Entropy-Based Image Descriminator

J.S. Virzi

December 31, 2022

1 Abstract

We present a computationally efficient descriminator helpful for finding areas of localized brightness inside a larger image. Images with clumps of bright pixels on a dim background generally have lower entropy, as compared to a homogenous image with the same overall brightness. The lower entropy relates to the higher amount of structure when few pixels contain a disproportionate amount of brightness. An algorithm of $O\left(N\right)$ is presented where the complexity of intermediate quantities are abstracted away using lookup tables. Arithmetic complexity is reduced to sums and multiplications. Iteration of the algorithm at different scales can be employed to find localized structure within a system. The algorithm is generalized to any system with countable states.

1.1 Background

Entropy measures the number of states available to a system.

For example, consider a gray scale image consisting of $N = W \times H$ pixels, where W is the width of the image and H denotes its height. For convenience, assume each pixel is represented by 8 bit integer, encoding 256 different values. A completely dark pixel is represented by 0; the brightest value of a pixel is 255. Let n_k represent the brightness of the k^{th} pixel. The maximum entropy of this image is defined as the logarithm of the number of available states, assuming each pixel can assume any of 256 values ($0 \leq n_k \leq 255$).

$$S_{max} = \log\left(256\right)^{W \times H} = N \times \log\left(256\right) \tag{1}$$

The total brightness of the image can be considered as the total energy $n \equiv E_{TOT}$ defined as

$$n \equiv \sum_{k=1}^{k=N} n_k \tag{2}$$

The total number of states available to this image is

$$\sigma = \frac{\left(n_1 + n_2 + n_3 + \ldots + n_N\right)!}{n_1!n_2!n_3!\ldots n_N!} = \frac{n!}{n_1!n_2!n_3!\ldots n_N!} \tag{3}$$

The entropy of this image is

$$S \equiv \log\left(\sigma\right) = \log\left(\frac{n!}{n_1!n_2!n_3!\ldots n_N!}\right) \tag{4}$$

Whereas Eqn.(**??**) assumes the form of the multinomial distribution, the maximum value is assumed when all factors are equal. From this, we know the maximum entropy of this image is assumed when all pixels have the same value, namely the average energy $\bar{n} \equiv \frac{n}{N}$. Under these conditions, the entropy is

$$S_0 \equiv \log\left(\sigma_0\right) = \log\left(\frac{n!}{\left(\bar{n}!\right)^N}\right) \tag{5}$$

We use Eqn.(5) to give a reference scale to Eqn.(4)

$$\delta \equiv \frac{\sigma}{\sigma_0} = \frac{\left(\bar{n}!\right)^N}{n_1!n_2!n_3!\ldots n_N!} \tag{6}$$

δ is a ratio between 0 and 1, comparing the number of available states of the image with energy n to the maximum number of states. An image with random values of energy (uniform brightness) will have a large entropy, because there are many, many ways to create such an image. An image with swaths of brightness will have a low entropy, reflecting the much fewer ways to construct such an image. We investigate the power of this entropy measure to discern if an image might contain otherwise unnaturally clumped swaths of brightness. It proves convenient to work with the logarithms of these large numbers

$$\Delta \equiv \log\left(\frac{\sigma}{\sigma_0}\right) = N * \log \bar{n}! - \log n_1! - \log n_2! - \log n_3! - \ldots - \log n_N! \tag{7}$$

2 Δ as Descriminator

Δ

3 Further Development - Clump Detection

Δ gives the same value regardless of the distribution of bright pixels. The pixels could be clumped together, or distributed in any manner across the image, and still give the same value of Δ. By iterating the algorithm on different scales, effectively grouping pixels together, any clumped structure begins to emerge. For example, suppose an 100×100 image contains a 10×10 clump of brightness in the upper left-hand quadrant. The upper right-hand quadrant, and both lower quadrants will contain virtually no structure and high entropy. Only the upper left-hand quadrant will contain structure, and thus low entropy.

4 Computational Efficiency

For this example, we have 256 different energies for each pixel. Computation of Δ in Eqn. (7) can be efficiently calculated using a lookup table, containing 256 entries. The k^{th} entry of the lookup table contains the value $\lambda_k \equiv \log k!$. Calculation of Δ involves a sum over pixels.

5 Generalization of Algorithm

For the sake of readability, we focused on a particular example. The methods developed herein are extended, in a straight-forward and self-explanatory manner, to any system with countable states.

6 Conclusions

We have developed a descriminator Δ which can be used as a tool to find localized clumps of structure within a larger system. Conversely, Δ can be used to rule out localized structure in an image, or subimage.