

屏東縣第 63 屆國中小學科學展覽會
作品說明書

科 別：生活與應用科學科(一)

組 別：國中組

作品名稱：不用再砍樹了嗎？

關 鍵 詞：電子紙、Arduino、OpenCV

編號：B6001

摘要

本研究乃透過 Arduino 開發版控制電子紙顯示，了解到如果只是顯示單純文字、圖案、Logo 或是表格效果成效良好。由於我們所採用是黑白電子紙，進一步想研究圖像（如：建築物、風景、人像）在電子紙上呈現的效果是否真的能取代印刷紙張出來的樣子。

發現到，由於從坊間購入的此款電子紙少了灰階的顯色，而使這些圖像上傳到電子紙的結果表現不一。之後試著利用 OpenCV 先將這些圖像轉換出兩種二值化及邊緣加強，再上傳到電子紙來研究是否能改善在有限的黑白兩色情形下也能讓圖像呈現出明顯的效果。

壹、研究動機

某日和家人到台東縣池上旅遊時，在當地的公車站，可以看到利用太陽能來讓電子紙刷新到站資訊(圖一)。現今學校在教學上所消耗的紙非常多，曾有研究指出將一位接受完 12 年教育學生所拋棄的教科書與普通教材重量達 28 萬 8286 公噸，需砍伐超過 576 萬棵生長 20 至 40 年的樹木，這相當的不環保。撇除教材不說，日常用紙也消耗大量的紙張，若是能將這些以「電子紙」來代替，便能減少許多紙張的使用。

例如：班級課表、布告欄上的海報、名牌等一些不常更換甚至一學期只換一次的東西就很適合以電子紙來代替。電子紙的優點如下：

- 一、電子紙顯示時，靜態圖像並不會耗電，只有在更新畫面時才會用電，因此耗電量低。
- 二、取下電池或拔除電源時螢幕還是會繼續顯示，所以即便電池沒電，也不會回復原狀而是保持最後畫面的狀態，此時裝置的電量消耗為 0 。
- 三、電子紙是運用反射原理，光源打到電子紙上再反射到閱讀者的眼睛，一般紙張無異。
- 四、電子紙重量輕、攜帶方便、製作成本低。



圖一、池上郵局前公車站的到站資訊

貳、研究目的

本研究透過 Arduino 開發板來驅動電子紙，發現電子紙可以取代紙張印刷，可應用在學校的部分，例如：課表、名牌、生活榮譽競賽獎狀。了解如果只是需要顯示文字、表格、圖案的文件，確實可以呈現良好的效果，但如果換成影像或是圖片呈現在電子紙上，會因為只能有黑白兩色而效果大打折扣。

因此我們透過 Python 程式語言引用處理電腦視覺影像功能強大的函式庫 OpenCV，先將原始影像二值化，發現影像在線條的呈現較不明顯，因此我們利用自適應二值化 (adaptive Threshold) 方法處理，分成算術平均法和高斯加權均值法，結果大部分圖片自在黑白影像上都有不錯的效果。因此若要將影像中的輪廓加強顯示在電子紙上，更可以利用 OpenCV 邊緣檢測來強化。

參、研究設備及器材

一、軟體方面

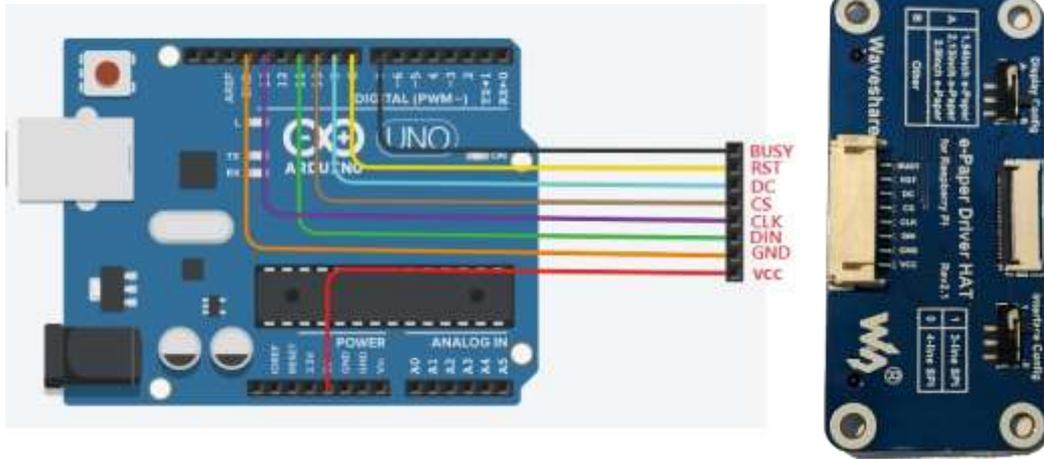
程式語言	Python、C 語言(Arduino 基本語法)
應用軟體	PyCharm IDE

二、硬體方面

	
Arduino MEGA2560 開發版	Arduino UNO 開發版
	
微雪電子紙驅動版	微雪 2.13 英吋電子紙
	
微雪 2.9 英吋電子紙(含驅動版)	微雪 5.83 英吋電子紙

肆、研究過程與結果

一、線路



本研究使用之開發版以 UNO 為主，利用一個 8PIN 的彩色杜邦線和驅動版連結，連接方式為：5V→VCC、GND→GND、D11→DIN、D13→CLK、D10→CS、D9→DC、D8→RST、D7→BUSY)

二、圖片轉換

除了電子紙廠商提供的官方範例程式碼，可以根據程式碼修改想要的顯示文字，但僅只於簡單的數字或是符號。因此在本研究中，無論是中文字、圖案、影像，任何欲顯示在電子紙上的東西，我們都視整個畫面為一張圖片，再將此圖片的每一個像素(pixel)轉換成十六進位的編碼，例如 0xC3 這個十六進位表示法，轉成二進位制是 11000011，在電子紙的像素呈現就如下圖所示：

1 1 0 0 0 0 1 1
○ ○ ● ● ● ● ○ ○

因此若能取得整個圖片的像素轉換成十六進位的編碼矩陣，就可顯示出任何圖片了。我們找到了一個能協助轉換成十六進位編碼的線上工具網站 [image2cpp](#)，只需上傳圖片，選擇要輸出的畫素大小，還有顯示畫素的掃描方向，便能取得此圖片的程式矩陣。

三、Arduino 程式碼

```
#include <SPI.h>
#include "epd2in13_v3.h"
#include "imagedata.h"

Epd epd;

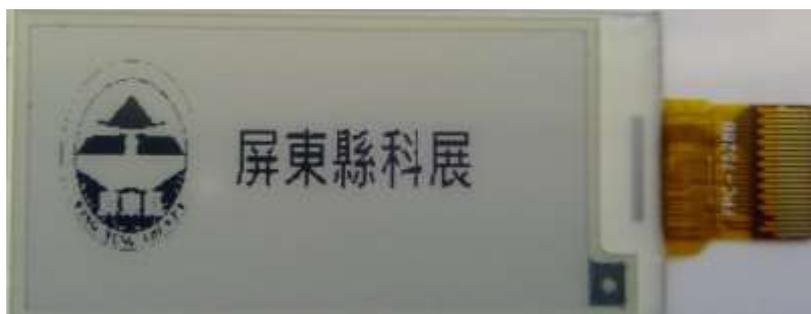
void setup()
{
  // put your setup code here, to run once:
  Serial.begin(115200);
  Serial.println("epd FULL");
  epd.Init(FULL);
  epd.Display(IMAGE_DATA);

  Serial.println("epd PART");
  epd.DisplayPartBaseImage(IMAGE_DATA);
  char i = 0;
  for (i = 0; i < 10; i++) {
    Serial.println("e-Paper PART IMAGE_DATA");
    epd.Init(PART);
    epd.DisplayPart(IMAGE_DATA);
    Serial.println("e-Paper PART Clear");
    epd.Init(PART);
    //epd.ClearPart();
  }

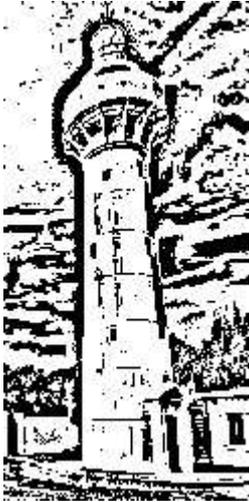
  //epd.Init(FULL);
  //Serial.println("e-Paper clear and sleep");
  //epd.Clear();
  //epd.Sleep();
}

void loop()
{
}
```

四、顯示結果



以上為文字、圖案在電子紙的呈現效果，除了顏色只有黑白，上傳到電子紙後感覺和印刷在紙上效果一樣。接下來，我們嘗試將圖像，分成：建築物以屏東縣鵝鑾鼻燈塔為例，如圖三；人像以先總統蔣中正玉照為例；如圖四、風景照如圖五上傳到電子紙時，須將影像先轉換成單純黑白二值化，發生了下列情形：



圖三



圖四



圖五



圖六

從圖四依舊可以判斷的出先總統蔣公的玉照，但圖三在鵝鑾鼻燈塔的部分，可以看出燈塔的形狀，但天空雲朵的部分在黑白兩色的呈現上不易看出雲層的線條變化。而從網路上所擷取的風景美照(如圖六)，這是一個海上的 Villa 渡假村，黑白效果更不易看出天空和海洋的分界。

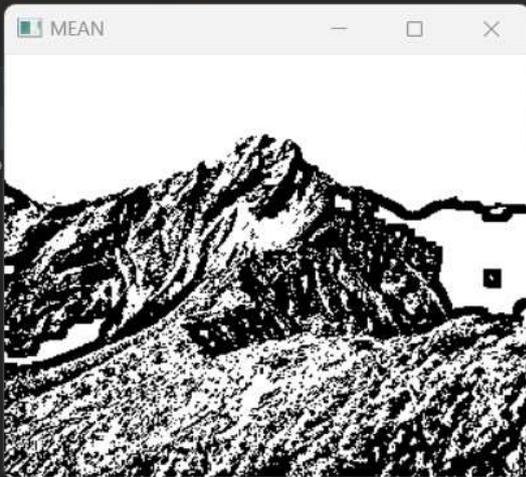
五、運用算術平均法自適應二值化及高斯法自適應二值化加強圖片

上述將圖片轉成單色(黑白化)的方法，只需用小畫家即可。在影像處理稱之為「二值化」，亦稱「閾值二值化」。閾值類似臨界值的概念，將圖片中每個像素的灰度值偵測出後，若該畫素大於預設的閾值，轉成黑色，反之小於則轉成白色，因此圖片最後只有黑跟白兩色。

而對影像處理而言，圖片的每個鄰近像素都彼此有關連，單純二值化就容易造成像上面圖五的狀況，在探討相關文章時，發現到了「自適應二值化」(Threshold)的方法，可以找到利用單一個像素與鄰近區域的關係，使的圖片的判讀性更高。因此接下來欲利用取「區域平均值」的算術平均法自適應二值化，以及取「高斯分布」的高斯加權均值法自適應化二值化(以下簡稱「高斯法」)進一步處理上述圖片。曾經學過一些 Python 語言的我們，老師建議我們透過 Python 使用相當廣泛使用的電腦視覺庫 OpenCV，並結合科學計算的函示庫 NumPy，來達到我們想要讓電子紙呈現黑白圖片效果是否更好的目的。

程式碼與執行結果如下：

```
import numpy as np
import cv2
file_name = "TaiwanMountain.bmp"
origin_img = cv2.imread(file_name, cv2.IMREAD_REDUCED_GRAYSCALE_2)
#開啟照片檔file_name，並改為灰階縮小1/8
ret, th4 = cv2.threshold(origin_img, 127, 255, cv2.THRESH_BINARY)
th5 = cv2.adaptiveThreshold(origin_img, 255, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 11, 2)
th6 = cv2.adaptiveThreshold(origin_img, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 11, 2)
cv2.imshow("123", th4)
cv2.imshow("MEAN", th5)
cv2.imshow("GAUSSIAN", th6)
cv2.imwrite("test4.jpg", th6)
im = cv2.imread("test4.jpg")
gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



圖七



圖八

視窗 MEAN 的圖片為算術平均法所運算的結果，視窗 GAUSSIAN 的圖片為高斯加權平均法運算的結果，我們可以和單純二值化黑白的圖七，以及原始圖圖八做一些對照。

建築物方面，我們也將鵝鑾鼻燈塔的照片，透過此兩種自適應二值化方法轉換，並上傳電子紙，結果如下：



圖九 算術平均法自適應二值化



圖十 高斯加權法自適應二值化

從上面兩個圖片可以觀察到，高斯法讓圖片顯現出更多細節，而算術平均法對於輪廓的呈現較為明顯。而人物方面，以先總統蔣中正的玉照為例，上傳到電子紙結果如下：



圖十一 純黑白化



圖十二 算術平均法



圖十三 高斯加權法

從上面三個圖可以看出，在人像的輪廓方面，算術平均法的表現也較為顯著。

六、利用邊緣檢測對二值化圖片加強

邊緣檢測主要用於偵測物體移動，但礙於電子紙只能顯示黑白兩色效果，因此我也利用此法來測試在電子紙的效果。原理是，先將圖片的物體周圍「膨脹」，再用另外一張將圖片的物體周圍「收縮」，兩張圖片的變化就因為只會發生在邊緣，因此將此兩張圖片用矩陣運算相減，便可看出圖片的邊緣反映。程式碼及圖片在電子紙上的結果如下。

(1)人像部分：

```
1 import numpy as np
2 import cv2
3 file_name = "President.jpg"
4 origin_img = cv2.imread(file_name,cv2.IMREAD_GRAYSCALE)
5 kernel = cv2.getStructuringElement(cv2.MORPH_RECT,(3,3))
6 xxx = cv2.dilate(origin_img, kernel)
7 yyy = cv2.erode(origin_img, kernel)
8 zzz = cv2.absdiff(xxx,yyy)
9 retval, threshold_img = cv2.threshold(zzz, 40, 255, cv2.THRESH_BINARY)
10 result = cv2.bitwise_not(threshold_img)
11 cv2.imshow("President",origin_img)
12 cv2.imshow("dilate_img",xxx)
13 cv2.imshow("erode_img",yyy)
14 cv2.imshow("absdiff_img",zzz)
15 cv2.imshow("threshold_img",threshold_img)
16 cv2.imshow("result",result)
17 cv2.waitKey(0)
18 cv2.destroyAllWindows()
```



圖十四 邊緣檢測後

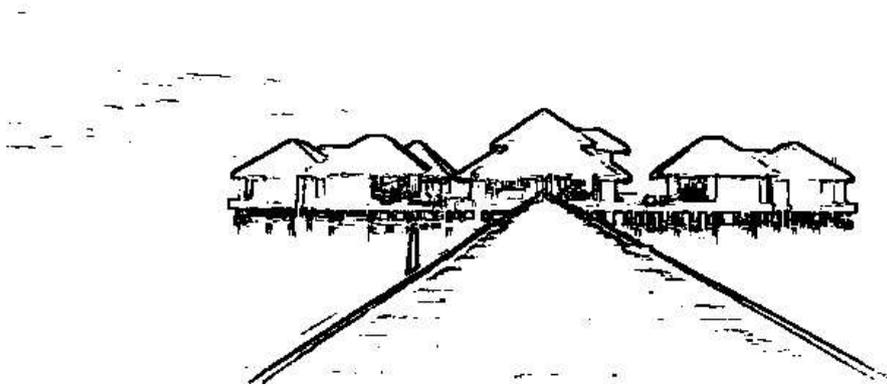


圖十五 將邊緣檢測二值化



圖十六 二值化反轉

(2) 建築部分



圖十七 將邊緣檢測二值化



圖十八 二值化反轉

由於程式碼和上述人像部分無異，在此省略程式碼部分。我們可以發現，邊緣檢測的效果，對於建築物的圖像很適合上傳到電子紙，線條簡單沒有多餘陰影的部分來影線圖片內容的判斷。但原圖是張在海上的建築物，邊緣偵測之後海平面則是完全消失。

(3) 風景部分



圖十九 將邊緣檢測二值化



圖二十 二值化反轉

最後，我們運用 Python 中的 GUI 函示庫 Tkinter，使的程式運算成果和使用者有互動。將之前所使用過的程式添加了幾個相關的指令，使得程式運行後可以將所以在本研究中的二值化圖片結果，以及透過邊緣檢測加強後的結果，讓使用者可以一次性看到這些圖片後，進而能針對想要上傳到電子紙的圖檔儲存起來。程式碼如下：

```

import cv2
import numpy as np
from tkinter import *
from PIL import Image, ImageTk
image = cv2.imread("1120105.jpg",cv2.IMREAD_GRAYSCALE)
kernel = cv2.getStructuringElement(cv2.MORPH_RECT,(3,3))
dilate_img = cv2.dilate(image, kernel)
erode_img = cv2.erode(image, kernel)
absdiff_img = cv2.absdiff(dilate_img, erode_img);
retval, threshold_img = cv2.threshold(absdiff_img, 40, 255, cv2.THRESH_BINARY);
result = cv2.bitwise_not(threshold_img);
output1 = cv2.adaptiveThreshold(image,255,cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY,11,2)
output2 = cv2.adaptiveThreshold(image,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY,11,2)
cv2.imwrite('jianzhu.jpg', image)
cv2.imwrite('dilate_img.jpg', dilate_img)
cv2.imwrite('erode_img.jpg', erode_img)
cv2.imwrite('absdiff_img.jpg', absdiff_img)
cv2.imwrite('threshold_img.jpg', threshold_img)
cv2.imwrite('result.jpg', result)

```

```

cv2.imwrite('result.jpg', result)
cv2.imwrite("test1.jpg",output1)
cv2.imwrite("test2.jpg",output2)
root = Tk()
root.title("ch17_29")

img = Image.open("1120105.jpg")
img2 = Image.open("jianzhu.jpg")
img3 = Image.open("dilate_img.jpg")
img4 = Image.open("erode_img.jpg")
img5 = Image.open("absdiff_img.jpg")
img6 = Image.open("threshold_img.jpg")
img7 = Image.open("result.jpg")
img8 = Image.open("test1.jpg")
img9 = Image.open("test2.jpg")
myPhoto = ImageTk.PhotoImage(img)
myPhoto2 = ImageTk.PhotoImage(img2)
myPhoto3 = ImageTk.PhotoImage(img3)
myPhoto4 = ImageTk.PhotoImage(img4)
myPhoto5 = ImageTk.PhotoImage(img5)
myPhoto6 = ImageTk.PhotoImage(img6)
myPhoto7 = ImageTk.PhotoImage(img7)
myPhoto8 = ImageTk.PhotoImage(img8)
myPhoto9 = ImageTk.PhotoImage(img9)

```

```

text = Text()
text.image_create(END,image=myPhoto)
text.insert(END,"\n")
text.insert(END,"原始圖片")
text.image_create(END,image=myPhoto2)
text.insert(END,"\n")
text.insert(END,"jianzhu")
text.image_create(END,image=myPhoto3)
text.insert(END,"\n")
text.insert(END,"dilate_img")
text.image_create(END,image=myPhoto4)
text.insert(END,"\n")
text.insert(END,"erode_img")
text.image_create(END,image=myPhoto5)
text.insert(END,"\n")
text.insert(END,"absdiff_img")
text.image_create(END,image=myPhoto6)
text.insert(END,"\n")
text.insert(END,"threshold_img")
text.image_create(END,image=myPhoto7)
text.insert(END,"\n")
text.insert(END,"result.jpg")
text.image_create(END,image=myPhoto8)
text.insert(END,"\n")
text.insert(END,"test1.jpg")
text.image_create(END,image=myPhoto9)
text.insert(END,"\n")
text.insert(END,"test2.jpg")
text.pack(fill=BOTH,expand=True)

root.mainloop()

```

註：程式碼接續右張圖片

伍、討論

一、硬體方面：

本研究所採用的黑白二色電子紙 2.13 吋的市價約 280 元，礙於黑白二色的顯示效果，若要取代紙張，建議只適合用來顯示文字或是單一顏色且線條明顯的圖案。而 5.83 吋的電子紙雖尺寸較大，市價來到 1400 元，上傳程式碼後直到電子紙顯示出圖像，過程花了約 15 秒。而若要改為無線方式(如：wifi、NFC)來刷新電子紙顯示，此款電子紙 2.13 吋的就需要 850 元，價差來到了三倍。當然在未來若要廣泛使用電子紙取代，還是無線傳輸較為方便。

二、軟體方面

由於黑白二色的顯示限制，造成很多影像刷新到電子紙之後，看不到光影、深淺、甚至是線條的變化，透過 OpenCV 的影像處理運算後，再上傳到電子紙，大部分圖片確實得到了加強的效果。特別是「算術平均法自適應二值化」對於建築物、人像、風景都有把輪廓線條加粗，對於部分解析度不高的電子紙顯示效果遠比純黑白二值化好。

陸、結論

在我們研究電子紙的同時，國內電子紙廠商元太科技不斷推陳出新，且市面上已有販售可顯示灰階的電子紙，或是黑白黃、黑白紅等三色的電子紙，取代紙張的效果在顯示色彩上，必超越在這個研究中所採用的黑白電子紙。

當然黑白電子紙仍價格較為低廉或是較容易買到，未來若有機會，我們想討論開發一個讓使用者較為方便的程式，直接讓使用者一鍵便可上傳電子紙要顯示的影像。換句話說，將本研究中所有二值化效果透過預覽，讓使用者選擇需要的結果，在此程式中直接產出 Arduino 程式碼，進而傳輸給開發版，而上傳到電子紙上。

柒、參考資料

1. 屏東縣政府，*屏東縣縣徽*。 <https://www.pthg.gov.tw/planra/Default.aspx>
2. Liying Li, 唯美城市風景高畫質動態壁紙 <https://www.pinterest.com/pin/720435271636739311/>
3. 中華民國總統府，*先總統蔣中正玉照*。 <https://www.president.gov.tw/Page/83>
4. 玉山國家公園，*玉山主峰寫真*。 <https://www.ysnp.gov.tw/Album/C006000>
5. 嗡嗡的隨手筆記，OpenCV 的圖片自適應二值化，產生更好效果的黑白圖片
https://www.wongwonggoods.com/python/python_opencv/opencv-adaptivethreshold/
6. 張德豐(2022)。一本書秒殺電腦視覺最新應用 80 個 Python 大師級實例。台北市：深智數位股份有限公司。