

10 Feature extraction

Key terms: PGM histogram

Reading: Wikipedia's "Netpbm format"²⁴

Reading: MoreStream's "Histogram"²⁵

Exercise: Write a program that indicates the dimensions of any PGM files whose pathnames are passed as arguments.

At the foundation of information processing is the translation of an input document into an output document. The general function $y = f(x)$ can encompass not just integer operations, but all kinds of processes. Strategic management of the creation and conversion of documents has been named "document engineering". There is more uncertainty and more degrees of freedom in anglicizing Bashō's haikus than in compiling a computer program.

Feature extraction for machine learning is an important sort of document description. The relative quantities of pixels of different brightness is a key basic feature in image processing applications. Here, we extract (f) coloration features (y) of ASCII PGM images (x). The PGM (Portable Gray Map) format, whose ASCII ("plain") form is human-readable, supports greyscale in different bit depths, indicated after the dimensions. The format is simple but the cost of that simplicity is space inefficiency, even compared to other lossless formats. The file consists of a header and then a long series of integer pixel values separated by any combination of spaces and newlines.²⁶

You can find a sample greyscale image in `/users/abrick/resources/maquinna.pgm`. To get you started, here is a code snippet that can load a PGM file; you have permission to reuse this code if it helps you.

```
# Process and input the integer elements of the ASCII PGM file.
with open(sys.argv[1]) as content:
    parts = re.split(r'\s+',re.sub(r'#.*',r'\n',content.read()))
    x_dim, y_dim, depth = int(parts[1]), int(parts[2]), int(parts[3])
    pixels = [int(n) for n in parts[4:] if n]
    assert len(pixels) == x_dim * y_dim
```

And here is an example of PGM creation:

```
# Transform random bits into a square PGM image.
import random, sys
dimension, depth = 500, 255
greys = range(depth)
with open('output.pgm','w') as target:
    header, pixels = 'P2\n{0} {0}\n{1}\n'.format(dimension,depth), ''
    for row in range(dimension):
        for column in range(dimension):
            pixels += str(random.choice(greys)) + ' '
        pixels += '\n'
    target.write(header+pixels)
```

Next week, we will address the matter of web services.

²⁴https://en.wikipedia.org/wiki/Netpbm_format

²⁵<https://www.moresteam.com/toolbox/histogram.cfm>

²⁶<http://netpbm.sourceforge.net/doc/pgm.html>