**Teoría de la Información**

*Práctico especial*

****

**Número de grupo:** 15

**Ayudante:** Nacho Orlando

**Integrantes:**

* Osinaga, Lautaro **e-mail:** [**osinagalj@gmail.com**](mailto:osinagalj@gmail.com)
* Murakami, Agustín **e-mail:** [**Agustinmurakami@gmail.com**](mailto:Agustinmurakami@gmail.com)

**Resumen**

*De qué se trata el trabajo y qué se estudia.*

*Pequeño resumen describiendo la solución que propusimos, y los resultados que obtuvimos*

El trabajo consta de procesamiento de imágenes, la idea es poder extraer información de un conjunto de estas para encontrar similitudes, compararlas, comprimirlas, etc.

Pudimos encontrar similitudes en las fotos, ordenarlas según diferentes criterios, como también calcular diferentes valores probabilísticos sobre cada una de ellas.

**Introducción**

*Describir el problema tratado relacionándolo con los contenidos de la materia*

Se nos presentan varios problemas, en primer lugar se requiere ordenar distintas fotos según su parecido con una original, para ellos utilizamos el factor de correlación, el cual determina de manera lineal la relación entre dos variables, que llevado al caso de dos imágenes representa la relación entre pixel y pixel de cada una, por lo que podremos decir que si tiene un coeficiente de correlación alto (Cercano a 1) las imágenes son linealmente dependientes, aumentando su parentesco, por ejemplo, si las imágenes tuviesen un factor de 1, representaría que se analizó la correlación consigo misma.

Por otra parte, al añadir otra foto se nos requiere calcular las diferentes intensidades de esta, de la original, y de la más parecida del grupo con respecto a la original, para ello, generamos un histograma de cada una, le calculamos la media, que en procesamiento de imágenes nos representa el promedio de los valores de intensidad en los pixeles de la misma, y el desvío, el cual nos muestra la variación de los valores de píxeles alrededor de su media.

Luego tenemos la intención de poder comprimir imágenes con un mecanismo sin pérdida, para ello utilizamos el algoritmo de Huffman, el cual nos permite asignarle a una entrada de símbolos con diferentes probabilidades una secuencia binaria que la represente, con estas secuencias luego “traduciremos” cada pixel a su secuencia asociada, generando una única secuencia de bits, que será la presentación comprimida de la imagen.

**Desarrollo**

*Explicación de los cálculos realizados y planteo de pseudocódigos de los algoritmos utilizados, análisis de los resultados, comparaciones, gráficos, etc.* ***NOTA:*** *utilizar pseudocódigos para describir en el informe los algoritmos implementados (no código fuente).*

Para el ejercicio uno se nos pide ordenar por factor de correlación las 5 imágenes con respecto a la original, para ello utilizamos el cálculo representado con el *PseudoCorrelacion(****HACER****)*

Los índices obtenidos con respecto a la imagen original fueron los siguientes:

*Will\_1:  0.9788775174506447*

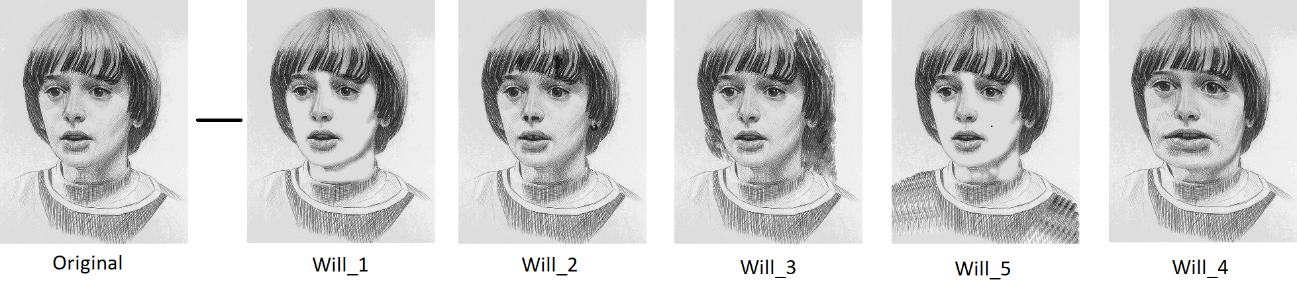
*Will\_2:  0.977957238009362*

*Will\_3:  0.9225582613856096*

*Will\_5:  0.9133072300754177*

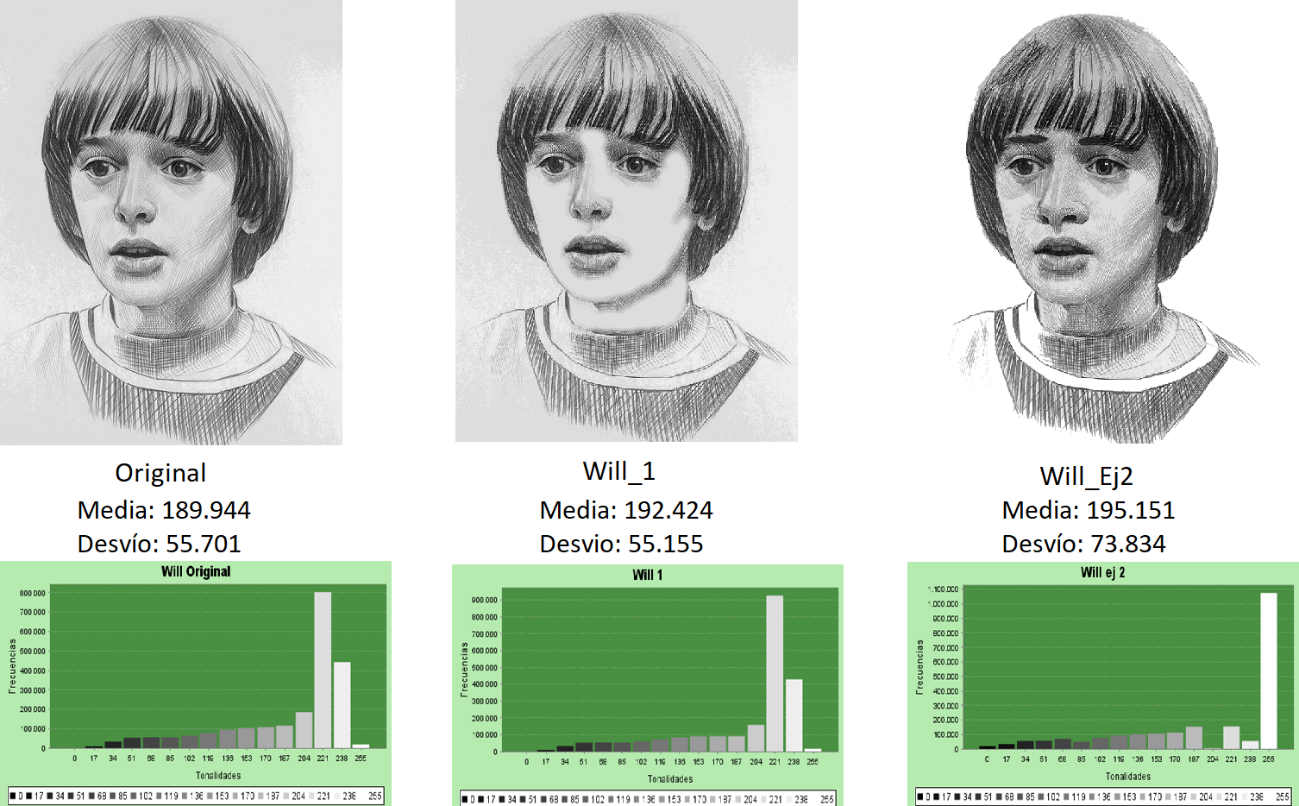
*Will\_4:  0.8207062451284244*

*(Ordenadas de mayor a menor para indicar cuál es más parecida)*



Viendo una comparativa a simple vista podemos deducir que la foto *Will\_1* es la más parecida, puesto que lo único que le cambia es un poco la intensidad del color de la cara y cuello, *Will\_2* vemos que presenta ciertos rasgos cambiado, como por ejemplo la nariz, pero mantiene una buena relación de las entonaciones, paremos a hablar de la menos parecida *Will\_4*, se puede observar a simple vista las diferencias, puesto a que tiene una gran diferencia en la forma a de la cara, entre las imágenes *Will\_3* y *Will\_5* podemos notar que son bastante similares en cuanto a su persona, pero presentan cambios a grandes rasgos que nos bajan la similitud con la imagen original, la imagen *Will\_3*, por ejemplo, presenta un sombreado adicional detrás del pelo, haciendo que sea una zona con menos relación, se puede ver en los números como bajó bruscamente… Con las primeras dos teníamos unos valores aproximados a 0.98, y en esta saltamos a 0.92, en la imagen *Will\_5* ocurre lo mismo, presenta mucha similitud en cuanto a Will, pero hay una zona que presenta mucha diferencia, que es la ropa.

Para el segundo ejercicio se nos pide analizar la media y el desvío de la imagen original, de la más representativa del ejercicio anterior y una nueva imagen *Will\_ej2*:



Para la obtención de los valores se utilizó el *pseudoMediaYDesvio(****HACER****).*

Podemos ver claramente por que la media en las tres fotos es simular al mirar los histogramas, ya que, se observa que la mayoría de los pixeles se corresponden con un valor en los colores más grandes (Color blanco), ahora en cuanto los desvíos podemos ver que en la primera imagen dan valores similares, debido a que están parejamente dispersos, es más, podemos observar ambas imágenes y vemos que la mayor diferencia entre estas es que la imagen dos presentas rasgos de colores más altos en la cara, luego son bastante equilibradas (Vemos también que la curva del histograma es parecida), en cambio en la imagen 3 se observa una diferencia amplia en cuanto al fondo, ya que, en esta es completamente blanco, por ende la punta más pronunciada de la curva se encuentra al final, haciendo que la variación de color sea más notoria (aún si los otros colores fuesen los mismos), asimismo presenta más cantidad de pixeles de color oscuro, aumentando todavía un poco más el desvío.

Para el ejercicio 3 donde teníamos que codificar una imagen mediante el algoritmo de Huffman se utilizó el código *PseudoHuffman(****HACER****)* para poder generar su correspondiente arbol*,* luego de esto, traducimos cada pixel de la imagen a su notación binaria en char mediante *PesudoCompresión(****HACER****)*, teniendo este código como una secuencia de char lo que hicimos fue convertirlo en una secuencia binaria (En bytes) mediante el método que nos brindó la catedra, asimismo tuvimos que generalizar para un código que no tenga solo extensión múltiplo de 8, añadiendo una cabecera con los datos necesarios para poder decodificarla, y la información necesaria para después poder volver a generar el mismo arbol de Huffman, volcamos esta cabecera en un archivo (fos) y luego le escribimos también la secuencia de bytes que representan a la imagen. Para decodificar la imagen teniendo este archivo lo único que hay que hacer es leer la cabecera para poder generar el árbol de huffman nuevamente, obtener la secuencia de bytes como una secuencia de char mediante el método brindado, y luego decodificar la imagen con *PseudoDecodificar(****HACER****).*

**Conclusiones**

*qué se hizo en el trabajo y qué resultados se obtuvieron.*

1. Dada la foto *Will(Original).bmp* que llevó Joyce a la estación de policía y la lista de fotos obtenidas por el sistema de búsqueda, implementar un algoritmo que permita ordenar la lista según su parecido con la foto de Will que entregó Joyce, utilizando el factor de correlación como medida de similitud. Analizar los resultados obtenidos.

ver para que sirve el factor de correlacion y ver por que nos da como nos da

*vamos a tener que implementar un algoritmo que nos  permita medir el factor de correlación entre 2 imagenes, correr el algoritmos sobre todos los pares de imágenes, y la idea es estudiar los resultados. Explicar por qeu el factor de correlación da tales resultados.*



1. Un policía amigo de Joyce llegó a la estación de policía un rato más tarde, trayendo consigo otra foto para analizarla. El policía de guardia, para ahorrarse la búsqueda en el SIDPE, propuso estudiar las similitudes respecto a la imagen que llevó Joyce mediante esta estrategia a implementar: Obtener la distribución de intensidades de la imagen de la foto que llevó Joyce, de la foto del niño más parecido (obtenida por el sistema de búsqueda) y de la foto que tenía el policía en su billetera. Generar el histograma de cada imagen y calcular la media y el desvío de cada distribución. Analizar las similitudes y diferencias entre uno y otro indicador.

1. Implementar un algoritmo que permita codificar una imagen mediante el método de Huffman y posteriormente reconstruirla mediante otro algoritmo decodificador (el archivo comprimido deberá contener la secuencia de bits codificados junto con los datos necesarios para la reconstrucción de la imagen original). Con dicho algoritmo codificador:
   1. Comprimir con Huffman semi-estático la imagen original.
   2. Comprimir la imagen del resultado más parecido con el código de Huffman asociado a la imagen original.
   3. Comprimir la imagen que trajo el policía con el código de Huffman asociado a la imagen original.
   4. Comprimir la imagen que trajo el policía utilizando Huffman semi-estático, esta vez utilizando el código generado para la propia imagen. Comparar con el resultado del inciso anterior.
   5. Comparar las tasas de compresión obtenidas por las compresiones en los ejercicios (a), (b), (c) y (d).

**Nota:** el algoritmo decodificador deberá permitir reconstruir los datos generados por el codificador y obtener la imagen original.

1. Tres canales de televisión distintos (Canal 2, Canal 8 y Canal 10) transmiten todos los días la fotografía de Will. Sin embargo, un fenómeno extraño hace que cada canal distorsione las imágenes de diferente forma. Al notarlo, el comisario de policía decidió estudiar el ruido de cada uno, esperando que esta información permita encontrar al joven Will. Considerando que la imagen de entrada a cada canal es la foto de Will provista por su madre al llegar a la estación de policía, y las imágenes a la salida de cada canal son las siguientes, resuelva:

|  |  |  |
| --- | --- | --- |
| **https://lh3.googleusercontent.com/gbmANfef5azetJ-AusTdwGLyLDkiHNiL9X8sFCRGCJDUoLs2fD0IVC9x4HV7KBrc1B8iVZIOAJ4rwT038Z2LnClz8lKLxSwJNLXoVvXps81EtelROxsOHs3KFTRBhZE9N4pn6oA** | **https://lh6.googleusercontent.com/yYcpCbx4FjNRvrk-p_bIC3P1U3eJLxrZ14c0Srr2Lc0_dJXlEwEq8bRKmb4bvxQPeBv7ktkDBkPVdDoZ3BlF5FHBB8h2tOWc3qOF_AdwOYcmxlRYyrOvvxRrThfsSol1um-F4AM** | **https://lh4.googleusercontent.com/Xgo9y4pfTF89aT3goejnFwvBRldiKvd3TT9Myp3wqChQ9VNzL3Q-QNVMeOyS5mhDAJ0vx4kPJRgTTROQv7x0uiL6gJ7TRU8P1kklZROXZ1Xh5Ar7rSXGdxg1bQtu0vC-_gov0rI** |
| Canal 2 | Canal 8 | Canal 10 |

* 1. Implementar un algoritmo que calcule la matriz de transición que describe cada canal.
  2. Implementar un algoritmo que calcule el ruido de cada canal utilizando muestreo computacional. Generar un gráfico que permita estudiar la evolución del error y la convergencia.