

ALGORITMOS DE COMPRESION DE IMÁGENES PARA OPTIMIZACIÓN DE PROCESOS EN GANADERÍA DE PRECISIÓN

Isabella Echeverri Villa
Universidad EAFIT
Colombia
iecheverrv@eafit.edu.do

Samuel Arturo Flórez Rincón
Universidad EAFIT
Colombia
saflorezr@eafit.edu.co

Laura Echavarría Peláez
Universidad EAFIT
Colombia
lechavarrp@eafit.edu.co

Mauricio Toro
Universidad EAFIT
Colombia
mtorobe@eafit.edu.co

RESUMEN

El proceso de incorporar las tecnologías de la información en el campo de la ganadería ha dado lugar a la llamada Ganadería de Precisión (GdP). Este problema cobra importancia ante la necesidad de reducir los recursos utilizados en el almacenamiento, envío y subida a la nube de las imágenes de monitoreo tomadas en las instalaciones. Un problema relacionado es optimizar el almacenamiento, recibo y envío de cantidades masivas de datos desde dispositivos móviles no especializados como teléfonos de mediana gama.

Este informe hace un análisis de diferentes soluciones implementadas y propone dos algoritmos de compresión y descompresión de imágenes. El primer algoritmo implementa la estructura de *Singular Value Decomposition* (con pérdidas), y el segundo parte del *Algoritmo de Huffman* (sin pérdidas). Ambos algoritmos prueban ser efectivos en la implementación y son un punto de partida adecuado, sin embargo la complejidad temporal y espacial de estos resalta la necesidad de, en futuros estudios, optimizar los algoritmos para lograr una mayor eficiencia en el uso de los recursos disponibles.

Palabras clave

Algoritmos de compresión, aprendizaje de máquina, aprendizaje profundo, ganadería de precisión, salud animal.

1. INTRODUCCIÓN

Explica la motivación, en el mundo real, que lleva al problema. Incluyan la historia de este problema. (En este semestre, la motivación es la razón por la que necesitamos comprimir imágenes para clasificar la salud animal en el contexto de la ganadería de precisión).

1.1. Problema

La industria ganadera invierte muchos recursos en el cuidado del ganado, puesto que hasta el momento ha sido una tarea asignada completamente a un ser humano, sin ayuda tecnológica alguna. Teniendo en cuenta que el ganado enfermo tiene una apariencia similar, la cual podría ser reconocida por una máquina, es muy acertado pensar en un algoritmo que realice esta distinción.

Debido a la baja calidad de las redes en las zonas campestres, donde se encuentra ubicado el ganado, es necesario desarrollar una estructura de datos que comprima y descomprima las imágenes que el algoritmo posteriormente

clasificara como ganado sano o enfermo. En este trabajo se propone un algoritmo que identifique mediante imágenes que parte del ganado se encuentra en buen estado de salud y que parte no.

Este algoritmo propone una solución muy eficiente, que facilita el cuidado y la preservación del ganado.

1.2 Solución

En este trabajo, utilizamos una red neuronal convolucional para clasificar la salud animal, en el ganado vacuno, en el contexto de la ganadería de precisión (GdP). Un problema común en la GdP es que la infraestructura de la red es muy limitada, por lo que se requiere la compresión de los datos.

En este trabajo se pretende utilizar una red neuronal de clasificación para facilitar y hacer más eficiente el cuidado del ganado, sin embargo, para la realización de esto necesitamos dar solución al problema de las redes limitadas en el sector de la ganadería de precisión. Por lo que optamos por crear algoritmos para comprimir y descomprimir las imágenes que, posteriormente, serían clasificadas como ganado sano o enfermo. Para esta compresión, utilizamos algoritmos con y sin pérdidas. En el algoritmo con pérdidas utilizamos el algoritmo de “Descomposición del valor singular”, más conocido como “Singular Value Decomposition” o SVD. Por otra parte, en el algoritmo sin pérdidas optamos por el “Algoritmo de Huffman”.

1.3 Estructura del artículo

A continuación, en la Sección 2, describiremos algunos trabajos relacionados con el procesamiento de imágenes para el beneficio del sector agroindustrial. Posteriormente, en la Sección 3, mostraremos a grandes rasgos los métodos de los que nos valimos para desarrollar el algoritmo.

Para este problema, hemos analizado que el algoritmo de Huffman es una opción muy viable para realizar la compresión de las imágenes, puesto que, además de ser un algoritmo sin pérdidas, involucra la utilización de árboles, y creemos que los árboles ayudan a organizar el algoritmo de una manera relativamente eficiente. Por otra parte, este algoritmo fue de fácil comprensión para nosotros, por lo que nuestro entendimiento de la importancia de las estructuras de datos fue mayor.

2. TRABAJOS RELACIONADOS

En lo que sigue, explicamos algunos trabajos relacionados en el dominio de la clasificación de la salud animal y la

compresión de datos en el contexto del PLF. En cada uno de estos trabajos se muestra la variedad de aplicaciones que se pueden realizar con diferentes tecnologías, cada una de ellas fundamentada en algoritmos y estructuras de datos. Aunque algunos de estos problemas no están diseñados especialmente para la industria de la ganadería de precisión, podrían ser de gran utilidad.

2.1 Colección automatizada y análisis de termogramas infrarrojos para la medición de la temperatura del ganado.

Desarrollaron un algoritmo que, con ayuda de termografía infrarroja, monitorea al ganado. Cuando llevan mucho tiempo sin moverse o la temperatura es muy alta, se lanza una alarma.

Lowe, g. 2020, Automated Collection and Analysis of Infrared Thermograms for Measuring Eye and Cheek Temperatures in Calves, Nueva Zelanda.

2.2 Internet de las cosas para monitoreo de ganado

El sistema está compuesto por dispositivos de hardware, un sistema en la nube, una aplicación para el usuario final y técnicas innovadoras de medición de datos y algoritmos de análisis. El sistema fue probado en un escenario de la vida real y ha demostrado que puede monitorear de manera efectiva al ganado.

Nikodem M., 2020, IoT-Based Cow Health Monitoring System, Polonia.

2.3 Visión por computadora aplicada para detectar letargo a través del monitoreo del movimiento de los animales

Las últimas mejoras en el *deep learning* y la visión por computadora son herramientas poderosas que potencialmente abren un nuevo campo de investigación en epidemiología y control de enfermedades. Estas técnicas se utilizaron aquí para desarrollar un algoritmo destinado a rastrear y calcular el movimiento de los animales en tiempo real. Este algoritmo se utilizó en ensayos experimentales para evaluar el curso de la infección por peste porcina africana (PPA) en el jabalí euroasiático.

Fernández-Carrión, Eduardo, 2020, Computer Vision Applied to Detect Lethargy through Animal Motion Monitoring: A Trial on African Swine Fever in Wild Boar.

3. MATERIALES Y MÉTODOS

En esta sección, explicamos cómo se recogieron y procesaron los datos y, después, diferentes alternativas de algoritmos de compresión de imágenes para mejorar la clasificación de la salud animal.

3.1 Recopilación y procesamiento de datos

Recogimos datos de *Google Images* y *Bing Images* divididos en dos grupos: ganado sano y ganado enfermo. Para el

ganado sano, la cadena de búsqueda era "cow". Para el ganado enfermo, la cadena de búsqueda era "cow + sick".

El conjunto de datos se dividió en un 70% para entrenamiento y un 30% para pruebas. Los conjuntos de datos están disponibles en <https://github.com/mauriciotoro/ST0245-Eafit/tree/master/proyecto/datasets>.

3.2 Alternativas de compresión de imágenes con pérdida

3.2.1 Compresión fractal

Es un método de compresión con pérdida utilizada para imágenes digitales, reconocida por ser cara a nivel computacional pero muy rápida.

En este método es conveniente usar imágenes en que la información es repetida, pues los algoritmos fractales convierten estos datos parecidos en datos matemáticos denominados "códigos fractales", que son utilizados para recrear la imagen original. La codificación de la imagen se realiza de la siguiente manera:

Se parte la imagen en una cantidad de bloques de igual tamaño y para cada bloque este algoritmo busca otro con características similares y a estos bloques parecidos les asigna una función matemática que dé como resultado los bloques originales.

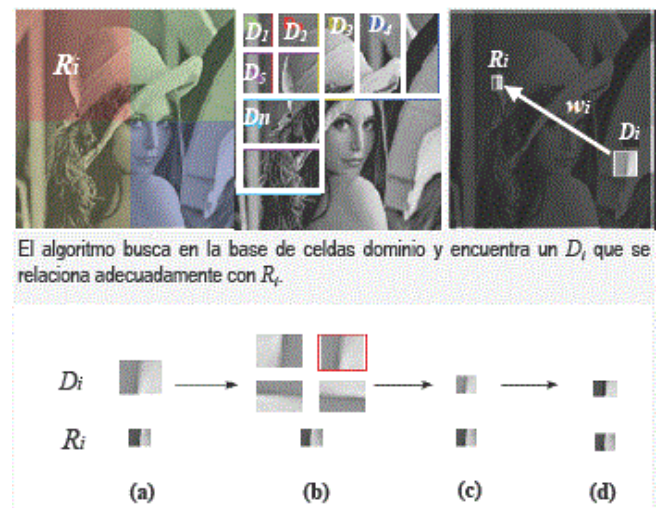


Figura 1: Compresión Fractal, ejemplo.

Tomada de:

http://www.revistasbolivianas.org.bo/scielo.php?script=sci_arttext&pid=S1729-75322014000200001&lng=es&nrm=iso&tlng=es

3.3 Alternativas de compresión de imágenes sin pérdida

3.3.1 Codificación Huffman

Creado por David A. Huffman, consiste en un algoritmo para la compresión de datos. Este algoritmo hace uso de una tabla con ciertos códigos que hacen referencia a un símbolo. La

elección de la representación de cada símbolo no se realiza de manera aleatoria, pues cada una de estas representaciones genera un código prefijo, el cual nunca será igual para dos símbolos. Este código prefijo representa los símbolos más comunes con las cadenas de bits más cortas, y viceversa.

La técnica básica de este algoritmo es la construcción de un árbol en el que se organizan en los nodos cada uno de los símbolos con su respectivo peso o frecuencia de aparición y se van agrupando hasta solo quedar un nodo. Esta agrupación se realiza primero agrupando los primeros dos nodos con menor frecuencia de aparición, estos generan un nodo intermedio y este proceso ocurre sucesivamente hasta que solo quede un nodo, que es llamado el nodo raíz del árbol de Huffman. Durante la construcción de este árbol también se asignan valores 1 y 0 a cada símbolo, en el caso de que en solo haya dos símbolos, asignara siempre 1 al primero y 0 al segundo, sin importa la frecuencia de aparición de cada uno, pero es de anotar el hecho de que en este caso nunca se realiza compresión de los datos, por lo que no es acertado el uso de este algoritmo en este caso específico.

Este algoritmo tiene la propiedad de ser optimo únicamente cuando la probabilidad de cada símbolo de entrada es una potencia negativa de dos, por lo tanto, es importante analizar para cuáles casos es más conveniente este algoritmo. Sin embargo, la codificación de Huffman es generalmente acertada debido a su velocidad, simplicidad y falta de problemas de patentes. También es remarcable el hecho de que la codificación de Huffman es utilizada en otros métodos de compresión como la deflación códec multimedia como JPEG y MP3.

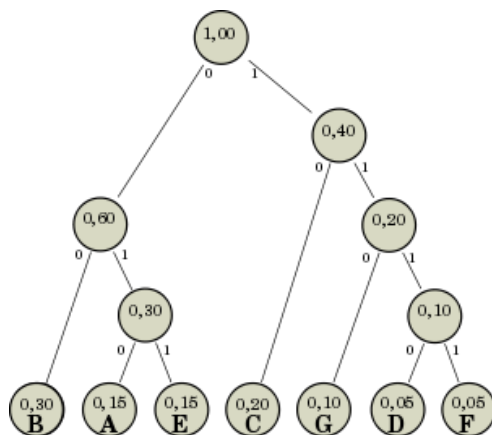


Figura 2: Árbol de Huffman, Ejemplo.

Tomada de:

https://es.wikipedia.org/wiki/Algoritmo_de_Huffman

3.3.2 Compresión de Burrows-Wheeler

Es conocida también como compresión por ordenación de bloques. Inventado por Michael Burrows y David Wheeler, en este algoritmo de compresión, no se cambia el valor de ninguno de los caracteres de la cadena, pues la

transformación de la cadena de caracteres permuta el orden de estos. Cuando en una cadena de caracteres hay secuencias de caracteres repetidos, la cadena transformada tendrá varias posiciones en las que un mismo carácter se repita en una fila. Para este caso se utilizan técnicas como *move-to-front transform* y *run-length encoding*.

La técnica de codificación *Run-Length* es la más simple de las técnicas de compresión, pues permite la compresión de cadenas de caracteres con caracteres repetidos, a estos les asigna un índice correspondiente al número de repeticiones que tiene en la cadena.

En la compresión de Burrows-Wheeler se organizan todas las posibles rotaciones de los caracteres del texto de entrada (en la primera columna), posteriormente se ordenan las filas (en la segunda columna) hasta obtener un texto de salida de codificación fácil (en la tercera columna), sin embargo, esto no es lo más interesante de este método de compresión, pues lo más interesante de este método es que es reversible, lo que permite regenerar el documento original partiendo de la última columna de datos, pues con esta se pueden generar todas las columnas anteriores hasta obtener el documento de entrada.

3.3.3 Codificación aritmética

La codificación aritmética consiste en que se entrega al algoritmo una imagen en una matriz con el valor de sus píxeles, y este algoritmo codifica la información de la imagen con un string de valores entre 1 y 0 en la línea de los números reales, de esta manera, los datos de la imagen logran tener un tamaño más pequeño. Aunque esta técnica es relativamente sencilla, no es tan eficiente al momento de comprimir los datos de la imagen.

Esta técnica permite que ocurra una compresión sin pérdidas y, como en la técnica de codificación Huffman, se basa en la probabilidad de aparición de un dato para crear el string de valores entre 1 y 0.

Para implementar esta técnica es necesario primero definir un modelo de predicción de patrones que podrían ser encontrados en el mensaje a codificar. La creación de este modelo permite que se produzca una codificación y decodificación más optima.

4. DISEÑO E IMPLEMENTACIÓN DE LOS ALGORITMOS

A continuación, explicaremos los algoritmos y las estructuras de datos utilizados en este proyecto. Las implementaciones se encuentran en GitHub en <https://github.com/isabellaecheverri/ST0245-002/tree/master/proyecto>.

4.1 Estructuras de datos

En el algoritmo de compresión sin pérdidas, hicimos uso del árbol de Huffman que, como se ve en la imagen, es un árbol binario que crea nodos a partir de la probabilidad que tiene cada carácter de aparecer. Empieza juntando los de menos

probabilidad, genera un nodo nuevo y realiza este proceso sucesivamente hasta conseguir un solo nodo.

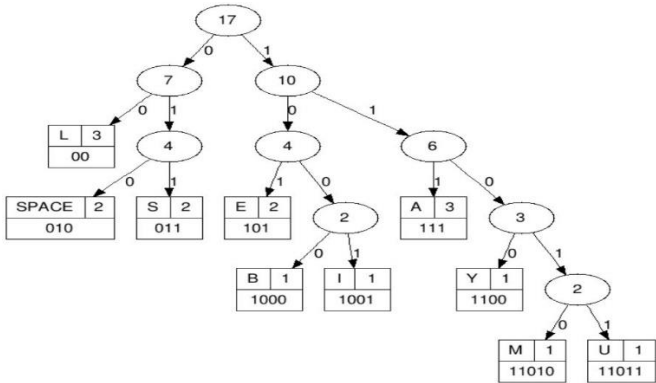


Figura 3: Árbol de Huffman generado a partir de las frecuencias exactas del texto "ISABELLA Y SAMUEL".

4.2 Algoritmos

En esta parte, describiremos e ilustraremos los algoritmos utilizados con y sin pérdidas. Explicaremos a grandes rasgos, el funcionamiento de estos algoritmos en cuanto a la compresión y la descompresión de imágenes.

Proponemos un algoritmo de compresión que, aunque es con pérdidas, devuelve imágenes de muy buena calidad. También explicamos cómo funciona la descompresión para el algoritmo propuesto.

4.2.1 Algoritmo de compresión de imágenes con pérdida

Explique, brevemente, cómo se aplicó un algoritmo de compresión de imágenes con pérdidas, como, por ejemplo, el tallado de costuras o el escalado de imágenes. Explique también la descompresión.

Para el algoritmo de compresión con pérdidas, utilizamos el algoritmo de “Singular Value Decomposition”. Este algoritmo se basa en múltiples métodos del algebra lineal y la generación de matrices a partir de la imagen original.

Para desarrollar este algoritmo de compresión, después de cargar la imagen, la escalamos a escala de grises con la librería IMAGE. Esto con el fin de facilitar la manipulación de matrices de la imagen, puesto que la matriz de pixeles de una imagen a color contiene tres veces más valores que la matriz en escala de grises. Posteriormente, obtenemos la matriz de pixeles de la imagen. Lo que sucede en SVD es que toma una matriz cuadrada y la divide en dos matrices ortogonales y una diagonal, es decir, toma la matriz de pixeles de la imagen y la divide en matrices más pequeñas, esto hace que podamos reescribir la matriz original como la sumatoria de las matrices más pequeñas. Y en este método, al comprimir la imagen, podemos decidir el número de matrices pequeñas que vamos a sumar. Mientras más de estas matrices sumemos, la calidad de la imagen se acercará más a la imagen original, sin embargo, el espacio en memoria aumentará.

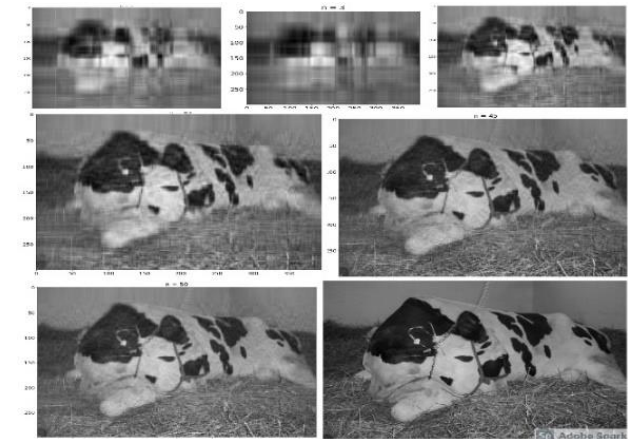


Figura 4: Compresión de imagen con índice de sumatoria progresivamente mayor con SVD.

Para la descompresión de la imagen se realiza un procedimiento muy sencillo. Se utiliza la matriz de pixeles de la imagen y se realiza el mismo procedimiento que para comprimir la imagen.

4.2.2 Algoritmo de compresión de imágenes sin pérdida

La implementación propuesta del algoritmo Huffman recibe como dato de entrada la matriz de pixeles de la imagen en escala de grises, transformación que es realizada previamente por operaciones previas. La codificación se da de forma tal que a partir de esta matriz se genera una cadena de números binarios de considerada longitud. Finalmente, la cadena de números binarios es descodificada y se retorna la matriz de pixeles inicial.

4.3 Análisis de la complejidad de los algoritmos

| Singular Value Decomposition | La complejidad del tiempo |
|------------------------------|---------------------------|
| Compresión | $O(N)$ |
| Descompresión | $O(N)$ |

Tabla 1: Complejidad temporal del algoritmo SVD de compresión y descompresión de imágenes. N representa el ancho de la matriz de pixeles y M representa el largo de la matriz de pixeles de la imagen.

| Algoritmo de Huffman | Complejidad del tiempo |
|----------------------|------------------------|
| Compresión | $O(N*M)$ |
| Descompresión | $O(N)$ |

Tabla 2: Complejidad temporal del algoritmo de Huffman de compresión y descompresión de imágenes N representa el ancho de la matriz de pixeles y M representa el largo de la matriz de pixeles de la imagen.

4.4 Criterios de diseño del algoritmo

Los algoritmos fueron elegidos en base a dos criterios principales: la estructura de datos, y el método de codificación.

El algoritmo de compresión sin pérdidas Singular Value Decomposition utiliza unos procedimientos algebraicos relativamente sencillos que se pueden entender con facilidad, además, es eficiente. Este algoritmo, sin embargo, no es tan bueno en cuanto a la compresión, pues si elegimos una compresión muy alta, la imagen que resulta es de muy mala calidad.

El algoritmo de Huffman utiliza una estructura de árbol para almacenar los valores de los píxeles de la imagen según su probabilidad de aparición, de modo que dada una imagen en escala de grises la extensión del árbol es reducido, gracias a la alta probabilidad de hallar numerosos valores repetidos. Tal asignación de probabilidades se da en valores de unos y ceros, factor que reduce tanto el tiempo de ejecución como el uso de memoria. Se exploraron alternativas, tales como el algoritmo LZW. El LZW usa una estructura de tablas hash y su proceso de compresión puede conseguir una complejidad de memoria menor al de Huffman. Sin embargo, en términos de complejidad temporal fue mucho más eficiente el logrado con el algoritmo de Huffman.

5. RESULTADOS

5.1 Evaluación del modelo

En esta sección, presentamos algunas métricas para evaluar el modelo. La exactitud es la relación entre el número de predicciones correctas y el número total de muestras de entrada. La precisión es la proporción de estudiantes exitosos identificados correctamente por el modelo a estudiantes exitosos identificados por el modelo. Por último, sensibilidad es la proporción de estudiantes exitosos identificados correctamente por el modelo a estudiantes exitosos en el conjunto de datos.

5.2 Tiempos de ejecución

En lo que sigue explicamos la relación entre el tiempo promedio de ejecución y el tamaño promedio de las imágenes del conjunto de datos completo, en la Tabla 6.

| Singular Value Decomposition | Tiempo promedio de ejecución (s) | Tamaño promedio del archivo (MB) |
|-------------------------------------|---|---|
| <i>Compresión</i> | 0.516676 | 0.06333 |
| <i>Descompresión</i> | 0.1639015 | 0.06333 |

Tabla 3: Tiempo de ejecución del algoritmo Singular Value Decomposition para diferentes imágenes en el conjunto de datos.

| Algoritmo de Huffman | Tiempo promedio de ejecución (s) | Tamaño promedio del archivo (MB) |
|-----------------------------|---|---|
| <i>Compresión</i> | 0.732333 | 0.06333 |
| <i>Descompresión</i> | 0.6141 | 0.06333 |

Tabla 4: Tiempo de ejecución del algoritmo de Huffman para diferentes imágenes en el conjunto de datos.

5.3 Tasa de compresión

Presentamos los resultados de la tasa de compresión del algoritmo en la Tabla 8.

| | Singular Value Decomposition | Algoritmo de Huffman |
|------------------------------------|--|-----------------------------|
| Tasa de compresión promedio | Varía según lo elegido en el algoritmo | 2:1 |

Tabla 5: Promedio redondeado de la tasa de compresión de todas las imágenes de ganado sano y ganado enfermo.

6. DISCUSIÓN DE LOS RESULTADOS

Se implementaron dos algoritmos de compresión de imágenes con y sin pérdidas, el Singular Value Decomposition y el algoritmo de Huffman. Ambos, de forma diferente, cumplen el objetivo de compresión y descompresión y permiten ser comparados por su eficiencia en consumo de tiempo, su uso de memoria, y la relación de compresión que provea cada algoritmo.

6.1 Trabajos futuros

Se plantean múltiples objetivos para trabajos futuros. Primero, optimizar los algoritmos propuestos en este trabajo. Segundo, explorar los resultados de hacer una implementación mixta de estos, por ejemplo hacer compresión con pérdidas y descompresión sin pérdidas. Tercero, implementar algoritmos como la Transformación de Coseno Discreto y LZW para hacer una comparación más detallada de la eficiencia de estos. Cuarto, adicionar el uso de redes neuronales y estrategias de entrenamiento de algoritmos para conseguir la clasificación de imágenes de ganado enfermo y ganado sano. Finalmente, la meta es probar estos algoritmos en el contexto de la Ganadería de Precisión.

RECONOCIMIENTOS

Los dos primeros autores son apoyados por una beca Sapiencia financiada por el Municipio de Medellín y por una Generación E financiada por el Estado Colombiano.

Todos los autores quieren agradecer a la Vicerrectoría de Descubrimiento y Creación, de la Universidad EAFIT, por su apoyo en esta investigación.

REFERENCIAS

1. D.A. Huffman, "A method for the construction of minimum-redundancy codes", Proceedings of the I.R.E., sept 1952, pp 1098-1102.
2. Gonzales, Ramon (2013), <http://ramon-gzz.blogspot.com/2013/04/codificacion-huffman.html>
3. Burrows M and Wheeler D (1994), A block sorting lossless data compression algorithm, Technical Report 124, Digital Equipment Corporation
4. Wikipedia,
https://es.wikipedia.org/wiki/Compresi%C3%B3n_de_Burrows-Wheeler
5. International Journal of Computer Applications. (2015) Gray-scale image compression Techniques: A review (No 13).
<https://www.ijcaonline.org/research/volume131/number13/kulkarni-2015-ijca->