Lossless Image Compression through Super-Resolution

Autor: Baptista, Ernie , Carlos Leonardo Jireh Ferroa Guzmán

***Abstract*— En el presente articulo se presenta un algoritmo de comprensión de imagen sin perdida de calidad. Se almacena una imagen en baja resolución de formato RAW, para así continuar con iteraciones de SR super resolución sin pérdida de calidad. El proceso se basa en predecir la probabilidad de una imagen de alta resolución basándose en la imagen de alta resolución ya almacenada. Finalmente, se usa codificación entrópica para comprimir la operación de SR (super resolution).**

I. Introducción

Las imagenes capturan los aspectos de nuestro mundo visual que tiene diversidad y complejidad en cuanto a lo visual. Por otro lado, no todos los pixeles de color forman una imagen real. Este es la principal percepcion que se tiene de la compresion de imagenes sin perdida de calidad. En este articulo se predice una distribución en todas las posibles soluciones para reconstruir imagenes. Cada pixel en baja resolucion induce a una distribucion aleatoria entre cuatro pixeles de alta resolucion que seran mostrados. El algoritmo guarda una version de baja resolucion de la imagen como pixeles en formato RAW, luego aplica iteraciones de perdida minima de calidad en las compresiones. En el presente articulo, se explicara el proceso que fue diseñar un modelo probabilistico de valores de pixeles.

II. Marco Teórico

A. Lossless Image Compression

Es un tipo de compresion de datos en el cual el tamaño de cualquier imagen es reducido sin perder la calidad de esta. Para lograr esta comprension algunos metodos serán usados:

1) Adaptive dictionary algorithms : Estos algoritmos reducen archivos mediante busqueda de elementos similares que coincidan con los datos almacenados en los diccionarios. Cuando el metodo de comprension es utilizado, se comparan datos del archivo con datos almacenados en el diccionario, despues reemplaza valores en la imagen con los encontrados en el diccionario.

2) Entropy Encoding: Este metodo comprime imagenes mediante la busqueda de parametros similares y reemplazando los que se presentan de manera frecuente en la imagen, mediante codigos que son menores en tamaño.

3) Run length encoding: Los datos muchas veces contienen secuencias que se repiten en un mismo archivo. Este metodo busca esas secuencias repetitivas y las almacena como un contador representado por las veces que se repiten y su valor.

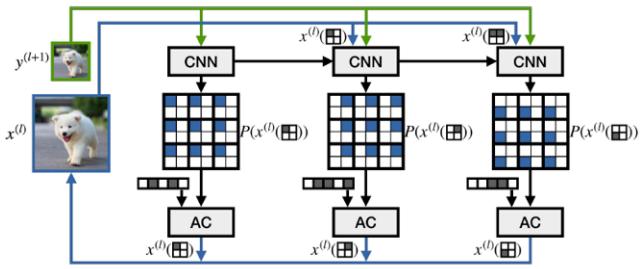
III. Estado del Arte

El método presentado aprovecha avances recientes con respecto al tema de Super Resolucion (RS). El algoritmo predice una probabilidad para cada imagen de alta resolución para la entropía1 codifica la imagen sin pérdidas, enfocándose más en la entropía codificada de cada imagen de forma independiente a una tasa de bits caerca de los mejores enfoques.El algoritmo condicionalmente codifica la imagen de mayor resolución “X(1)” dada una imagen de menor resolución “Y(l+1) usando AC basado en la probabilidad estimada por una red de súper resolución. A través de 4 pixeles, 3 alterados y 1 sacado con fórmula, se aprovecha la correlación entre estos 4 pixeles, factorizando su probabilidad autorregresiva. Ello posee muchas ventajas. Una de las cuales es que usa una estructura jerárquica con algunas diferencias, debido a que el tiempo de ejecución es O(log(W+H)), para una estructura jerárquica normal, donde W y H son el tamaño de la imagen, pero se utilizan PixelCNN poco profundos, siendo esta práctica aún más lenta. Por ello, el modelo utiliza una factorización más simple para dar las probabilidades de los pixeles, el cual permite, en diferentes niveles jerárquicos, compartir una función de incrustación y uso de una arquitectura más eficiente. Esto genera que el modelo final sea, aproximadamente, 60 veces

más rápido que otros modelos como el Reed et al. Además, es, aproximadamente, 55 veces más rápido en imágenes de alta resolución. Por otro lado, todos los pases de red y cálculos de parámetros se realizan en paralelo durante toda la imagen, lo que permite una decodificación eficiente en una GPU. Por otro lado, una desventaja de este algoritmo es que la decodificación es marginalmente más lenta que la codificación, debido a que obliga a varias sincronizaciones de GPU.

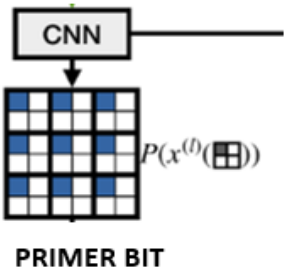
IV. Metodología

figure[H]



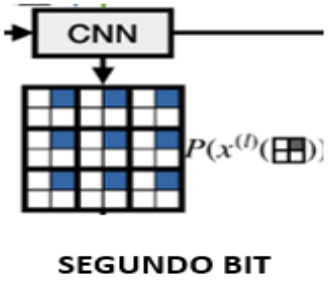
En primer lugar, el algoritmo codifica el nivel de resolución más bajo como pixeles sin formato "y(l+1)", que es el muestreo inferior. Eso, luego almacena los bits de redondeo para construir un "X(l)" y posteriormente codifica los códigos aritméticos de la red de Súper Resolución y los bits de redondeo para todos los niveles consecutivos. Este redondeo codifica como dos bits por canal de color correspondiente a los 4 valores de redondeo dados: $\frac{-1}{4}, 0, \frac{1}{4}, \frac{1}{2}$

Siempre el primer bit, como referencia en la Fig1, es, del cuadrado, la superior izquierda, siendo un X2i,2y



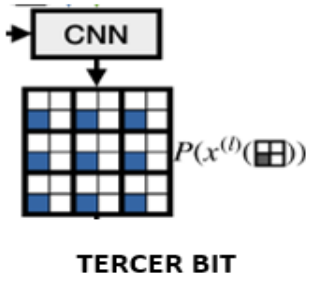
figure[H]

Por consiguiente, la superior derecha, siendo este X2i,2y+1



figure[H]

Y finalmente el inferior izquierdo, siendo este, X2i+1,2y



figure[H]

Para el bit faltante, que sería el inferior derecha, no es necesario calcular debido a que será reconstruido de forma gratuita utilizando la fórmula.

$$x_{2i+1,2j+1} = 4y_{i,j} - x_{2i,2j} - x_{2i,2j+1} - x_{2i+1,2j}.$$

Ahora, para la mezcla de colores, los primeros componentes realizan ello son verde y azul, ellos dependen de los valores previamente codificados y deben ser apareados en ese orden. Cuando el primer bit o pixel es codificado, continua el segundo pixel da paso a la red y produce la nueva mezcla, decodificándose de forma análoga al primero. Posteriormente, pasa el tercero y último para su tercer valor, debido a que el cuarto, como está especificado anteriormente, es sacado por fórmula matemática. Pero, ¿qué sucede cuando el ancho y alto de una imagen no es divisible entre 2? Cuando la imagen tiene un número impar de columnas o filas, en el momento de la descompresión se descarta una columna más a la derecha o la parte inferior si hablamos de filas, para mantener el tamaño original. Para poder realizar los experimentos respectivos con este algoritmo se evaluaron en dos conjuntos de datos, ImageNet64 de 64x64 e imágenes abiertas como JPEG y PNG. Para una comparación justa se redujeron las imágenes a 768 pixeles en el lado

más largo. Se midió el tiempo que demora en correr el algoritmo en una máquina con AMD Ryzen 5 1600 y NVIDIA GTX 1060.

	time (s)	%
$x^{(3)}$	0.0002	0.01
$x^{(2)}$	0.077	6.7
$x^{(1)}$	0.230	20.0
$x^{(0)}$	0.842	73.3
Total	1.149	

(a) **Speed.**

	bpsp	%
rounding bits	0.656	24.3
metadata	0.002	0.1
$x^{(3)}$ (raw image)	0.125	4.6
$x^{(2)}$	0.126	4.7
$x^{(1)}$	0.432	16.0
$x^{(0)}$	1.360	50.3
Total	2.701	

(b) **Compression Rate.**

W×H	enc (s)	dec (s)
32^2	0.036	0.049
64^2	0.045	0.085
320^2	0.277	0.327
640^2	0.977	1.101
720^2	1.166	1.549
960^2	2.148	2.373

(c) **Scalability.**

Fig. 1. La figura presenta el análisis de la velocidad y la tasa de compresión en las tablas b y c. La tabla de se encuentra en bits por subpixel (bpsp).

V. Resultados

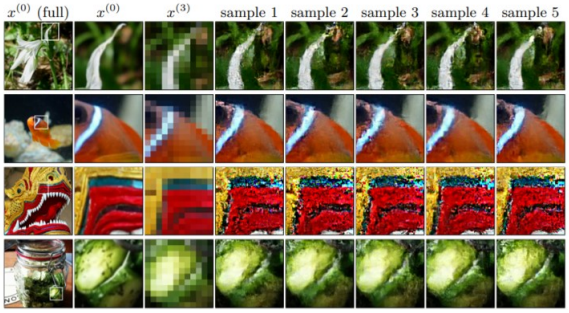


Fig. 2. La distribución de SR (super resolution), muestra un recorte de una parte de la imagen x^0, almacena esa imagen en baja resolucion en formato RAW x^3, luego hace las iteraciones de compresion con perdida de calidad minima sample 1,2,3,4,5.

Para el proceso iterativo se usó SR + factorización a unos 2.69 bpsp (bits per subpixel). Se evaluaron los tiempos de respuestas (a). Compression Rate compara la verosimilitud de diferentes niveles de SR. Cada nivel adicional mejora la performance de saturación en tres niveles. Scalability muestra los tiempos de respuesta con código aritmético cuando se escala en imágenes de alta resolución. Se muestra que en imágenes de 960x960 los tiempos de encoding y decoding se muestran similares. Encoding será más eficiente que decoding, ya que requiere varias sincronizaciones del CPU-GPU.

VI. Conclusiones

Habiendo realizado las respectivas pruebas, se muestra una súper resolución sin perdidas, siendo los operadores eficientes para almacenar gracias a las limitaciones naturales inducidas por el ajuste a super resolución. En conclusión, el método es simple y eficiente, siendo, el modelo de ejecución, igual o más rápido que otros, los cuales están basados en redes profundas de métodos de comprensión.

References

[1] C. Mack. "How to Write a Good Scientific Paper", Published by SPIE, Bellingham, Washington, 2018,

[2] A. Armani. "10 Simple Steps to Writing a Scientific Paper". Published by SPIE, 2020. https://spie.org/news/photonics-focus/janfeb-2020/how-to-write-a-scientific-paper

[3] K. Korpela. "The Art of Data Visualization: A Gift or a Skill?". ISACA Journal, 2016. URL: https://www.isaca.org/resources/isaca-journal/issues/2016/volume-1/the-art-of-data-visualization-a-gift-or-a-skill-