindow('Image')
cv2.setMouseCallback('Image', draw_mask)

while True:
    cv2.imshow('Image', image)
    cv2.imshow('Mask4', mask)

    key = cv2.waitKey(1)
    if key == 27:  # ESC 退出
        break
    elif key == ord('r'):  # 'r' 重置
        mask = np.zeros(image.shape[:2], dtype=np.uint8)

cv2.destroyAllWindows()
```

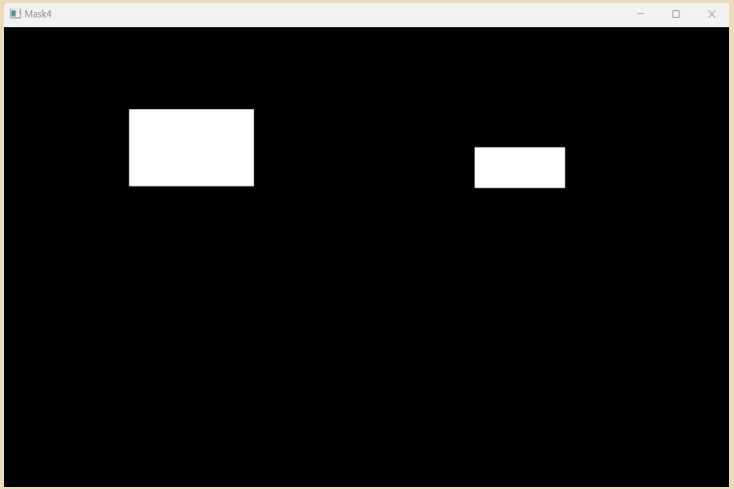
方形

```python
import cv2
import numpy as np

image = cv2.imread('default (4).jpg')
mask = cv2.imread('Mask4.jpeg', 0)
result = cv2.inpaint(image, mask, 3, cv2.INPAINT_TELEA)

# 顯示修復後的照片
cv2.imshow('Result', result)

# 將修復後的照片儲存到文件
cv2.imwrite('default (44).jpg', result)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

研究步驟3 -修補破損

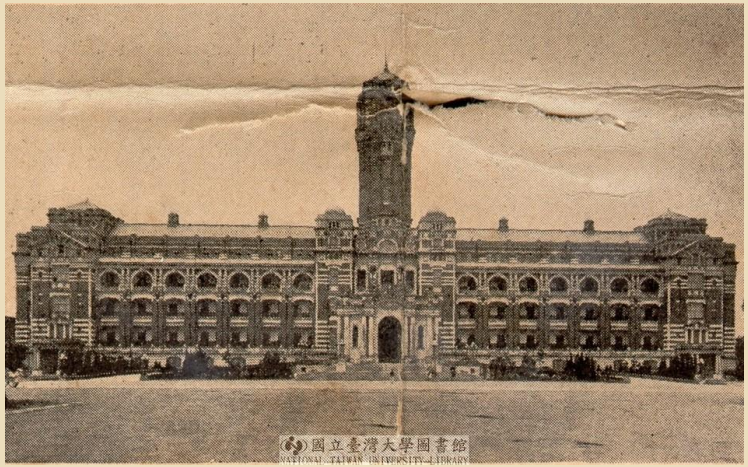


圓形



方形

研究步驟3 -修補破損(結果)



Before

After

研究步驟3 -去浮水印(額外)

Before

After

研究步驟4 –調整對比(程式)

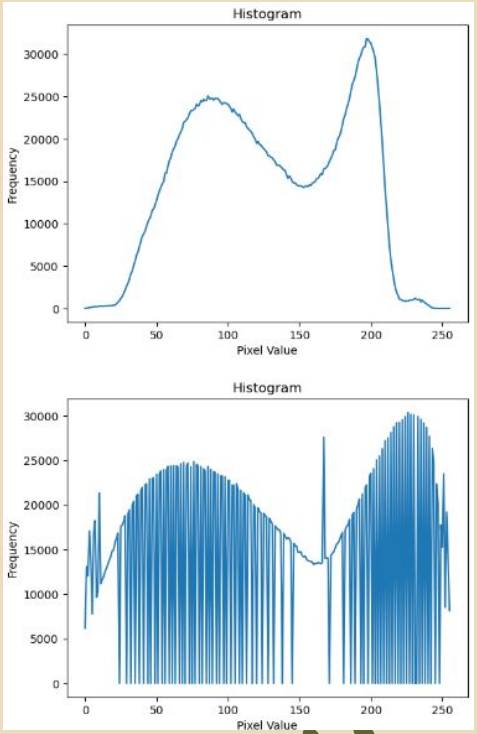
```python
import cv2
import numpy as np
import matplotlib.pyplot as plt

def display_histogram(image):
    hist = cv2.calcHist([image], [0], None, [256], [0, 256])

    plt.figure()
    plt.title('Histogram')
    plt.xlabel('Pixel Value')
    plt.ylabel('Frequency')
    plt.plot(hist)
    plt.show()

def histogram_equalization(image):
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    equalized_image = cv2.equalizeHist(gray_image)

    return equalized_image
```

```python
image_path = 'default (7).jpg'
image = cv2.imread(image_path)

cv2.imshow('Original Image', image)
cv2.waitKey(0)

display_histogram(image)

equalized_image = histogram_equalization(image)

cv2.imshow('Equalized Image', equalized_image)
cv2.waitKey(0)

display_histogram(equalized_image)

output_path = 'equalized7.jpg'
cv2.imwrite(output_path, equalized_image)

print(f"均衡化後的圖像已儲存至 {output_path}")
```

研究步驟4 -調整對比(結果)

Before

After



研究步驟5 - 銳化調整

- Unsharp Masking

```
#Unsharp Masking
def sharpen(img, i, sigma=25):
    # sigma = 5、15、25
    blur_img = cv2.GaussianBlur(img, (0, 0), sigma)
    usm = cv2.addWeighted(img, 1.5, blur_img, -0.5, 0)
    name = "sharpened_noise_{}".format(i)
    #cv2.imwrite("/Users/celineai/Desktop/大三/醫學影像處理/Final Project/noise_sharpen/{}.jpg".format(name), usm)
    return usm
```

```
#原圖銳化&去雜訊後銳化做比較
fig, axs = plt.subplots(nrows=13, ncols=2, figsize=(12,60))
for i in range(13):
    image = eval(f'img{i+1}')
    img_noise = noise(image, i+1)
    sharpened1 = sharpen(image, i+1)
    sharpened2 = sharpen(img_noise, i+1)

    for k in range(2):
        if(k == 0):
            f_imshow(axs[i][k], sharpened1, f'Origin & sharpen img{i}')
            name = "origin_unsharp_sharpen_{}".format(i)
            cv2.imwrite("/Users/celineai/Desktop/大三/醫學影像處理/Final Project/unsharp_sharpen/{}.jpg".format(name),
        if(k == 1):
            f_imshow(axs[i][k], sharpened2, f'Denoised & sharpen img{i}')
            name = "denoise_unsharp_sharpen_{}".format(i)
            cv2.imwrite("/Users/celineai/Desktop/大三/醫學影像處理/Final Project/denoise_unsharp_sharpen/{}.jpg".format
```

研究步驟5 - 銳化調整

(1) Unsharp Masking 實作結果

04
結果討論

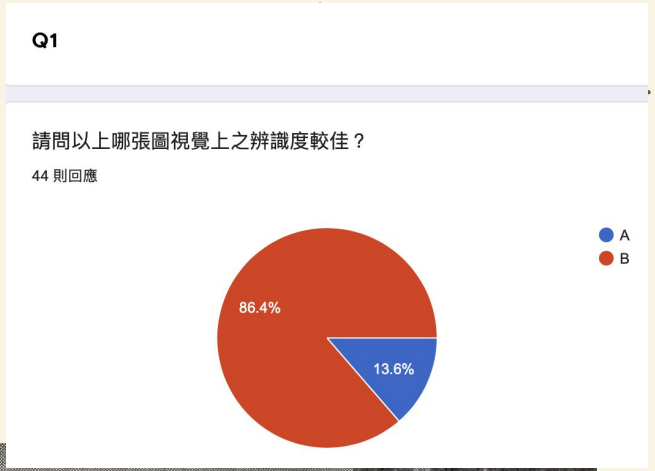
結果討論

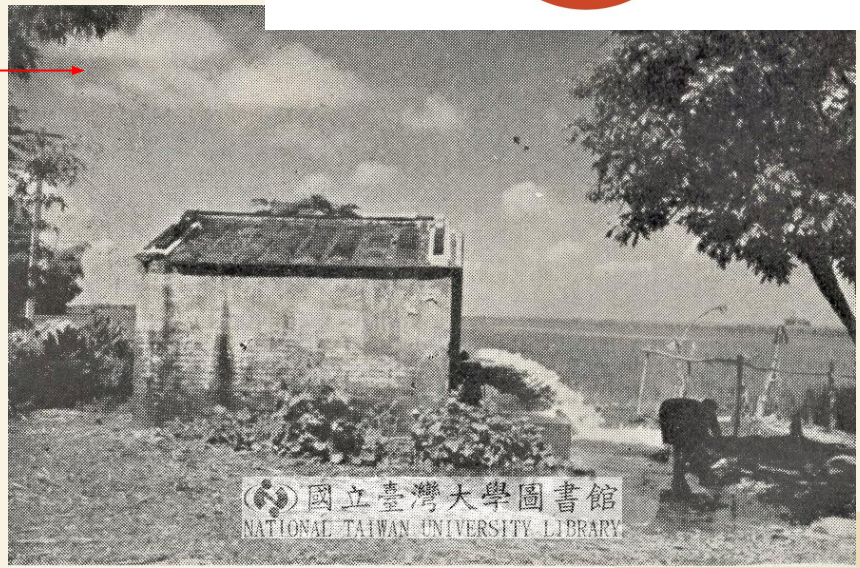
- 問卷調查(共13張相片):
  利用圖片二選一方式，放上原圖以及我們實作處理過後的圖片，
  讓填寫問卷者選擇哪張圖片之視覺辨識度較佳。

共收取到45則回應

結果討論

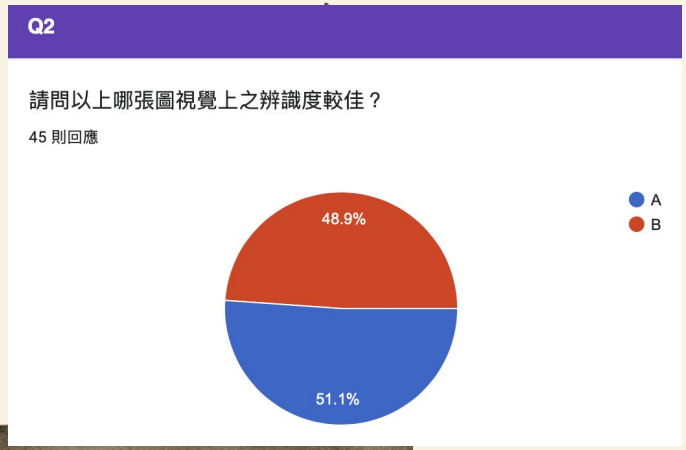
- 問卷調查(共13張相片):
  紅色 -> 處理過後
  藍色 -> 原圖





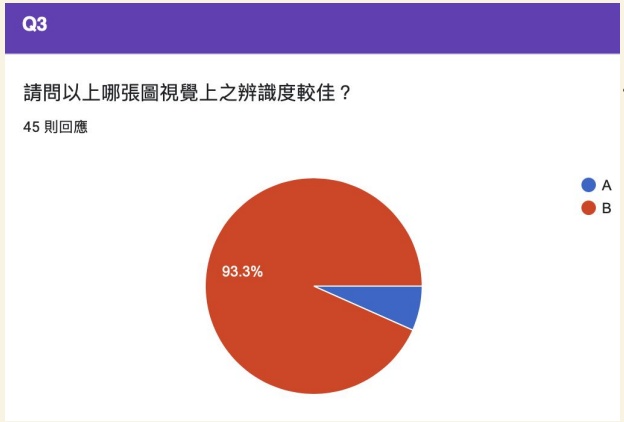
結果討論

- 問卷調查(共13張相片):
  紅色 -> 處理過後
  藍色 -> 原圖



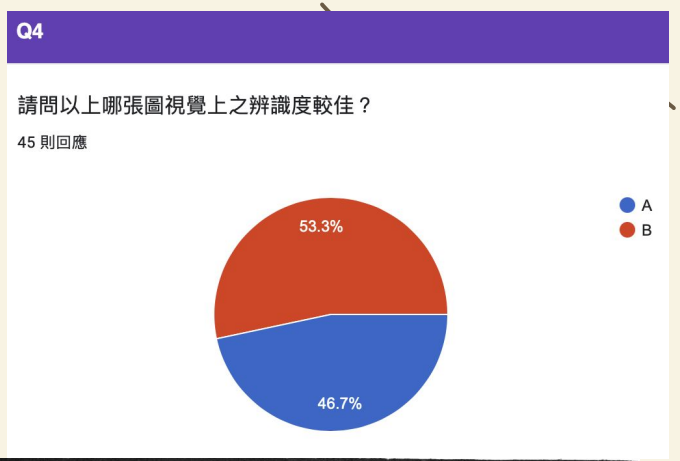
結果討論

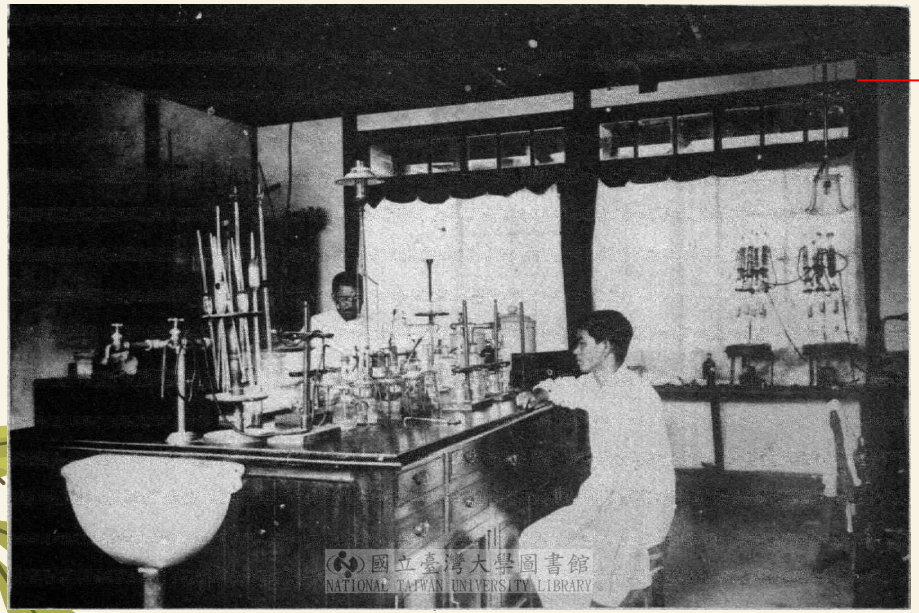
- 問卷調查(共13張相片):
  紅色 -> 處理過後
  藍色 -> 原圖

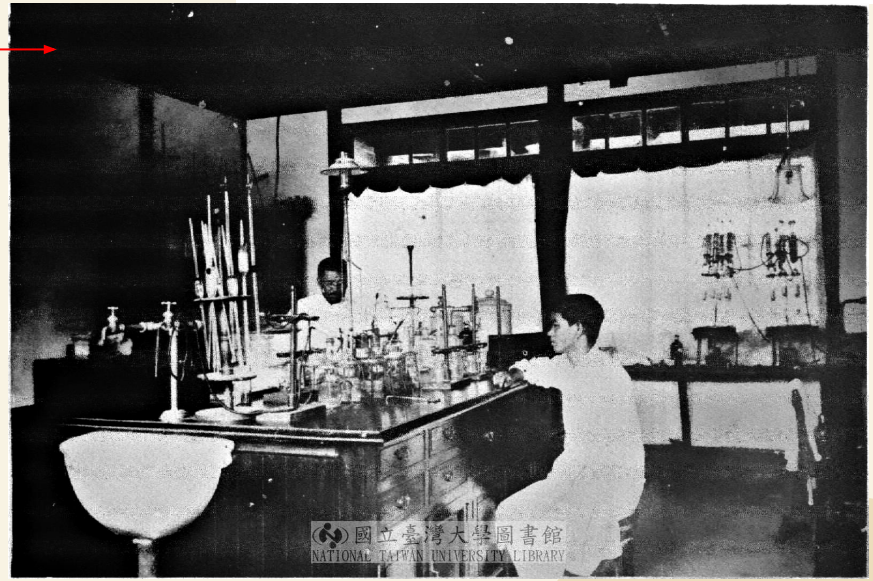


結果討論

- 問卷調查(共13張相片):
  紅色 -> 處理過後
  藍色 -> 原圖

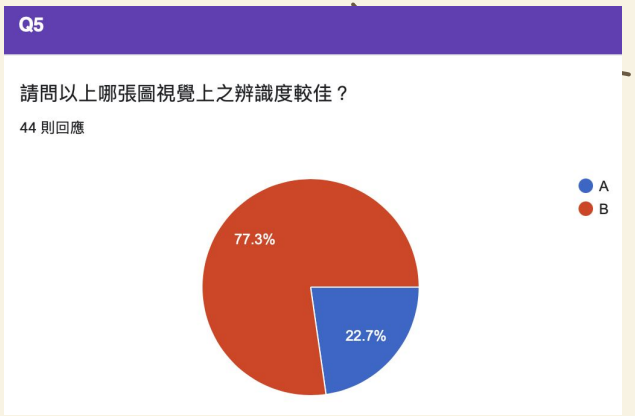


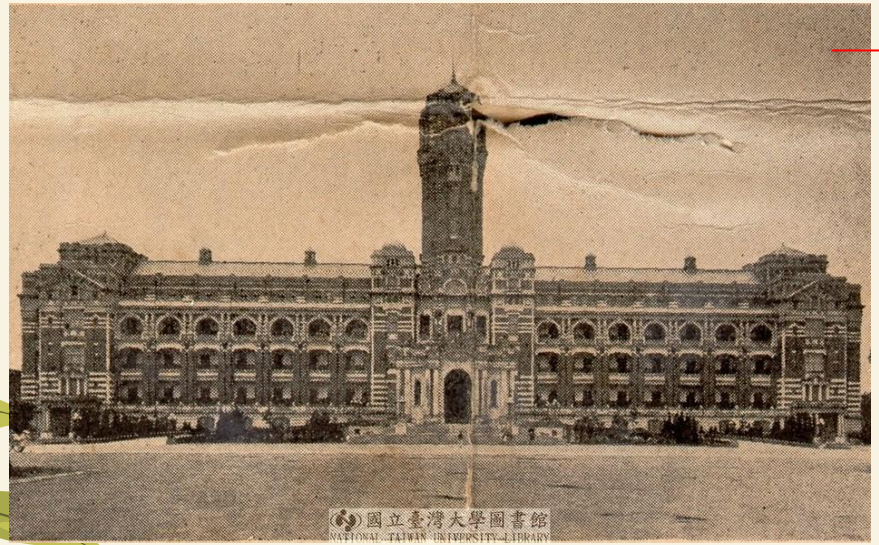




結果討論

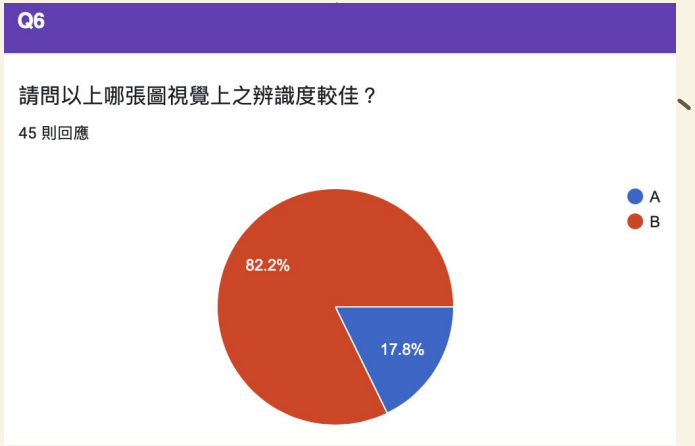
- 問卷調查(共13張相片):
  紅色 -> 處理過後
  藍色 -> 原圖





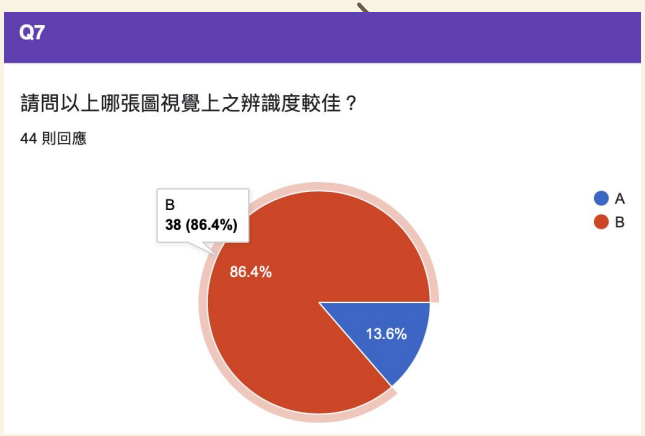
結果討論

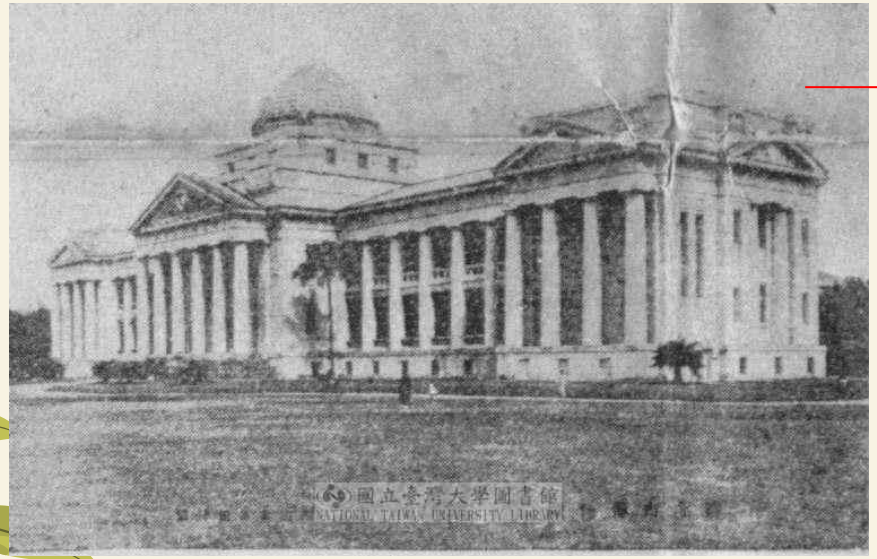
- 問卷調查(共13張相片):
  紅色 -> 處理過後
  藍色 -> 原圖

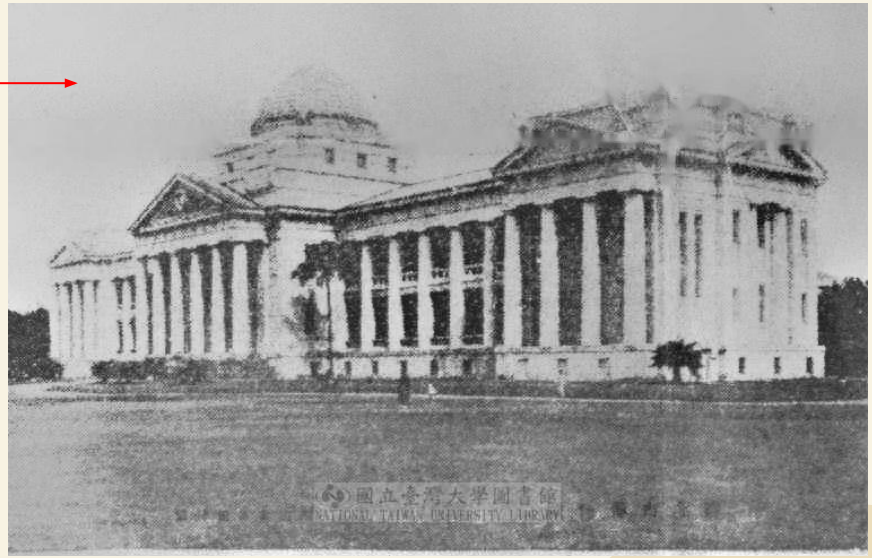


結果討論

- 問卷調查(共13張相片):
  紅色 -> 處理過後
  藍色 -> 原圖

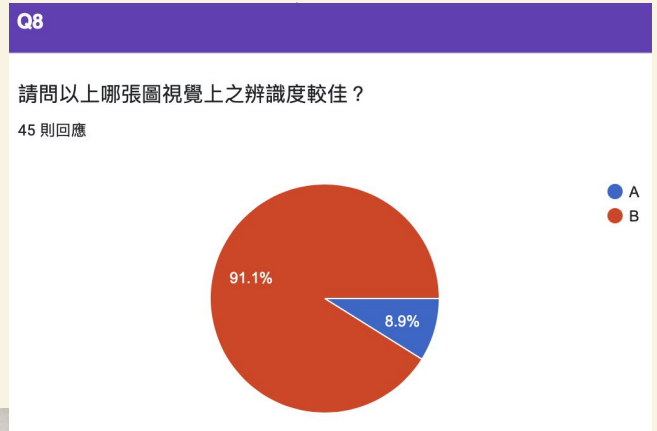






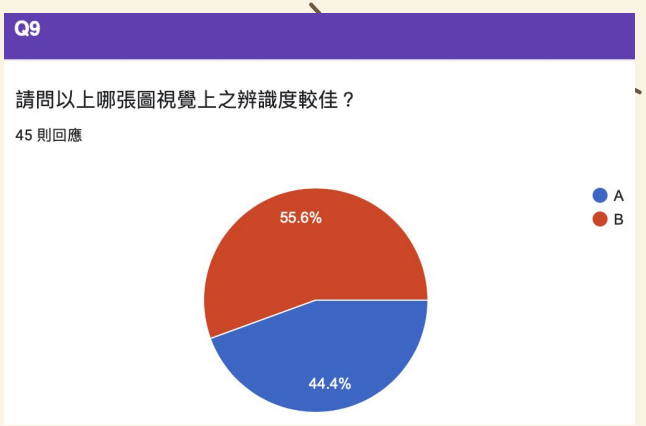
結果討論

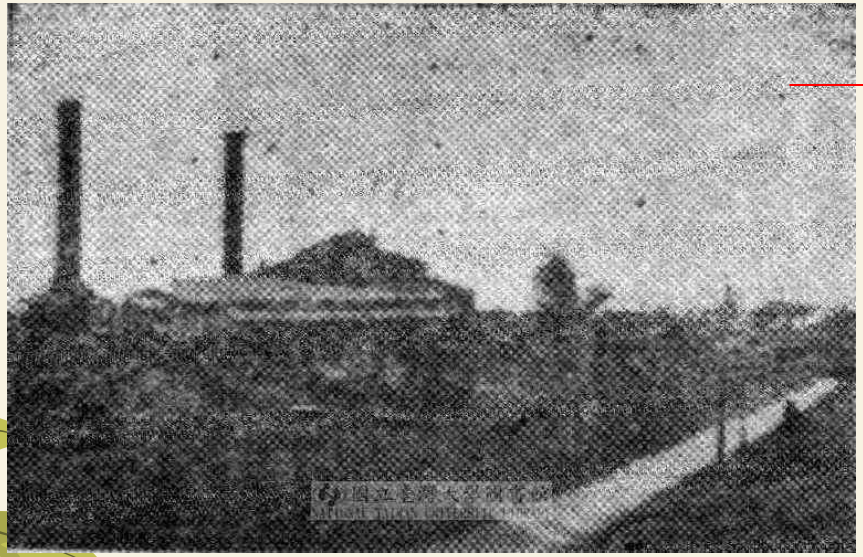
- 問卷調查(共13張相片):
  紅色 -> 處理過後
  藍色 -> 原圖

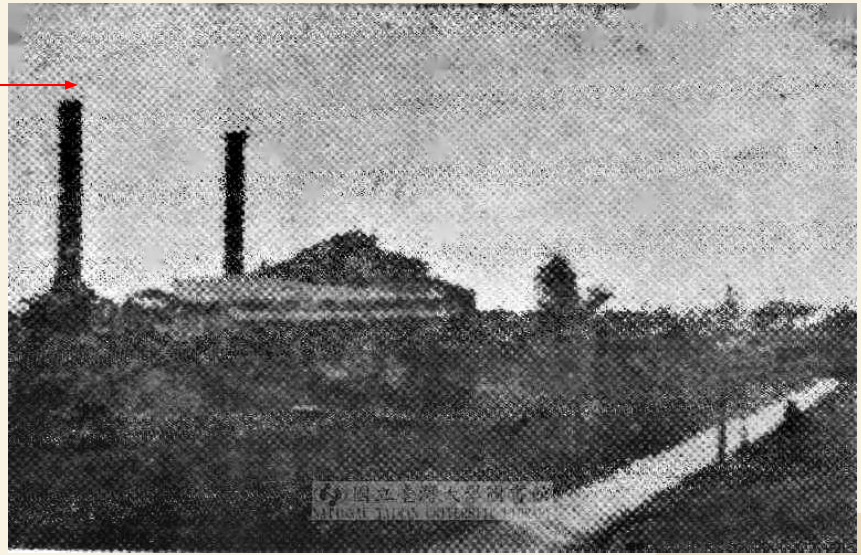


結果討論

- 問卷調查(共13張相片):
  紅色 -> 處理過後
  藍色 -> 原圖

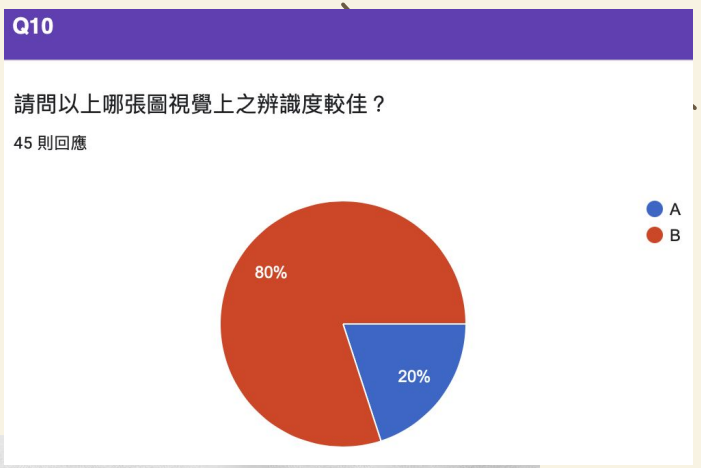


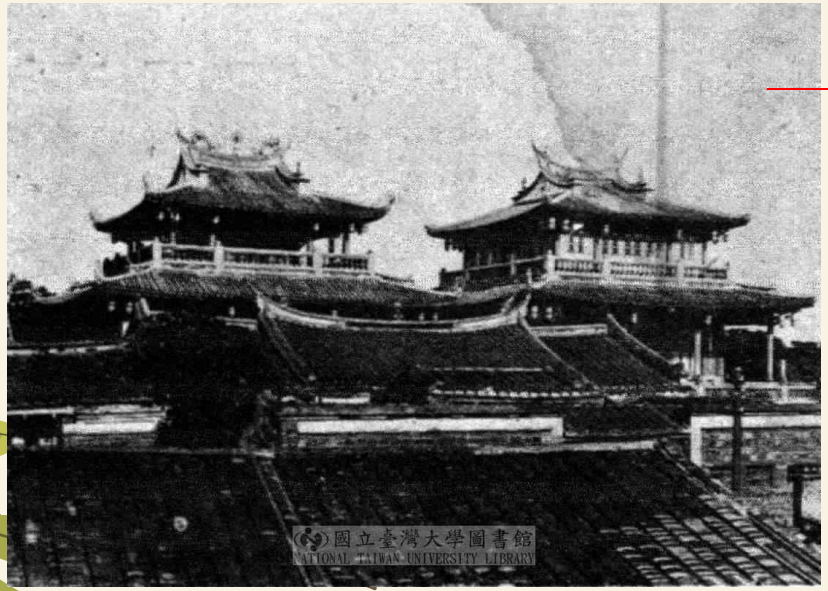




結果討論

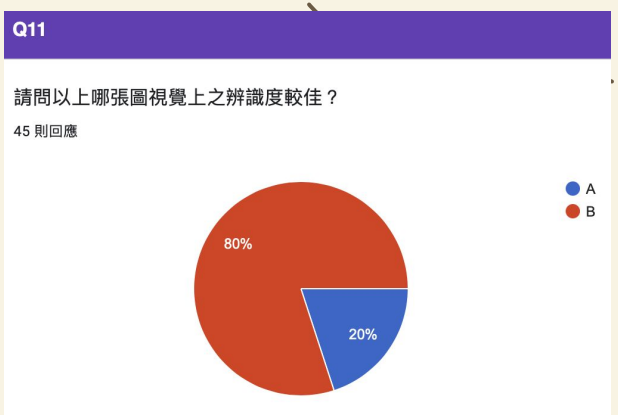
- 問卷調查(共13張相片):
  紅色 -> 處理過後
  藍色 -> 原圖





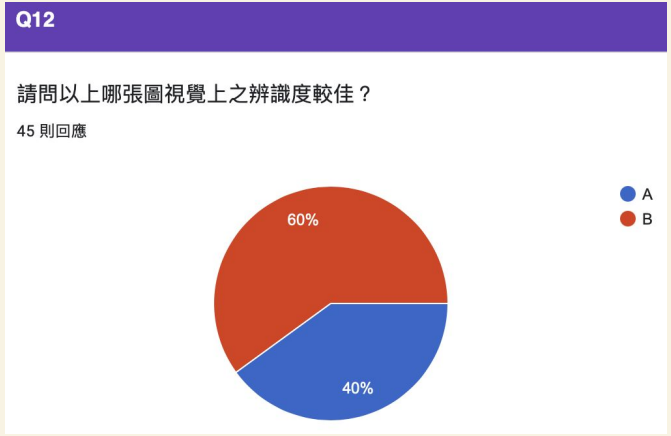
結果討論

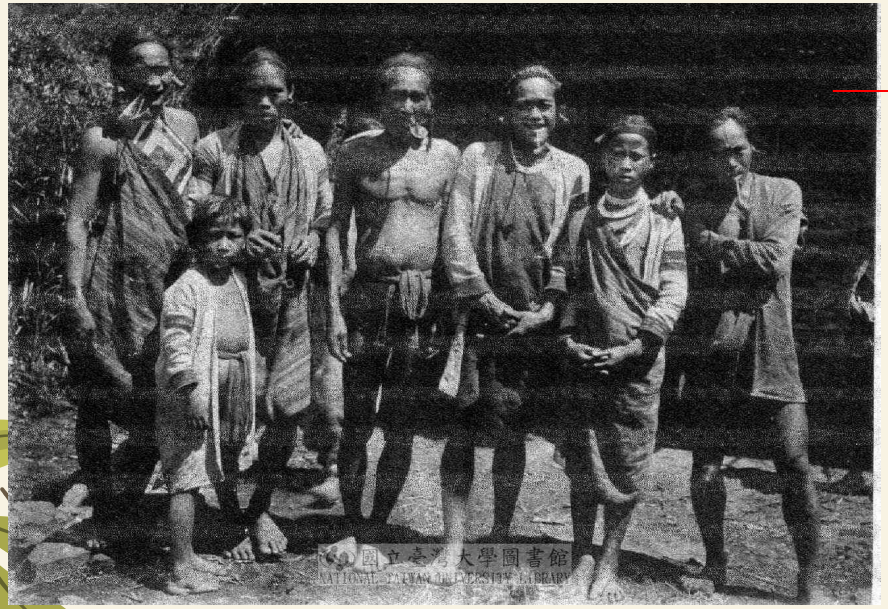
- 問卷調查(共13張相片):
  紅色 -> 處理過後
  藍色 -> 原圖

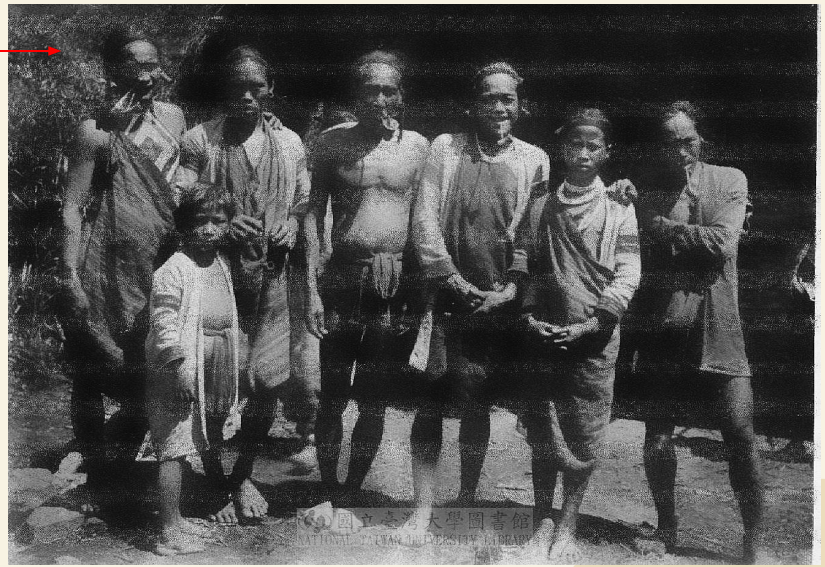


結果討論

- 問卷調查(共13張相片):
  紅色 -> 處理過後
  藍色 -> 原圖

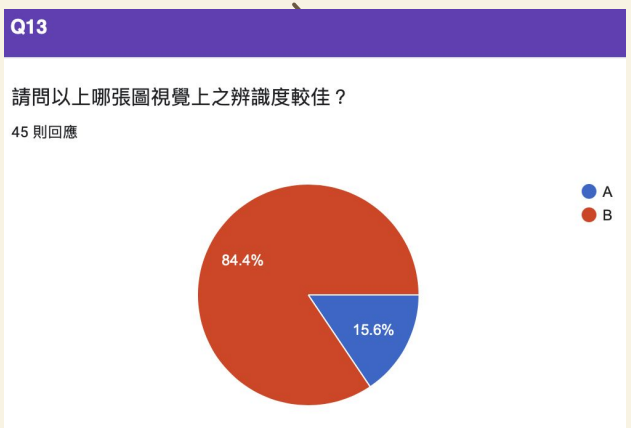


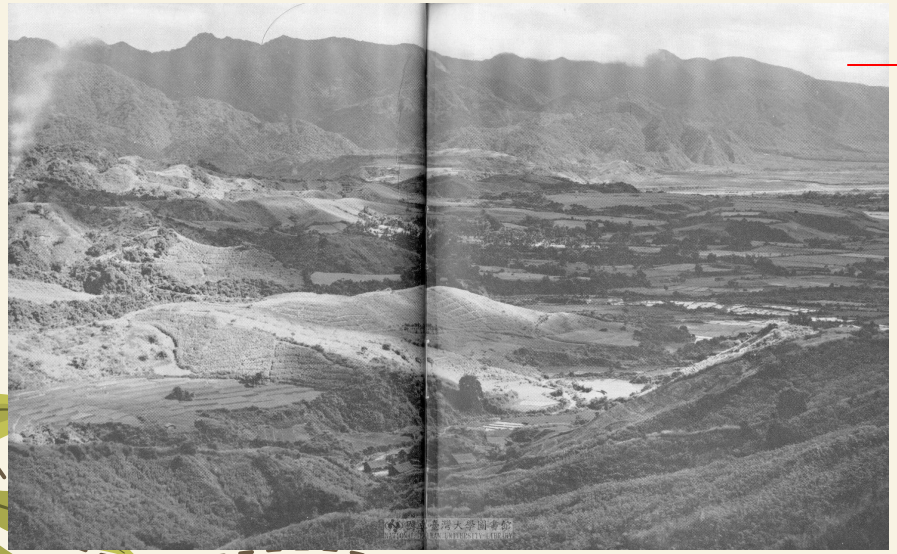


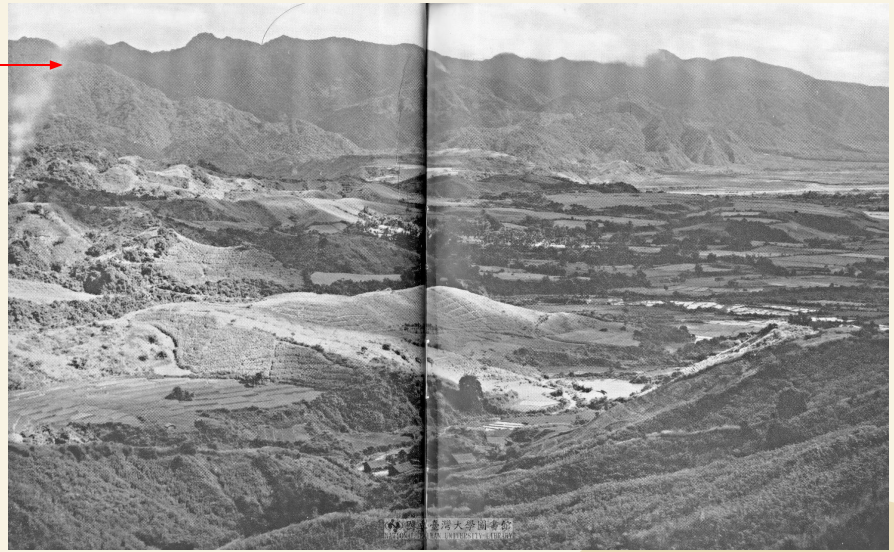


結果討論

- 問卷調查(共13張相片):
  紅色 -> 處理過後
  藍色 -> 原圖







05
結論

結論

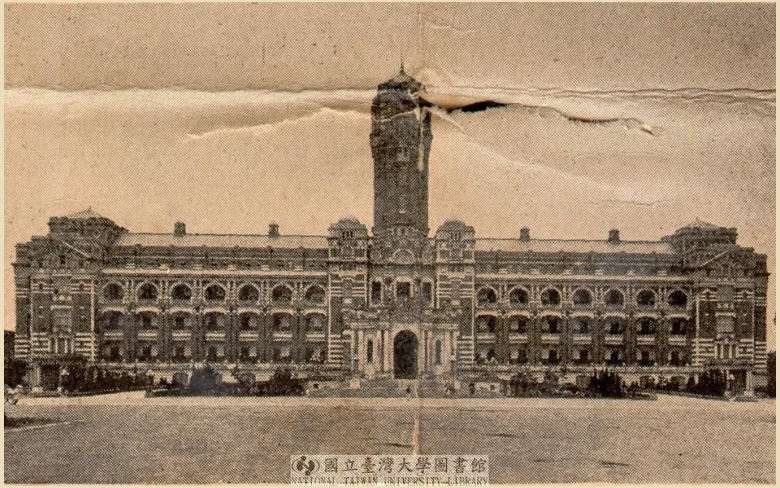
- 不同照片情況適用於不同處理方法
- 去除雜訊的同時須保持圖片不過於平滑 -> 較難達成
- Unsharp Masking處理當中，高斯濾波器所取出之模糊影像程度會大大影響整體 銳化結果

結論

手動繪製優點與缺點:

優 : 可以針對每張影像進行調整, 修補效果也會比較好

缺 : 如果資料量很大的時候, 需花上很多時間



Before

After

結論

06

參考資料
&
小組分工

小組分工

姓名	學號	工作內容
吳佳恩	B0928013	程式：去雜訊&Resize PPT製作 報告
艾思嘉	B0928015	程式：銳化&統整＆問卷調查 PPT製作 報告
鄭茹云	B0928031	程式：調整對比&修補破損 PPT製作 報告

參考資料

- https://jason-chen-1992.weebly.com/home/-unsharp-masking
- https://blog.csdn.net/weixin_42985978/article/details/126517088
- https://chat.openai.com/c/3141d3e1-42c6-4eb4-ac00-6604c2bdf718
- https://dl.lib.ntu.edu.tw/s/photo/page/Home