

# 影像處理與機器人視覺 HW2

姓名：吳建澄

學號：NM6111035

# 影像處理與機器人視覺 HW2

## 1. 調整亮度:

### 1-1. 增強暗部

Read origin data's header.

```
In [115]: import math
          tmp = []
          inttmp = []
          log = []
          c = 30

In [116]: with open("logarithm.bmp", "rb") as f:
          byte = f.read(1)
          while byte != b"":
              # Do stuff with byte.
              tmp.append(byte)
              byte = f.read(1)
```

Converter bytes to int, then using inner function to compute logarithm.

```
In [118]: for i in range(len(tmp)):
          inttmp.append(int.from_bytes(tmp[i], "big"))

In [119]: for i in range(1078, len(inttmp)):
          if(inttmp[i] != 0):
              log.append(int(c * math.log(1 + inttmp[i] )))
          else:
              log.append(0)
```

After calculating, write a new file with origin header & computed data.

```
In [121]: res = open("log.bmp", "wb+")

In [122]: res.write(bytes(inttmp[0:1078]))
          res.write(bytes(log))

Out[122]: 240000

In [123]: res.close

Out[123]: <function BufferedRandom.close>
```

## 1-2. 增強亮部

Read origin data's header.

```
In [31]: import math
        tmp = []
        inttmp = []
        c = 255
        r = 1.8

In [32]: with open("power law.bmp", "rb") as f:
        byte = f.read(1)
        while byte != b"":
            # Do stuff with byte.
            tmp.append(byte)
            byte = f.read(1)
```

Converter bytes to int, then using inner function to do power-law.

```
In [33]: for i in range(len(tmp)):
        inttmp.append((int.from_bytes(tmp[i], "big")))

In [34]: power = []
        powerint = []

In [35]: for i in range(1078, len(inttmp)):
        power.append(c * pow(float(inttmp[i]) / 255, r))

In [36]: for i in range(len(power)):
        powerint.append(int(power[i]))
```

After calculating, write a new file with origin header & computed data.

```
In [39]: res = open("power.bmp", "wb+")

In [40]: res.write(bytes(inttmp[0:1078]))
        res.write(bytes(powerint))

Out[40]: 169472

In [41]: res.close

Out[41]: <function BufferedRandom.close>
```

### 3. 直方圖等化

Read origin data's header, converter bytes to int.

```
In [1]: import math
        tmp = []
        inttmp = []
        total = []
        prob = []
        probsum = []
        histo = []

In [2]: with open("HistogramEQ.bmp", "rb") as f:
        byte = f.read(1)
        while byte != b"":
            # Do stuff with byte.
            tmp.append(byte)
            byte = f.read(1)

In [3]: for i in range(len(tmp)):
        inttmp.append(int.from_bytes(tmp[i], "big"))
```

To sum the number of pixel value

```
In [4]: for i in range(256):
        total.append(0)

In [5]: for i in range(1078, len(inttmp)):
        total[inttmp[i]] += 1
```

To compute each pixel values appear probability, and cumulative probability.

```
In [6]: for i in range(256):
        prob.append(float(total[i]) / (len(inttmp)-1078))

In [7]: sum = 0.0
        for i in range(256):
            probsum.append(sum + prob[i])
            sum += prob[i]
```

Equalization value = pixel values cumulative probability \* max(pixel values)

```
In [10]: for i in range(1078, len(inttmp)):
        histo.append(int(probsum[inttmp[i]] * 255) )

In [36]: for i in range(len(power)):
        powerint.append(int(power[i]))
```

After calculating, write a new file with origin header & computed data.

---

```
In [12]: res = open("histo.bmp", "wb+")
```

```
In [13]: res.write(bytes(inttmp[0:1078]))  
res.write(bytes(histo))
```

```
Out[13]: 700416
```

```
In [14]: res.close
```

```
Out[14]: <function BufferedRandom.close>
```

#### 4. 遇到的問題&如何解決

In slides, the power-law & logarithms pixel value is between 0~1, but bmt value is 0~255 so I normalize the bmt value by divide 255.

There is no other problem.