# **Optimización de consumo de batería en ganadería de precisión con compresión de imágenes**

|  |  |  |  |
| --- | --- | --- | --- |
| Miguel Chacon  Universidad EAFIT  Colombia  machaconl@eafit.edu.co |  | Simón Marín Universidad Eafit Colombia smaring1@eafit.edu.co | Mauricio Toro  Universidad Eafit  Colombia  mtorobe@eafit.edu.co |

# **RESUMEN**

En esta investigación estudiaremos diferentes algoritmos de compresión que den solución a la cuestión del ahorro de batería y recursos en la gdp para la clasificación de la salud animal, el cual es un tema bastante importante debido a que al ser la ganadería de precisión una industria tan competitiva y en que la diferencia de resultados radica en el uso de los datos, lo ideal es buscar una forma óptima de aprovechar esta información con el menor gasto de recursos como la batería y tan rápido como sea posible. Algunos de los problemas relacionados a esta situación son la tarea de encontrar los mejores procedimientos para la compresión de datos y también como funciona la clasificación de la salud animal en la ganadería de precisión.

## **Palabras clave**

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

# **1. INTRODUCCIÓN**

Los productos generados por el ganado son una parte fundamental de nuestra dieta debido a los importantes aportes a la nutrición humana y la es la ganadería la industria que se encarga de manejar y sacar provecho de estos productos. durante una gran cantidad de tiempo la ganadería se ha manejado de una forma tradicional, archivando los datos útiles del ganado en cuadernos o libros, pero recientemente se ha buscado un nuevo enfoque haciendo uso de herramientas tecnológicas e innovación para el manejo de estos datos, esta nueva visión es la ganadería de precisión (gdp) cuyos principales objetivos es la digitalización automática de todos los datos importantes generados y su uso para la optimización y mejora de procesos en esta industria esto mediante un refinado uso de los datos, reduciendo el consumo de energía en los equipos utilizados en las granjas y disminuyendo la dimensión de estos datos y es en este punto que los algoritmos de compresión de datos los cuales hacen posible lo anteriormente mencionado hacen presencia.

# **1.1. Problema**

La ganadería de precisión es una muy buena forma de administrar esta industria pero al trabajar con tantos datos, cámaras, sensores, etc. el ahorro de energía y tiempo toman un papel fundamental, la compresión de imágenes trae a los ganadores un uso más óptimo de memoria y un menor tiempo de carga, ambas características muy preciadas en el contexto de la gdp ya que al estar en zonas rurales los recursos como luz, batería e internet son considerablemente limitados, al mejorar este proceso hace más fácil conseguir el objetivo por el cual se utilizan estas imágenes que es clasificar la salud de los animales utilizando tecnología e innovación, mejorando la detección y el tiempo de tratamiento de estas enfermedades y hacienda más sencillo este proceso para el ganadero.

**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.

**1.3 Estructura del artículo**

En lo que sigue, en la Sección 2, presentamos trabajos relacionales con el problema. Más adelante, en la Sección 3, presentamos los conjuntos de datos y los métodos utilizados en esta investigación. En la Sección 4, presentamos el diseño del algoritmo. Después, en la Sección 5, presentamos los resultados. Finalmente, en la Sección 6, discutimos los resultados y proponemos algunas direcciones de trabajo futuras.

**2. TRABAJOS RELACIONADOS**

## En lo que sigue, explicamos cuatro trabajos relacionados. en el dominio de la clasificación de la salud animal y la compresión de datos. en el contexto del PLF.

## **3.1 Sistema de adquisición y análisis de información acústica para ganadería de precisión**

En esta tesis se aborda el monitoreo del comportamiento alimentario de rumiantes, lo que resulta imprescindible para conocer el estado nutricional del animal y aumentar la eficiencia en el uso de los recursos pastoriles. Para esto fue necesario diseñar, desarrollar e implementar algoritmos de bajo costo computacional que puedan operar en sistemas embebidos de capacidad de cómputo reducida. Los algoritmos desarrollados se centraron en dos tareas de diferente escala temporal pero complementarias entre sí: (i) reconocimiento de movimientos mandibulares (arranque, masticación y arranque-masticación); y (ii) reconocimiento de actividades alimentarias (rumia y pastoreo). Para el diseño algorítmico se exploraron e implementaron diferentes técnicas de aprendizaje maquinal y de procesamiento de señales. Como fuente de entrada de información para el sistema se utilizaron los sonidos producidos por el animal durante su alimentación. Se alcanzaron resultados similares e incluso superiores en desempeño a métodos del estado del arte pero con menor costo computacional. Además se diseñó y desarrolló un sistema embebido dedicado para ejecutar dichos algoritmos.[1]

## **3.2 Desarrollo de un algoritmo de compresión de datos optimizado para imágenes satelitales**

Las imágenes satelitales son cada vez de mayor tamaño, esto genera problemas a la hora de procesarlas y utilizar su información mas relevante. Teniendo en cuenta también los desafíos del manejo de datos tanto en la plataforma satelital como en la tierra.  En esta investigación, se desarrolla e implementa un algoritmo de compresión con pérdida orientado a resolver esta problemática, utilizando la Transformada Discreta de Wavelets y la codificación Huffman.[2]

## **3.3 Clasificación de Datos Basado en Compresión**

## En este artículo se presenta un vista general de diferentes trabajos realizados a lo largo del mundo que utilizan técnicas de compresión de datos como base para el desarrollo de un método de clasificación, estas técnicas se basan en la Complejidad de Kolmogorov y la utilización de esta para implementar una medida de similaridad entre datos. El aporte principal de estos métodos es la no necesidad de un proceso de extracción de características para realizar la clasificación, lo cual hace que sea un método libre de parámetros, por lo que se puede aplicar a cualquier tipo de datos, ya sean texto, imágenes, audio, etc.[3]

## **3.4 Control biológico de la garrapata en el sector ganadero con procesamiento de imágenes.**

La escuela de Veterinaria y Zootecnia necesita la implementación de un algoritmo para el procesamiento de imágenes que reconozca de manera eficaz, la efectividad de un extracto natural como garrapaticida. Lo que haremos es comparar las imágenes para detectar el movimiento logrado y así obtener un resultado preciso ya que si se hace manualmente pueden obtenerse resultados erróneos, gracias a este sistema podemos identificar cuál de los extractos es el más eficiente.[4]

## **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".

En el siguiente paso, ambos grupos de imágenes fueron transformadas a escala de grises usando Python OpenCV y fueron transformadas en archivos de valores separados por comas (en inglés, CSV). Los conjuntos de datos estaban equilibrados.

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 .

Por último, utilizando el conjunto de datos de entrenamiento, entrenamos una red neuronal convolucional para la clasificación binaria de imágenes utilizando *Teachable Machine* de Google disponible en https://teachablemachine.withgoogle.com/train/image.

## **3.2 Alternativas de compresión de imágenes con pérdida**

## En lo que sigue, presentamos diferentes algoritmos usados para comprimir imágenes con pérdida.

**3.2.1 Tallado de costuras:**

Este algoritmo a diferencia de la mayoría está basado en la energía para adaptarse al recorte, posteriormente, en función de este valor de píxel se emplean 8 dominios conectados para la programación Dinamica para obtener el valor mínimo, aplicando este algoritmo columna por columna o fila por fila se obtendrá una línea de energía que en realidad es la fase. la línea de píxel con el valor de pixel más pequeño de dos columnas o filas adyacentes se elimina de la imagen original.

Los pasos básicos del algoritmo son los siguientes:

1. asigna un valor de energía a cada píxel
2. Encuentra las ocho rutas conectadas del píxel con el menor valor de energía
3. Eliminar todos los píxeles en la ruta
4. Repita los pasos 1-3 anteriores hasta que el número de filas / columnas eliminadas alcance el estado ideal.

A continuación, adjuntaremos las imágenes que describen el proceso de programación dinámica que utiliza el algoritmo para calcular una costura optima.

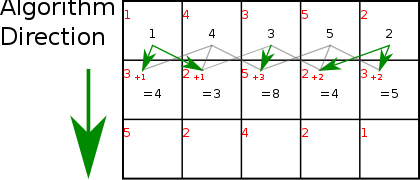


Imagen1. La fila superior no tiene nada encima, por lo que las energías son las mismas que las de la imagen de origen.

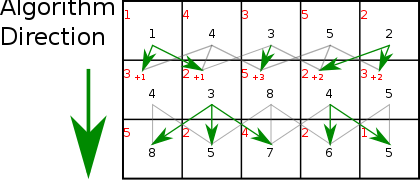


Imagen 2. Para cada píxel en el resto de las filas, la energía es su propia energía más el mínimo de las tres energías anteriores. Repita hasta llegar al fondo.

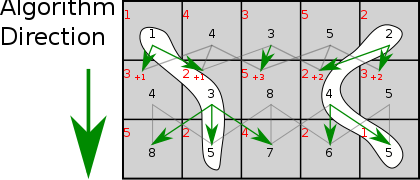


Imagen 3. Para las energías más bajas que tengamos al final, trabaje de nuevo los mínimos para recuperar la costura con la mínima energía.

**3.2.2 Transformada del coseno discreta (DCT)**

La transformada de coseno discreta es una transformada basada en la [Transformada de Fourier discreta](https://es.wikipedia.org/wiki/Transformada_de_Fourier_discreta), pero utilizando únicamente números reales.

La dct es un operador lineal invertible definido en el dominio real de Rn a Rn. Se puede expresar de forma equivalente como una matriz de nxn. representa una imagen como una suma de sinusoides de diferentes magnitudes y frecuencias. La función calcula la transformación de coseno discreta bidimensional de una imagen.

El DCT tiene la propiedad que, para una imagen típica, la mayor parte de la información visualmente significativa sobre la imagen se concentra en sólo unos pocos coeficientes del DCT. Por esta razón, el DCT se utiliza a menudo en aplicaciones de compresión de imágenes.

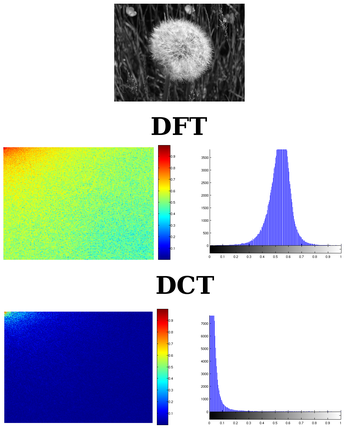


Imagen 4. En esta imagen podemos ver como la información en su mayoría se concentra en los primeros valores

**3.2.3 Compresión wavelet:**

es [formato de archivo](https://es.wikipedia.org/wiki/Formato_de_archivo) [privativo](https://es.wikipedia.org/wiki/Formato_cerrado) para imágenes [raster](https://es.wikipedia.org/wiki/Raster" \o "Raster) desarrollado por la empresa [Earth Resource Mapping](https://es.wikipedia.org/w/index.php?title=Earth_Resource_Mapping&action=edit&redlink=1" \o "Earth Resource Mapping (aún no redactado)) (en la actualidad propiedad de [Leica Geosystems](https://es.wikipedia.org/wiki/Leica_Geosystems)) y que presenta unos ratios muy altos de compresión, desde 10:1 hasta de 50:1, mediante el uso de técnicas de [ondículas](https://es.wikipedia.org/wiki/Ond%C3%ADcula). Como consecuencia de esto se reduce considerablemente el tamaño de los archivos, manteniendo una alta calidad gráfica y permitiendo una rápida compresión y desccomprime transformando las imágenes al espacio wavelet usando la Transformada Discreta de Wavelet (DWT) multi-nivel. A continuación, reduce la cantidad de información de la imagen mediante cuantización, para seguidamente comprimir las imágenes del espacio wavelet. La imagen comprimida ECW se procesa línea a línea directamente a partir de la imagen original. La técnica de compresión ECW puede comprimir imágenes de cualquier tamaño usando un [algoritmo recursivo](https://es.wikipedia.org/wiki/Algoritmo_recursivo) de [segmentación](https://es.wikipedia.org/wiki/Segmentaci%C3%B3n) que no precisa del uso de almacenamiento en disco mientras se realiza la DWT.ompresión mediante un uso escaso de la memoria [RAM](https://es.wikipedia.org/wiki/Memoria_de_acceso_aleatorio).

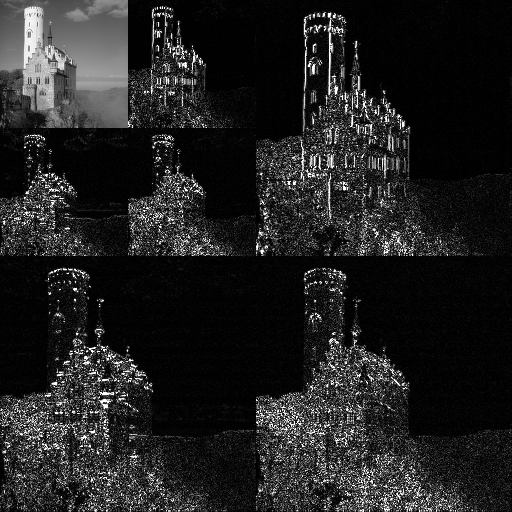


Imagen 5. Esta imagen es un ejemplo de la transformada wavelet usado en JPEG 2000.

**3.2.4 Compresión fractal**

La compresión fractal es un método de [compresión con pérdida](https://es.wikipedia.org/wiki/Algoritmo_de_compresi%C3%B3n_con_p%C3%A9rdida) para [imágenes digitales](https://es.wikipedia.org/wiki/Imagen_digital), basado en [fractales](https://es.wikipedia.org/wiki/Fractal). es el más apropiado para texturas e imágenes naturales, basándose en el hecho de que partes de una imagen, a menudo, se parecen a otras partes de la misma imagen. Los [algoritmos](https://es.wikipedia.org/wiki/Algoritmos) fractales convierten estas partes en datos matemáticos llamados «códigos fractales» los cuales se usan para recrear la imagen codificada.

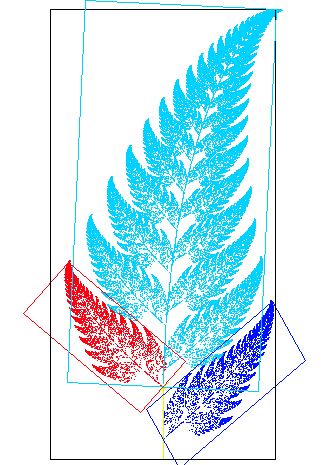


Imagen 6.

## **3.3 Alternativas de compresión de imágenes sin pérdida**

## En lo que sigue, presentamos diferentes algoritmos usados para comprimir imágenes sin pérdida.

**3.3.1 Transformada Burrows-Weller (BWT)**

La transformada de Burrows y Weller también conocida como compresión por ordenación de bloques es un [algoritmo](https://es.wikipedia.org/wiki/Algoritmo) usado en técnicas de [compresión de datos](https://es.wikipedia.org/wiki/Compresi%C3%B3n_de_datos) como en [bzip2](https://es.wikipedia.org/wiki/Bzip2). Fue inventado por Michael Burrows y David Wheeler en 1994 y basado en una transformación inédita descubierta por Wheeler en 1983, publicada en su artículo "A Block-sorting Lossless Data Compression Algorithm".

BWT es un algoritmo que toma bloques de datos como cadenas y los reorganiza en ejecuciones de caracteres similares. Luego de la transformación, el bloque de salida contiene exactamente los mismos elementos de datos antes de que comenzara, pero difiere en el orden. El algoritmo tiende a colocar caracteres similares uno al lado del otro lo que facilita la compresión del orden de datos resultante, esta es la razón por la se utiliza en el ámbito de la compresión.

Los pasos del algoritmo cuando se tiene una cadena R son:

1. agregue un carácter especial a la cadena R
2. después de agregar el carácter especial a la cadena R, se crean todas sus rotaciones escribiéndolas en una matriz conceptual (paso a).
3. Las filas de la matriz se ordenan lexicográficamente, siendo la última columna el resultado de la transformación (paso b).

La siguiente imagen ilustra el proceso.

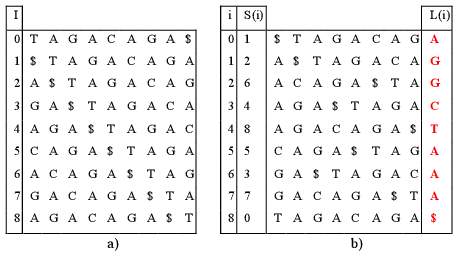


Imagen 7.

**3.3.2 LZ77**

**LZ77** o denominado como lz1 es un compresor basado en [algoritmo](https://www.ecured.cu/Algoritmo) sin pérdida se utilizan cuando la información a comprimir es crítica y no se puede perder información, por ejemplo en los archivos ejecutables, tablas de [bases de datos](https://www.ecured.cu/Base_de_Datos), o cualquier tipo de [información](https://www.ecured.cu/Informaci%C3%B3n) que no admita pérdida. El modelo lz77 es muy usado porque es fácil de implementar y es bastante eficiente.

En este [algoritmo](https://www.ecured.cu/Algoritmo) el codificador analiza el texto como una secuencia de caracteres, mediante una ventana deslizable compuesta por dos partes; un [buffer](https://www.ecured.cu/Buffer) de anticipación que es la opción que está a punto de ser codificada y un [buffer](https://www.ecured.cu/Buffer) de búsqueda, que es la parte dónde se buscan secuencias iguales a las existentes en el [buffer](https://www.ecured.cu/Buffer) de anticipación. Para codificar el contenido, o parte de él, del [buffer](https://www.ecured.cu/Buffer) de anticipación, se busca la secuencia igual en el [buffer](https://www.ecured.cu/Buffer) de búsqueda y la codificación resulta en indicar esta repetición como una tripleta **[offset, longitud, carácter siguiente]**.

Donde:

* Offset es la distancia desde el principio del [buffer](https://www.ecured.cu/Buffer) de anticipación hasta el comienzo de la secuencia repetida.
* Longitud es la cantidad de caracteres repetidos.
* Carácter siguiente es el símbolo siguiente a la secuencia en el [buffer](https://www.ecured.cu/Buffer) de anticipación.

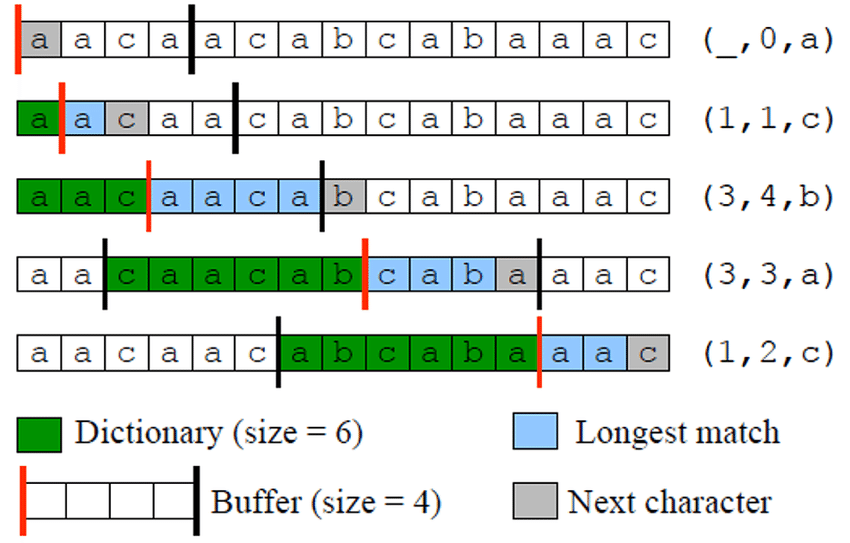


Imagen 8.

**3.3.3 LZW**

LZW es un algoritmo muy rápido tanto para la compresión como para la descompresión, basado en la multiplicidad de aparición de secuencias de caracteres en la cadena que se debe codificar. Su principio consiste en sustituir patrones con un código de índice y construir progresivamente un diccionario.

Además, funciona en [bits](https://es.ccm.net/contents/57-codificacion-binaria) y no en [bytes](https://es.ccm.net/contents/57-codificacion-binaria), por lo tanto, no depende de la manera en que el procesador codifica información. Es uno de los algoritmos más populares y se utiliza particularmente en formatos TIFF y GIF. Dado que el método de compresión LZW ha sido patentado por Unisys, el que se utiliza en imágenes PNG es el algoritmo LZ77, por el que no se pagan derechos de autor.

El algoritmo elabora un diccionario, el diccionario comienza con los 256 valores de la tabla [ASCII](https://es.ccm.net/contents/55-codigo-ascii). El archivo por comprimir se divide en cadenas de bytes (por lo tanto, para las imágenes monocromáticas codificadas en 1 bit, esta compresión no es muy eficaz), cada una de estas cadenas se compara con el diccionario y se agrega si no se encuentra ahí.

En la fase de compresión el algoritmo pasa por la cadena de información y la codifica. Si una cadena nunca es más corta que la palabra más larga del diccionario, ésta se transmite.

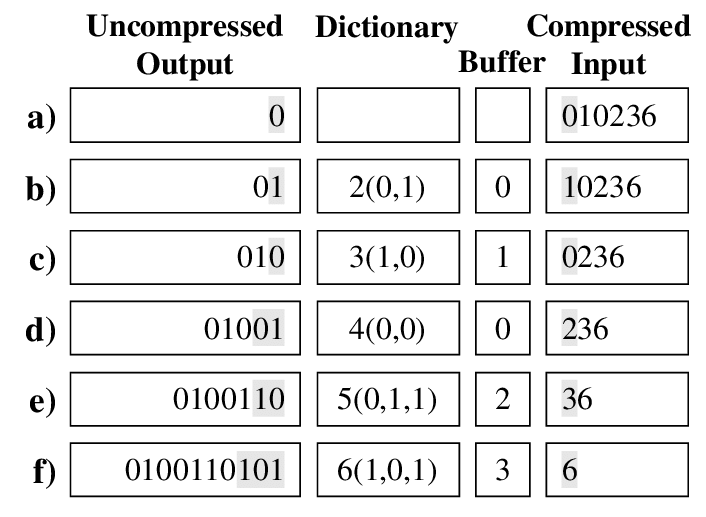


Imagen 9.

**3.3.4 codificación Huffman.**

El codificador Huffman crea una estructura arbórea ordenada con todos los símbolos y la frecuencia con que aparecen. Las ramas se construyen en forma recursiva comenzando con los símbolos menos frecuentes.

La construcción del árbol se realiza ordenando en primer lugar los símbolos según la frecuencia de aparición. Los dos símbolos con menor frecuencia de aparición se eliminan sucesivamente de la lista y se conectan a un nodo cuyo peso es igual a la suma de la frecuencia de los dos símbolos. El símbolo con menor peso es asignado a la rama 1, el otro a la rama 0 y así sucesivamente, considerando cada nodo formado como un símbolo nuevo, hasta que se obtiene un nodo principal llamado *raíz*.  
El código de cada símbolo corresponde a la sucesión de códigos en el camino, comenzando desde este carácter hasta la raíz. De esta manera, cuanto más dentro del árbol esté el símbolo, más largo será el código.

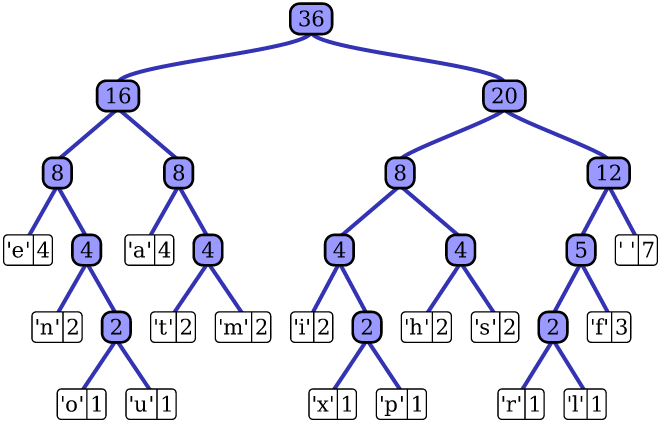


Imagen 10.

# **REFERENCIAS**

|  |  |
| --- | --- |
| [1] | Chelotti, J.O. 2018. *Sistema de adquisición y análisis de información acústica para ganadería de precisión*. |
| [2] | Cruz, K.J.A. 2017. Desarrollo de un algoritmo de compresión de datos optimizado para imágenes satelitales. |
| [3] | Roman-Gonzalez, A. 2012. Clasificacion de Datos Basado en Compresion. Revista ECIPeru. 9, 1 (2012), 69–74. |
| [4] | Zenteno Aguilar, C. del C. 2012. Control biológico de la garrapata en el sector ganadero con procesamiento de imágenes. (2012). |