תמונה שמכילה טקסט, צילום מסך, גופן, מספר

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

The line is of size , and so representing a position in that line requires bits. Furthermore, the length of each run also requires bits(since a run could take up to the entire line). And so, each run takes bits. The start of the line is signaled by the pair , which would also require bits.

We will mark the number of runs with .

And so we can conclude that : The first coming from the start of the line, coming from the number of runs in the line, and being the upper bound of runs(as the image is of size ).

Simplifying, we get that

Plugging in to our equation, we get that

תמונה שמכילה טקסט, צילום מסך, גופן

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

The ideal bit-per-pixel ratio would be 5.3 bits/pixel. Assuming the original image uses 8-bit pixels(grayscale), we get that the maximum compression would be

Meaning that the image may be compressed to about of its original size.

Huffman coding is optimal for memoryless sources, reaching the entropy limit when the symbol probabilities are powers of . Since many(if not most) pixel probabilities are not powers of , the resulting average would be more than the entropy.

We could also consider the fact that if the probability tree isn’t shared beforehand, we would also need to transmit the tree itself- adding additional overhead to the encoding.

We could preprocess the image to remove redundancies- either reducing spatial redundancy or removing information that is irrelevant to the image.

If we would like to not do any preprocessing on the image, we could change the encoding method altogether, to a coding method that is more efficient, such as arithmetic encoding.