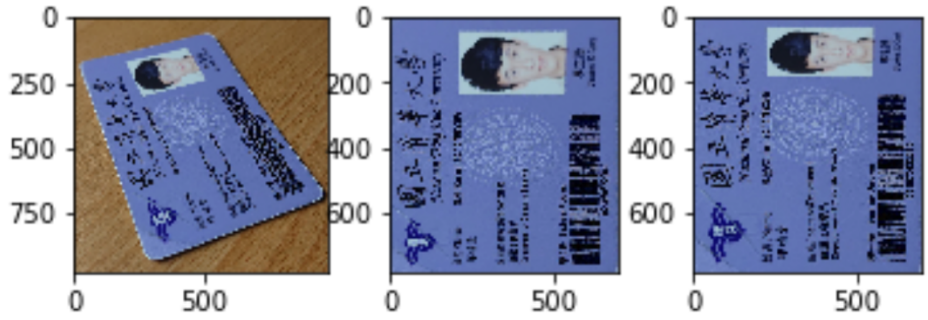
Assignment 2

107062208 邱靖豪

第一組圖片：

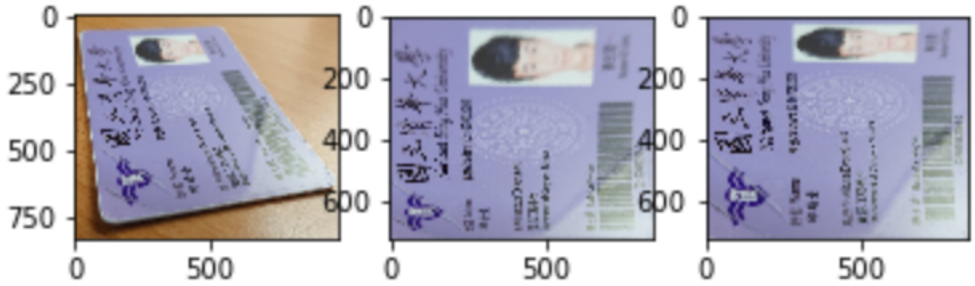
R matrix & Picture 1:

```
[[ 7.67831970e-01  2.19260510e-01  2.90000000e+01]
 [-1.53378981e-01  5.70527140e-01  1.81000000e+02]
 [ 1.34540023e-04 -4.06829447e-04  1.00000000e+00]]
```



R matrix & Picture 2:

```
[[ 6.62983337e-01  5.72765428e-02  2.90000000e+01]
 [-1.25491462e-02  4.91415610e-01  6.70000000e+01]
 [ 1.23897614e-04 -6.31266634e-04  1.00000000e+00]]
```



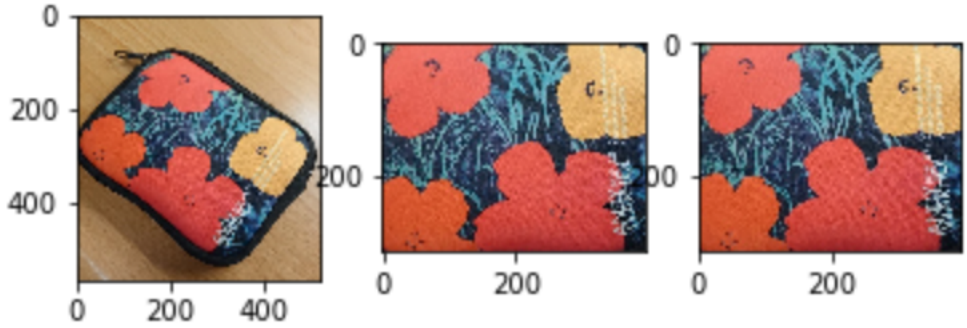
我測試的第一組圖片，是我的學生證。首先討論兩種由不同角度所拍攝的照片經運算轉移後得到的結果。第一張圖拍攝時與桌面夾角比較大，拍攝的結果看來，圖面上的文字比較清楚（雖然因為畫質的關係，加上學生證上的文字比較多，導致文字本身起初就較模糊），所以轉移後的圖片算滿清楚的。而下圖拍攝時與桌面的夾角比較小，拍攝方向也較為偏移，等於是將圖片的內容壓縮在一個更扁平的範圍裡，所以轉移後的圖片也就沒那麼清楚。所以，推測拍攝角度若越接近俯視，轉換效果越好。

接著討論兩種不同演算法，造成的不同結果。我發現，在第一組圖片中，用第一種演算法的結果，似乎都較符合原圖的樣式，因為在第二種演算法中，不論是第一張還是第二張圖，人像的地方都有明顯被壓縮的感覺，條碼的部分，也總是比第一種演算法做的還長。但是，如果從畫質的角度來看，我覺得兩者是差不多的。

第二組圖片：

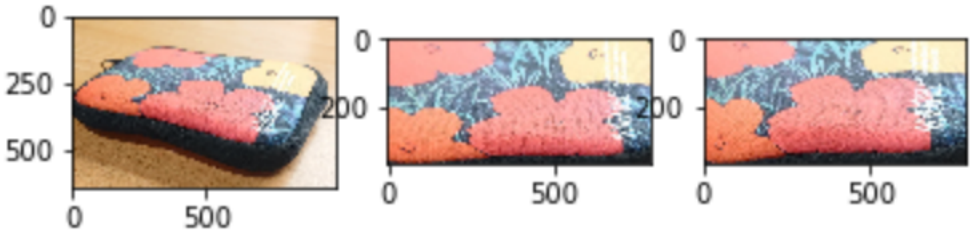
R matrix & Picture 1:

```
[[ 6.43607594e-01 -6.19513523e-01  1.97000000e+02]
 [ 4.51324710e-01  5.95591829e-01  7.40000000e+01]
 [-2.55421178e-04 -2.30944932e-04  1.00000000e+00]]
```



R matrix & Picture 2:

```
[[ 4.98384742e-01 -8.48956687e-01  3.21000000e+02]
 [ 5.19723236e-02  3.15481089e-01  1.25000000e+02]
 [-2.82298532e-04 -6.95809370e-04  1.00000000e+00]]
```



我測試的第二組圖片，是我的錢包。這兩張圖我也是分別從不同的角度去拍攝，其中第一張照片已經幾乎接近俯視，所以效果很好，而第二張的話，因為與桌面夾角很小，因此經過翻轉的結果幾乎是扭曲狹長的。所以證實我們第一題的推論，越接近俯視的角度，效果越好。

接著討論兩種不同演算法，造成的不同結果。我發現，在這組圖片中，兩個演算法的差異並沒有第一種那麼大，有可能是因為沒什麼密集的文字或明顯的圖像，方便我們觀察比對。

最後，根據這兩張圖片的結果，我的結論是，如果拍攝的角度若越接近垂直俯視於桌面，還原出來的圖片會越接近原本圖片的樣子。因為當拍攝角度越垂直於桌面，(x , y)對應點會越接近原圖點，因此也能提供較精準的對應結果。反之，若拍攝角度越平行，則代表(x , y)點跟原圖點比越為扭曲，結果就會偏差。

(在 code 裡面有四份不同圖片能跑的結果，但是考慮到結果的可看性及畫質，我把其他三種註解掉)