Trabajo Final Arquitectura de Computadores, Interfaces y Arquitectura de Hardware

*Juan Esteban Gallo, Daniela Llano Lozano, Manuel Quintero, Sara Ortiz Drada*

**INTRODUCCIÓN**

El desarrollo de algoritmos se enfoca en desarrollar aquellos que cumplan con una determinada función con el mayor desempeño posible, siendo normalmente el tiempo de ejecución la medida de desempeño. Generalmente, se tiene la creencia de que aquellos algoritmos con la misma complejidad tienen el mismo desempeño. Sin embargo, se ignoran muchos otros factores vinculados a la arquitectura de hardware, entre esos la jerarquía de memoria en los dispositivos, las cuales tienen un impacto en el desempeño o rendimiento de un algoritmo.

En el presente documento, se estudiarán tres tipos de un algoritmo de reconocimiento facial con enfoque en la extracción de características (Local Binary Patterns) a través de una metodología de diseño de experimentos.

**OBJETIVOS**

Este trabajo tiene como objetivo final la aplicación del diseño y análisis de experimentos con el fin de evaluar el desempeño de la etapa de extracción de características de una imagen y la efectividad del reconocimiento facial de un descriptor basado en tres tipos de operadores LBP.

**ANÁLISIS PRELIMINAR**

Componentes del experimento:

Variable de respuesta

Para este experimento, la variable de respuesta es el **tiempo de ejecución** del algoritmo.

Factores primarios

Los factores primarios son las variables controladas por el experimentador, de las cuales se quiere estudiar su impacto en la variable de respuesta.

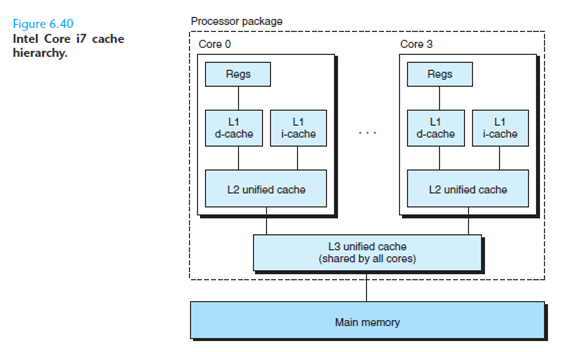
|  |  |
| --- | --- |
| Factores | Tratamientos |
| Tipo de algoritmo (operadores LBP) | 1. 8-1 2. 8-2 3. 16-2 |
| Aprovechamiento de localidad espacial | 1. Si 2. No |
| Tamaño de la imagen | 1. 320x243 2. 1280x1024 3. 2560x1440 4. 1730x2370 5. 464x512 6. 896x592 7. 900x1200 8. 640x360 9. 400x528 |

Factores secundarios

Son factores que influyen en la variable de respuesta pero no se quieren estudiar. Son factores que pueden controlarse para que no afecten el experimento ya sea eliminando su presencia o manteniéndolos constantes en el experimento.

* Tipo de Procesador

Durante todo el experimento se utilizó un equipo de cómputo con procesador core i7 de octava generación con la siguiente estructura de memoria especificada en las imágenes obtenidas del libro del curso .



Gráfica 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cache type | Access time (cycles) | Cache size (C) | Assoc. (E) | Block size (B) | Sets (S) |
| L1 i-cache | 4 | 256 KB | 8 | 64B | 64 |
| L1 d-cache | 4 | 256 KB | 8 | 64B | 64 |
| L2 unified cache | 11 | 1 MB | 8 | 64B | 512 |
| L3 unified cache | 30-40 | 8 MB | 16 | 64B | 8192 |

Tabla. Características la jerarquía de cache de Intel Core i7

* Sistema operativo

El sistema operativo utilizado fue Windows 10.

* Memoria

Para el experimento, se decidió utilizar un equipo de cómputo con una memoria RAM de 16GB.

* Interrupciones

Una interrupción puede provocar que el procesador interrumpa la tarea que esté realizando para atender la interrupción, aumentando así el tiempo de ejecución.

Para mitigar este efecto, durante la ejecución de los algoritmos no se realizan tareas adicionales.

**DISEÑO DEL EXPERIMENTO**

Se utilizó un diseño factorial completo, el cual se caracteriza por realizar todas las posibles combinaciones de los factores primarios. Se utiliza este diseño porque se quiere estudiar todas las posibles influencias en la variable de respuesta. Además, la facilidad y rapidez para obtener los datos permite considerar todas las opciones, lo cual es una de las ventajas principales de realizar experimentos relacionados con software.

De esta forma, un experimento contiene 54 observaciones.

3 tipos de algoritmo x 2 usos de localidad x 9 tamaños de imagen = 54 combinaciones

Se consideró que al ser 54 observaciones un cantidad poco significativa, se realizaria cada algoritmos 10 veces. Por lo tanto, una réplica contiene **540 observaciones**, además se consideró realizar **6 réplicas** del experimento, para obtener gran cantidad de datos,y asi, hacer mas significativo el análisis.

Por último, se realizó el experimento en **una unidad experimental**, es decir, en un mismo equipo de computo.

**EJECUCIÓN DEL EXPERIMENTO**

Para la ejecución se utilizó el entorno de programación Visual Studio y el lenguaje de programación C#. Para cada réplica, se reinicia el programa en ejecución, siendo la toma de datos significativa gracias a que se creaba cada vez una nueva instancia del modelo, incluyendo los datos que puedan verse almacenados en la caché a en el escritor de archivos en este caso StreamWriter, a su vez las réplicas incluían por cada imagen la inicialización de las variables utilizadas para la creación del arreglo bidimensional que representa su respectivo byte por pixel. Una vez teniendo esta información se utilizaba la herramienta que nos proporciona C# llamado StopWatch el cual proporciona un conjunto de métodos y propiedades que se pueden utilizar para medir con precisión el tiempo transcurrido en una medida de tiempo.

Para obtener un dato más preciso se midió el tiempo en nanosegundos, dentro de un ciclo se medía iterativamente el tiempo que le tomaba a cada algoritmo obtener los datos para las operaciones mediante los accesos a memoria, apenas se obtenía el tiempo se debaja de medir y para la siguiente iteración se volvía a instanciar el reloj y se reiniciaba para asegurar que no influenciara en tiempo la instanciación o acceso a un objeto ya creado previamente. Cada imagen en cada ciclo se probaba en cada método, el resultado se guardaba en un archivo txt diferenciándose solo por la réplica que estaba realizándose y la imagen correspondiente.

En cuanto al entorno de la ejecución se toma en cuenta que la máquina se mantiene conectada a la corriente, y a una temperatura ambiente, en donde todas las capacidades se vean óptimas. Por el mismo lado, se corre solamente el programa de Visual Studio y se cierran todas las demás aplicaciones en segundo plano, para que los recursos no se vean afectados y sean destinados meramente a las operaciones que se realicen en cada algoritmo.

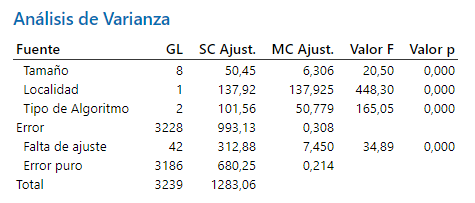
**ANÁLISIS DE LOS RESULTADOS**

Al obtener los datos, se realizó una normalización de estos, a partir de un análisis de los accesos, por lo que, se procedió a realizar una división entre el tiempo y la multiplicación del ancho y alto de la imagen correspondiente, y el número de accesos del algoritmo a evaluar.

Una vez obtenidos los datos normalizados se procedió al análisis utilizando el software estadístico MINITAB.

**Análisis de Varianza (ANOVA)**

Se realizo, en primer lugar, un ANOVA de 3 factores obteniendo los siguientes resultados:



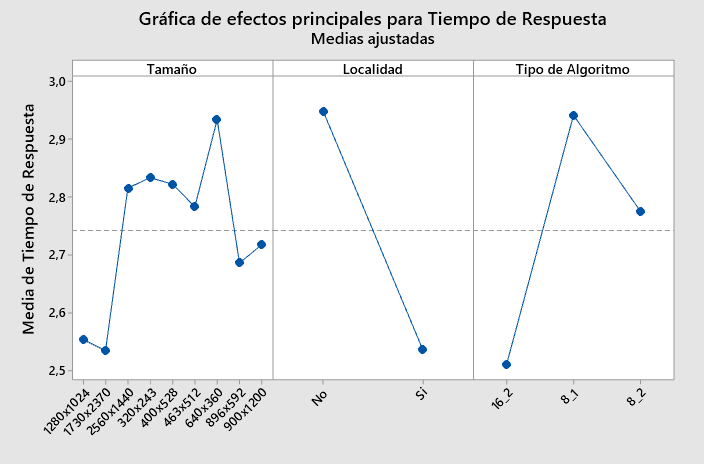
Para la interpretación de los resultados hay que recordar que un análisis de ANOVA prueba para cada factor definido las siguientes hipótesis:

**Hipotesis nula: el factor NO influye en la media de los datos**

**Hipotesis alterna: el factor SÍ influye en la media de los datos**

Si el valor p es menor al nivel de significancia (5%), se rechaza la hipótesis nula por lo tanto con un 95% de confianza se puede deducir lo siguiente:

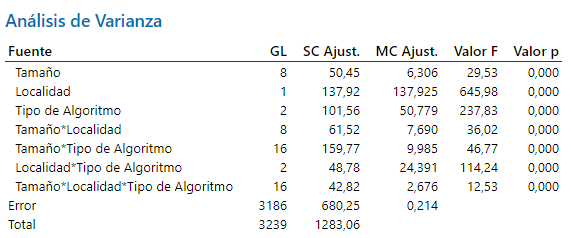
**Todos los factores tienen una influencia** **en la media de los datos** (tiempos de ejecución). Cada uno de los factores afecta independientemente a la media de los datos por lo que variando los niveles de cada uno de los factores, solo tomando en cuenta un sólo factor, encontramos que son estadísticamente desiguales. Por ejemplo, estadísticamente no es lo mismo tener una buena localidad a no tenerla, estas tienen una diferencia significativa. Lo anterior lo representa la siguiente gráfica:



Grafica 2.

Se puede ver un comportamiento extraño con respecto a los tamaños de las imágenes, puesto que en algunos casos las de menor tamaño tienen una media de tiempo de respuesta mayor que imágenes más grandes. Esto es un comportamiento atípico debido a que por tener una cantidad menor de píxeles, la imagen es menos pesada y podría ser almacenada en una caché de menor nivel. Lo anterior implica que al realizar los accesos de memoria debería ser mucho más rápido, además, debería tener una mayor cantidad de hits en la caché, ya que la imagen debería poder almacenarse en su totalidad en la caché o por lo menos una gran parte. Este comportamiento se analizará un poco más a fondo más adelante en el documento.

Este análisis no considera las interacciones que pueden darse en los diferentes factores. Por ello, se realizó un análisis de la varianza en los factores considerando todas las posibles interacciones, para así comprobar las ideas previamente expuestas, obteniendo los siguientes resultados:



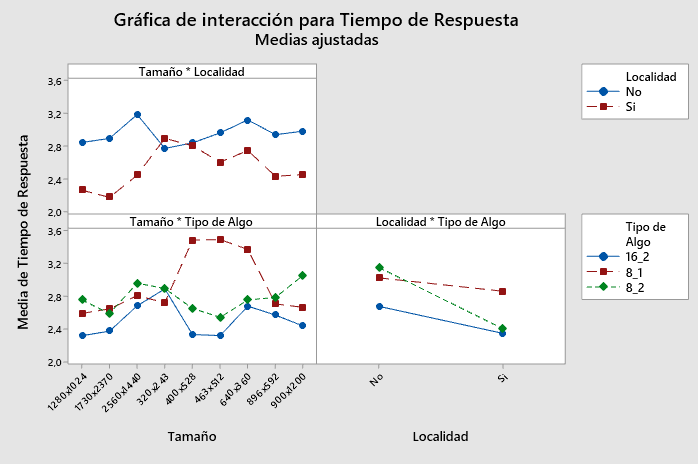
Para las interacciones, el análisis ANOVA define las siguientes hipótesis:

**Hipotesis nula: No hay interacción entre los factores**

**Hipotesis alterna: Si hay interacción entre los factores**

Considerando el valor p, se puede decir con un 95% de confianza que para **cualquier combinación de factores existe una interacción.** Cada una de las combinaciones de los factores obtuvo el mismo resultado, rechazar la hipótesis nula, la interacción entre los factores afecta el resultado del tiempo de ejecución por lo que sin importar la combinatoria de los factores ya sean 2 o sean los 3 juntos, el resultado final se verá afectado.

Estos resultados vienen resumidos en la siguiente gráfica:



Gráfica 3.

Como se puede observar en las tres gráficas de arriba, para las combinaciones de 2 factores se puede ver como los datos tienen altos y bajos diferenciados, especialmente cuando entre los factores se encuentra el tamaño. Primero, se puede ver cómo el algoritmo que toma la mayor cantidad de puntos (16-2) es el que menos se demora en cada una de las medidas del tiempo de respuesta pero tiene un tiempo de respuesta muy semejante al algoritmo 8-2. Además, la gráfica también permite mostrar la diferencia tan marcada que existe cuando hay buena localidad o no, pero es importante recalcar que para tamaños muy pequeños de imagen como lo es una foto de 320 x 243 y 400 x 528 no genera una gran diferencia en la medida del tiempo de respuesta haciendo buen uso o no de la localidad. Por último se puede ver como el tipo de algoritmo no varía mucho en su respuesta cuando estos son 16-2 y 8-2 y tienen buena localidad pero cuando esta no esta no se tiene 8-2 se asemeja más a 8-1.

Algunas de estas observaciones se ven justificadas por el almacenamiento en cada uno de los niveles de la caché del computador y por el uso de la localidad. Al no hacer buen uso de la localidad no se recorre secuencialmente la memoria, entonces los saltos en los accesos de memoria aumenta la velocidad del programa. Adicionalmente, si la imagen no cabe en su totalidad en la caché quedará particionada en dos niveles de memoria (L0 y L1 por ejemplo) cosa que aumenta el tiempo de acceso a los datos y al combinarlo con mala localidad la sobrescritura de los segmentos de la caché ocasionan una mayor cantidad de miss al leer la imagen.

Continuando con lo anterior, este comportamiento se ve mejor expresado en las *Gráfica 2* de tamaño, puesto que cuando este factor se junta con otro, la interacción entre ellos afecta la media de respuesta. En la gráfica 2 podemos ver que a pesar de ser tamaños diferentes, algunos tamaños demoran menos que otros a pesar de ser más grandes, esto es debido a que las imágenes cuyo tiempo de respuesta es cercano, la gran mayoría, su peso en KB es cercano también, por eso es que dan un tiempo de respuesta parecido gracias a lo que ya se mencionó sobre la caché. Por ejemplo, 320 x 243 - 400 x 528 - 463 x 512 tiene un peso de 35 KB - 50 KB - 70 KB respectivamente, por lo que no ocupan ni la mitad del primer nivel de la caché y se esperaría que tuvieran tiempos semejantes. Sin embargo, la imagen 2560 x 1440 tiene un tiempo aproximado a las imágenes anteriores aunque cuenta con un peso de 1278 KB, esto es un comportamiento atípico porque la imagen se almacena a un nivel mayor de la caché y por ende debería demorarse más. Se genera la duda de si leer una imagen en escala de grises y otra a color afecta significativamente al resultado.

Ahora, como se puede observar en la Gráfica 2, resultado del análisis ANOVA, cada uno de los factores (el tamaño, la localidad y el tipo de algoritmo) afecta de manera diferente e independiente a la media de los datos. Así pues, es de esperarse que una combinación de cualquiera de estos factores sea distinta a otra combinación. Analizando los tamaños de imágenes 400x528, 463x512 y 640x360 y el tipo de algoritmo 8-1 se puede apreciar que para la combinación Tamaño-Localidad se tiene un comportamiento diferente que al de Tamaño-Tipo de Algoritmo. Para la primera, la gráfica tiende a bajar y luego a subir, por el contrario para la segunda combinación los datos de la gráfica suben y luego bajan. Así pues, se puede ver la diferencia de los resultados de ambas combinaciones de factores.

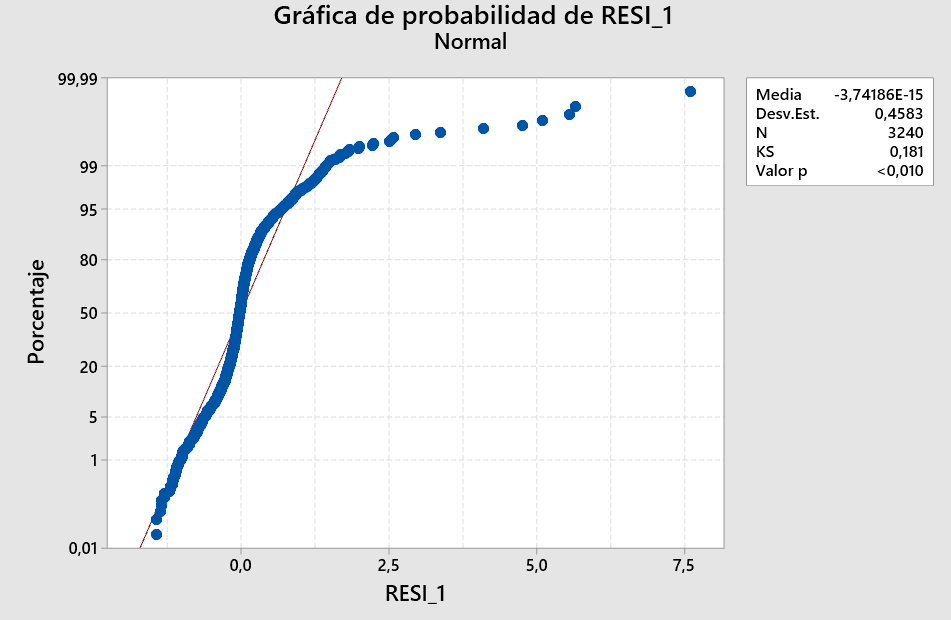
**Prueba Post Anova**

Se realizó la prueba de Post - Anova Fisher para diferenciar las combinaciones de niveles que son o no estadísticamente equivalentes. Los resultados de esta prueba se pueden encontrar en la carpeta del proyecto con el nombre ComparacionTiempoResouestaFisher.pdf

Al analizar los resultados, se evidencian 23 agrupaciones, es decir, 23 grupos de resultados que agrupan combinaciones estadísticamente equivalentes. Es importante destacar que existe un conjunto de combinaciones que pertenecen a varias agrupaciones al mismo tiempo. Por ejemplo, la combinación de imagen 2560 x 1440, sin aprovechar localidad espacial y con el tipo de algoritmo 16-2 se encuentra presente en 6 agrupaciones diferentes, siendo esta unión de niveles estadísticamente equivalente a casi el 25% de las posibles conjugaciones. A este, le siguen combinaciones que se encuentran en 4, 3 o 2 diferentes grupos. También, es importante recalcar que sólo existe una combinación que pertenece a un único grupo, la imagen de tamaño 400 x 528 que si aprovecha la localidad espacial y con el tipo de algoritmo 8-1 cuya media es la mayor.

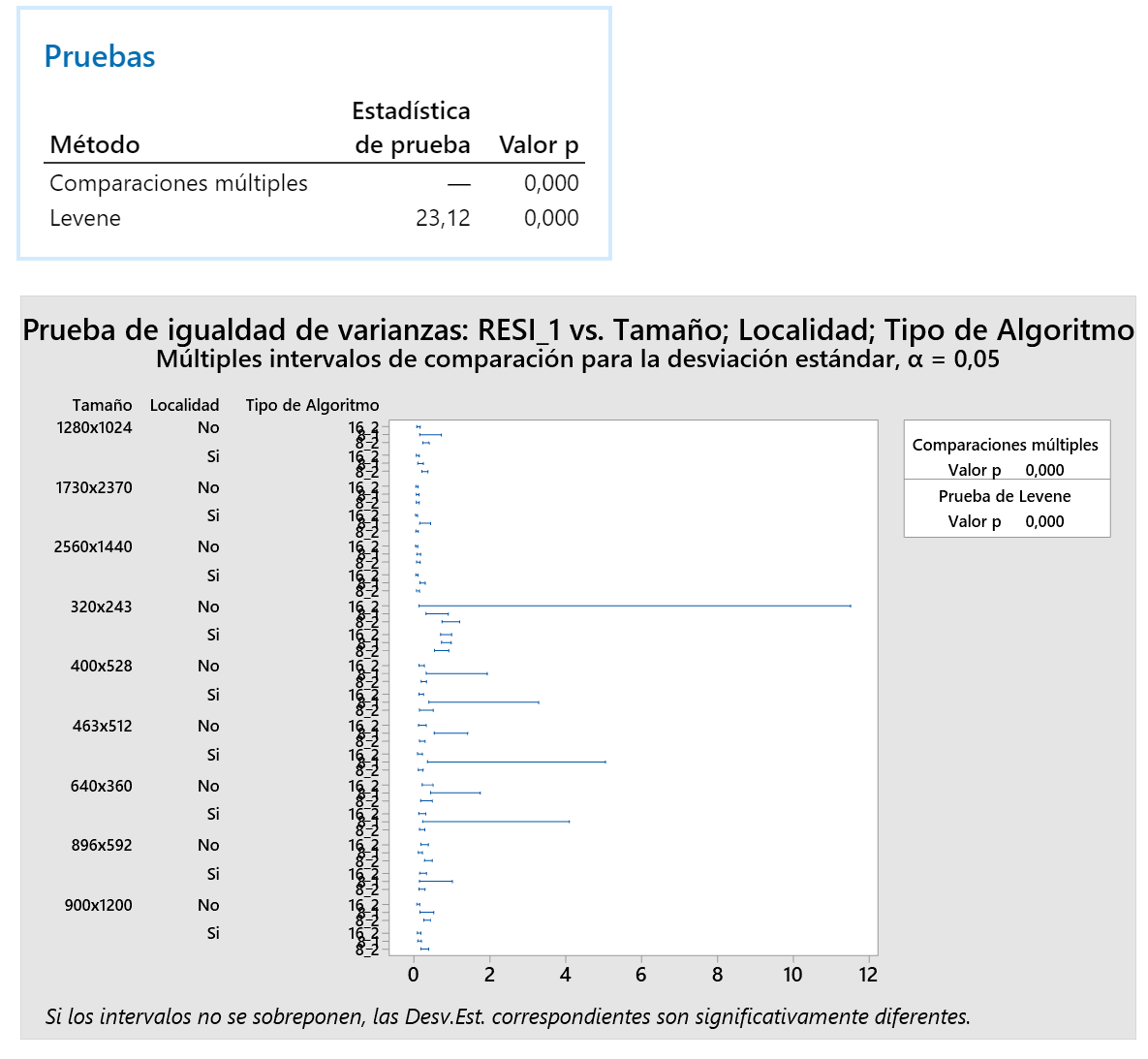
**Prueba de Normalidad**

Por otra parte, se determinó que era necesario hacer una prueba normalidad de los datos, por lo que, se realizó la gráfica de probabilidad normal, con el fin de obtener la información necesaria para contrastar la normalidad del conjunto de datos.



Se puede ver de que a pesar de que obtuvimos una gráfica a primera vista homogénea, se evidencio que con el valor p obtenido los datos realmente no están siguiendo una distribución normal. Por lo tanto, se cree que esto es debido a los factores secundarios y la carga de los datos que se da cada vez que se vuelve a realizar una ejecución.

**Levene interpretación**

****

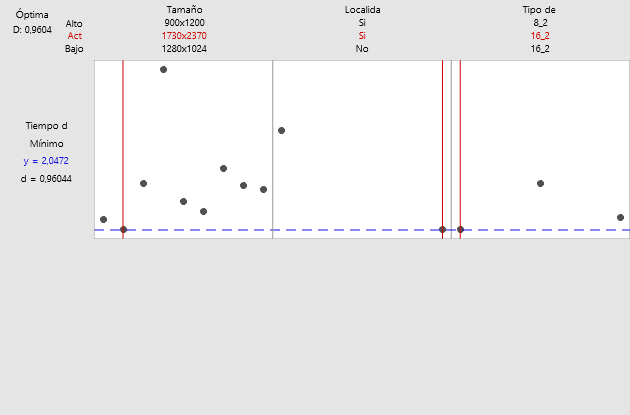
En la gráfica de resumen podemos evidenciar un valor p de 0,000 y 0,000 el cual son menores al nivel de significancia común de 0,05. Podemos rechazar nuestra hipótesis nula, por tanto todas las diferencia entre los grupos son estadísticamente significativas y los intervalos de evaluación no se sobreponen entre sí tomando en cuenta todas las igualdades programadas. Podemos decir entonces con un 95% de confianza que dichas diferencias significativas entre las valores sacados del tamaño de la imagen, localidad y tipo de algoritmo afectan directamente el tiempo de ejecución. Se puede evidenciar en un pico mayor de varianza debido a un dato atípico, durante una toma de dato en algoritmo 16-2 con no localidad con la imagen de 320x243.

**CONCLUSIONES**

1. **Al no contar con datos distribuidos normalmente se pueden esperar inconsistencias en el experimento.** Esto se ve reflejado en las tablas y gráficas provistas por MINITAB, lo cual complica el análisis de las mismas.
2. **Se evidenciaron datos atípicos que se ven reflejados en las pruebas de Levene y en las de normalidad**, por lo que se puede afirmar con mayor certeza la primera conclusión.
3. **Se encontró un posible posible factor que afectó el experimento en la toma de datos.** Las imágenes a escala de grises y las de color parece que afectan el tiempo que demora el algoritmo en desarrollarse.
4. **Independiente de cual sea el tipo de algoritmo, se evidencia un mejor rendimiento a la hora de usar la versión que aproveche la localidad espacial.** Aunque existieron datos que al no utilizar la localidad espacial, daban resultados semejantes.
5. **Los algoritmos 8\_2 y 16\_2 se comportan de manera muy similar cuando aprovecha la localidad espacial**, debido a que, al normalizarlos, se reduce el ruido provocado por el tamaño de la imagen y la cantidad de accesos a memoria generando que los dos algoritmos sean bastante equivalentes en su lógica.
6. **El tamaño de las imágenes no es un factor principal que afecta el tiempo de respuesta.** Se observó que el peso de la imagen influía más en el tiempo de respuesta. Sin embargo, al no tener los datos normales no se puede concluir con suficiente seguridad.
7. **Se demostró que para un rango de tamaños en específico, el algoritmo 8-1 toma un mayor tiempo de respuesta que los otros.**
8. **Cada factor, tomado de forma independiente, afecta significativamente el tiempo de respuesta.**

Por otra parte, MINITAB permite obtener la configuración óptima. Para este caso, se obtuvo que para obtener el mejor desempeño en el algoritmo LBP, la combinación de los factores debe ser con los siguientes niveles:

* TAMAÑO: 1730x2370
* LOCALIDAD: SI
* TIPO DE ALGORITMO: 16\_2

****