

Trabajo final Arquitectura de Computadores/ Interfaces y Arquitectura Hardware

Outcome B: Desarrollar y llevar a cabo un experimento apropiado, analizar e interpretar los datos, y usar el juicio de ingeniería para sacar conclusiones.

Los experimentos desempeñan un papel muy importante en ingeniería porque sirve de apoyo en las fases de validación de un servicio, modelo o desarrollo de un componente software o hardware de un sistema en condiciones más cercanas a los escenarios de trabajo, o cuando se quiere estudiar el comportamiento de un sistema.

A lo largo del curso se estudió la estructura y arquitectura de los computadores, donde el objetivo en sí mismo del curso es **evaluar cómo las estructuras hardware y componentes software de un computador afectan el rendimiento y la utilidad de los programas de aplicación**, tanto en el procesamiento, acceso y almacenamiento de los datos de los programas que los manipulan. Por tal razón la evaluación y análisis del desempeño es el fin último de un trabajo final.

Un componente importante en la arquitectura de un computador es el sistema de memoria, ya que normalmente se convierte en el cuello de botella de los tiempos de ejecución. Entonces la determinación y medición del tiempo de acceso es crucial para tomar decisiones de configuraciones correctas o adecuadas y determinar y explicar desde la perspectiva del programador, como diferentes programas o algoritmos con la misma complejidad algorítmica pero con diferente grado de aprovechamiento del principio de localidad, podrían tener diferencias significativas a la hora de su ejecución en un computador con una configuración determinada y especialmente en su interacción de con el sistema de memoria.

Para comparar y analizar la eficiencia de los algoritmos y que son escritos en lenguaje de alto nivel, establecer una medida precisa de la eficiencia de un algoritmo no es fácil, ya que, si compara con el tiempo de ejecución, ésta respuesta tiene una gran variabilidad. La complejidad de un algoritmo deberá estar relacionada con el número de operaciones elementales necesarias (asignaciones, comparaciones, sumas, restas, multiplicaciones, divisiones, etc.) para resolver el problema.

El estudio de la estructura y arquitectura hardware de un equipo de cómputo **permite comprender cuales son los componentes hardware y elementos software que impactan en el desempeño**. Dichos factores van desde el tipo de **carga** de trabajo o exigencia computacional, **tipo de CPU**, configuración de la jerarquía de **memoria** del computador, el **sistema operativo**, **compiladores**, **lenguajes de programación**, los **tipos de datos** utilizados en el programa, entre otros. De allí la importancia de realizar un diseño experimental bien hecho para establecer los niveles de interacción de los factores y el impacto en la variable respuesta en el que se esté interesado medir.

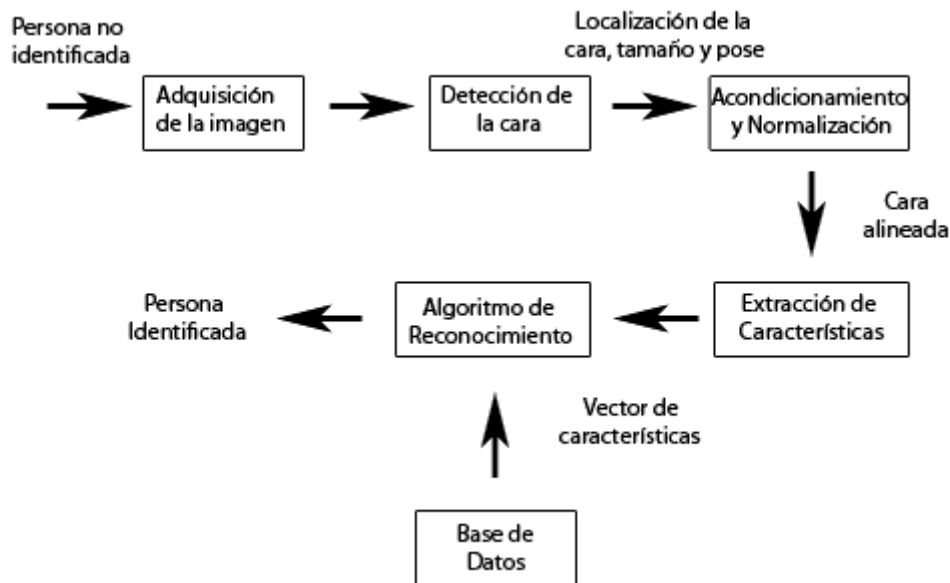
Marco teórico: Reconocimiento facial

El reconocimiento facial es una tarea que los humanos realizamos de manera rutinaria y sin esfuerzo en nuestra vida diaria. La amplia disponibilidad de sistemas de cómputo potentes y de bajo costo como los computadores embebidos como las raspberry pi y equipos de escritorio ha creado un enorme interés en el procesamiento automático de imágenes digitales en una variedad de aplicaciones, incluidas la autenticación biométrica, vigilancia, la interacción humano computador y la gestión multimedia. La investigación y el desarrollo en el reconocimiento facial automático sigue naturalmente. El reconocimiento facial tiene varias ventajas sobre otras modalidades biométricas, como la huella digital y el iris: además de ser natural y no intrusivo, la ventaja más importante de la cara es que puede capturarse a distancia y de forma discreta.

La identificación de rostros implica una coincidencia de uno a muchos que compara una cara de consulta con varias caras en una base de datos para asociar la identidad de la cara de consulta a una de las de la base de datos. En algunas aplicaciones de identificación, uno solo necesita encontrar la cara más similar. En una verificación sobre una lista de observación o vigilancia, el requisito es más que encontrar rostros más similares; se especifica un umbral de nivel de confianza y se informan todas las caras cuyo puntaje de similitud está por encima del umbral.

El desempeño de un sistema de reconocimiento facial depende en gran medida de una variedad de factores, como la iluminación, la posición facial, la expresión, la edad, el cabello, el desgaste facial y el movimiento. En función de estos factores, las aplicaciones de reconocimiento facial se pueden dividir en dos grandes categorías en términos de cooperación del usuario: (1) escenarios de usuario cooperativo y (2) escenarios de usuario no cooperativo. El caso cooperativo se encuentra en aplicaciones como el inicio de sesión en el computador, el control de acceso físico y el pasaporte electrónico, donde el usuario está dispuesto a cooperar presentando su rostro de manera adecuada (por ejemplo, en una postura frontal con expresión neutral y ojos abiertos) para que se le otorgue el acceso o privilegio. En el caso no cooperativo, que es típico en aplicaciones de vigilancia, el usuario no es consciente de ser identificado. En términos de distancia entre la cara y la cámara y la posición del rostro, las aplicaciones no cooperativas presentan problemas más desafiantes en comparación con los escenarios cooperativos (p. e., Control de acceso). En éste trabajo final solo nos centraremos en aplicaciones de reconocimiento facial cooperativo del usuario.

El proceso de reconocimiento facial consta de cuatro módulos principales:



En el link https://es.wikipedia.org/wiki/Sistema_de_reconocimiento_facial puede leer las definiciones de las etapas:

1. Detección de la cara
2. Alineación de la cara
3. Extracción de características
4. Reconocimiento

En el trabajo final solo se contemplará las dos últimas etapas, es decir solo la extracción de características y el reconocimiento. Por ello, la entrada del sistema serán imágenes que correspondan solo al segmento de la cara, y ya pre-procesadas, es decir imágenes en escala de grises.



Figura 1: ejemplos de sección de rostros

Etapa 1: Extracción de características (Local Binary Patterns)

El uso de patrones binarios locales (Local Binary Patterns) en el análisis facial comenzó en 2004 cuando se propuso una nueva representación facial para el reconocimiento facial. En este enfoque, la imagen de la cara se divide en varias regiones de las cuales se extraen las características de LBP y se concatenan en un histograma de características que se utiliza como descriptor de la cara. Este enfoque ha evolucionado exitosamente y ha sido adoptado y desarrollado por una gran cantidad de grupos de investigación y empresas de todo el mundo.

El operador LBP y sus variantes se han utilizado no solo en el reconocimiento facial sino también en varios otros problemas relacionados con la cara, como la detección facial, el reconocimiento de la expresión facial, la clasificación de género y la estimación de la edad. (Rahim, Hossain, Wahid, & Azam, 2013)

El éxito de LBP como descriptor o extractor de características de la cara se debe al poder discriminativo y la simplicidad computacional del operador, y su solidez a los cambios de la escala de grises causados, por ejemplo, por variaciones de iluminación. El uso de histogramas como características también hace que el enfoque LBP sea robusto para enfrentar la desalineación y plantear variaciones. (Stan Z & Anil K, 2011)

El código de Matlab de los operadores de LBP se puede encontrar y descargar gratuitamente desde <http://www.ee.oulu.fi/mvg/page/downloads>.

El operador LBP original forma etiquetas para los píxeles de la imagen al tomar como umbral el valor del pixel central y compararlos con los pixeles vecinos en una ventana de 3×3 y considerando el resultado como un número binario. El histograma de estas $2^8 = 256$ etiquetas diferentes se puede utilizar como un descriptor de imagen. Vea la Fig. 2 para ver una ilustración del operador básico de LBP.

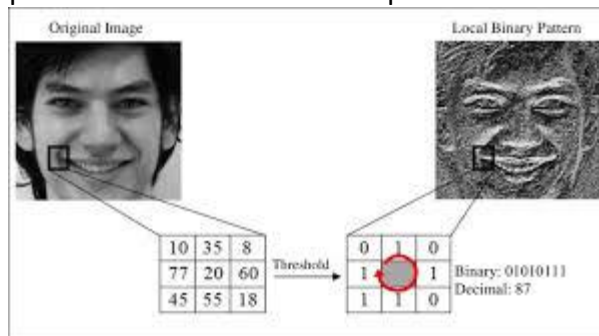


Figura 2: Operador LBP. Tomado de https://www.researchgate.net/figure/12-The-basic-LBP-operator_fig9_314193214

El operador se ha ampliado para utilizar barrios de diferentes tamaños. La notación (P, R) se utilizará para vecindades de píxeles, lo que significa P puntos de muestreo en un círculo de radio R. Observe la Fig. 3 para ver un ejemplo de vecindades circulares. Para el proyecto se usará los operadores LBP (8,1), LBP (8,2) y LBP(16,2)

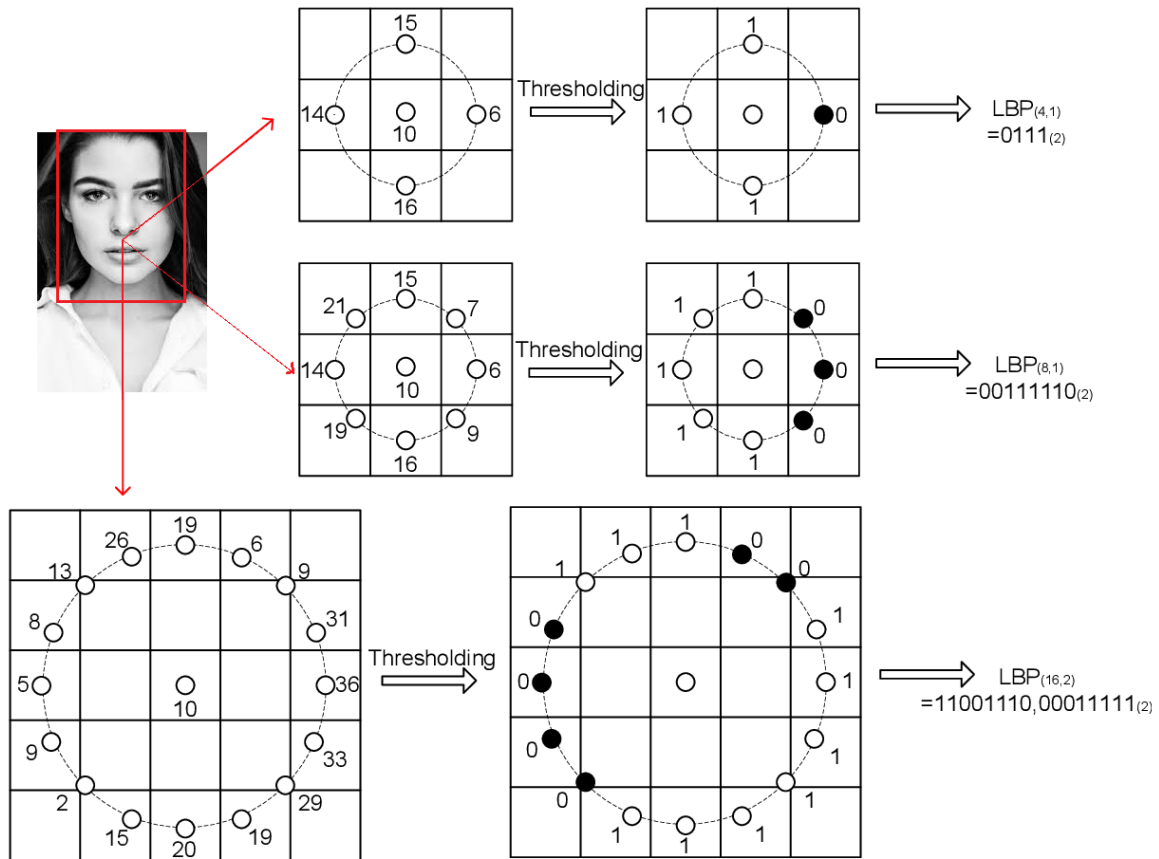


figura 3

En un píxel dado (x_c, y_c) de la imagen de la cara con el valor en escala de grises g_c , su valor del operador LBP se calcula comparando g_c con los valores de sus P píxeles adyacentes a la distancia R (se encuentran en un círculo de radio R -ver Fig. 3 para más detalles) como:

$$LBP^{P,R}(x_c, y_c) = \sum_{i=1}^P s(g_i^{P,R} - g_c) 2^{i-1}$$

fórmula tomada de (Rahim, Hossain, Wahid, & Azam, 2013)

Donde $s(x)$ es definido como:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0; \\ 0 & \text{if } x < 0. \end{cases}$$

El LBP (P, R) produce 2^P valores de LBP (P, R) para cada píxel de una imagen y construir un histograma con 2^P elementos como descriptor de la imagen. Después de obtener la imagen de etiquetas $LBP^{P,R}(x, y) [f_l(x, y)]$, entonces el histograma LBP se puede definir como

$$H_i = \sum_{x,y} I\{f_l(x, y) = i\}, \quad i = 0, \dots, n-1,$$

en el que n es el número de etiquetas diferentes producidas por el operador LBP y

$$I\{A\} = \begin{cases} 1, & \text{if } A \text{ is true,} \\ 0, & \text{if } A \text{ is false.} \end{cases}$$

El siguiente paso para crear un descriptor basado en LBP, esto requiere dividir la imagen resultante LBP en k bloques de $W \times H$ píxeles (por ejemplo, 2×4 , 4×4 , 8×8). Se genera un histograma local para cada bloque de la imagen para construir descriptores de imagen locales. Cada histograma local contiene 2^p elementos. Los histogramas locales se concatenan para formar un solo histograma global, como se muestra en la Figura 4. El enfoque del histograma global expresa efectivamente la información en tres niveles diferentes: el código LBP individual contiene información a nivel de píxel, los histogramas locales contienen información sobre un nivel regional, y los histogramas regionales concatenados contienen una descripción global. Como tal, el histograma resultante codifica características locales y globales en una representación compacta que lo hace más robusto para las variaciones de posición e iluminación del objeto.

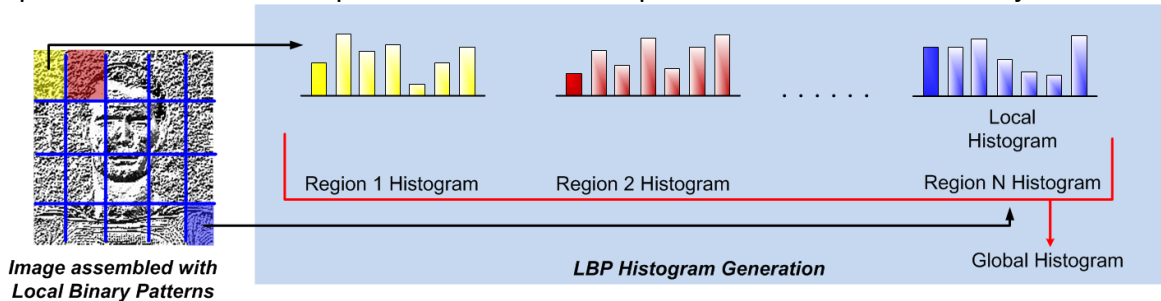


Figura 4: Generación del histograma LBP (Kyrkou, 2017)

Etapa 2: Reconocimiento facial usando LBP

Para la tarea de reconocimiento facial, normalmente, se utiliza una regla de clasificación de vecino más cercano. Esto se debe al hecho de que el número de imágenes de entrenamiento (galería) por sujeto es bajo, a menudo solo uno. El experimento lo hará mínimo con cuatro sujetos distintos, pero en lo posible similares (similar edad, mismo género, sin gafas, todos con o sin barba, similar longitud de cabello). Para cada una de las imágenes de entrenamiento se calculará el descriptor LBP, entonces se debe utilizar una función de distancia entre vectores y el clasificador de vecino más cercano (o distancia menor) para el reconocimiento del sujeto. La función de distancia se recomienda usar la Chi cuadrado. Con dos vectores $X = [x_1 \ x_2 \dots x_M]$; $Y = [y_1 \ y_2 \dots y_M]$ la distancia Chi cuadrado (Chi Square) entre X e Y es:

$$dist_{chi}(X, Y) = \sum_{i=0}^M \frac{(x_i - y_i)^2}{x_i + y_i}$$

No es necesario implementar esta función, la distancia Chi cuadrado está disponible en casi todas las librerías de manipulación de vectores en los diferentes lenguajes de programación.

Objetivo y condiciones trabajo final

El trabajo final tiene como **objetivo** 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 3 tipos de operadores LBP (Local Binary Patterns)**. Los tres tipos de operadores serán **LBP(P,R)=(8,1) y (8,2) y (16,2)**.

Cuantificar y explicar el impacto en el desempeño que tiene los 3 tipos de operadores LBP, ejecutándose a su vez en dos versiones del mismo algoritmo, pero variando la localidad espacial. También se desea estudiar el impacto (si lo tiene) del tamaño de las imágenes en el tiempo de respuesta normalizado en la ejecución de la obtención de la imagen LBP.

Condiciones del estudio experimental para la **fase 1** de extracción de características:

- a) **Implementar el algoritmo Local Binary Patterns para tres tipos de operadores LBP(P,R)=(8,1) y (8,2) y (16,2)**. Estas tres versiones deben devolver el mismo tamaño de imagen LBP. Para ello puede ignorar los bordes de la imagen de entrada. Estas funciones deben implementarse aprovechando el principio de localidad espacial.
- b) Como no se implementará algoritmos de detección de rostros sobre una fotografía, entonces las **imágenes de entrada serán imágenes de la sección del rostro**. (ver figura 1, ejemplos de sección de rostros)
- c) Para **cada una de las tres versiones del ítem a)** debe codificar una versión que **no aproveche la localidad espacial** en el recorrido de los píxeles de la imagen.
- d) Como no se está interesado en medir el impacto del lenguaje de programación, **usará el mismo lenguaje de programación**.
- e) Un **factor de estudio** o de tratamiento debe ser el **tamaño de la imagen**, por eso el tamaño de las imágenes debe variar y **tener al menos 4 niveles**, los dos niveles intermedios (2 y 3) deben copar al menos el 50% de la memoria cache L1 Y L2 respectivamente.
- f) Para ésta fase, se está interesado en **medir y estudiar el desempeño en el tiempo de ejecución de los algoritmos LPB** que obtiene la imagen LPB. Es decir, no se mide el tiempo que se invierte en generar el vector de histogramas locales ni globales.

Condiciones del estudio experimental para la **fase de extracción de características**:

- a) Para esta segunda fase, se está interesado en **medir la efectividad de reconocimiento** de las tres versiones de los operadores LBP [(8,1) y (8,2) y (16,2).]
- b) Para esta prueba debe hacerse al menos con **cuatro sujetos de entrenamiento** distinto.
- c) Usted debe **diseñar la forma de medir la efectividad de reconocimiento facial** recreando condiciones muy similares escenarios de usuario cooperativo como por ejemplo control de acceso.

Debe incluir una **sección donde explique en detalle y a la luz de los temas del curso**, la relación del comportamiento de la memoria y el tiempo de respuesta que describió los resultados del experimento. Tenga en cuenta entre otros conceptos, los principios de **localidad, la jerarquía de memoria de cada tipo de procesador y el AMAT.**

Para efectos de evaluación y calificación se aplicará una rúbrica la cual considera los siguientes aspectos que sigue la mayoría de los pasos de la metodología general de diseño de experimentos:

Aspecto a evaluar	Factor
Diseñar un experimento al seleccionar el tipo más adecuado, de acuerdo a una hipótesis dada, los recursos disponibles y los factores que deben ser medidos y controlados.	Identificación de los factores primarios, niveles y variable respuesta y definición del procedimiento de recolección de datos.
	Identificación de factores secundarios y de ruido y la manera de controlar sus efectos
	Selección del tipo de experimento de acuerdo con la hipótesis / cuestiones por resolver, los factores y las limitaciones existentes
Llevar a cabo un experimento diseñado siguiendo los procedimientos definidos para adquirir datos sobre las variables apropiadas y reportar y consignar adecuadamente los resultados.	La ejecución del experimento siguiendo el procedimiento.
	Reporte de los resultados
Analizar e interpretar los resultados de un experimento para exponer conclusiones que permitan contestar hipótesis o preguntas del experimento.	Análisis e interpretación de las observaciones y relacionarlas con la teoría del curso.
	Conclusiones apoyadas por los datos reportados

Grupos de trabajo 4 personas.

Fecha límite para la sustentación: viernes 29 de noviembre entre 8:00 a.m y 15:00 p.m
(los horarios serán asignados por el profesor)

Bibliografía

- Kyrkou, C. (11 de 2017). *Object Detection Using Local Binary Patterns*.
Obtenido de LBPs are local patterns that describe the relationship between a pixel and its neighborhood.:
<https://medium.com/@ckyrkou/object-detection-using-local-binary-patterns-50b165658368>
- Rahim, A., Hossain, N., Wahid, T., & Azam, S. (2013). *Face Recognition using Local Binary Patterns (LBP)* (Vol. Volume 13 Issue 4 Version 1.0). Bangladesh: Global Journals Inc. (USA). Obtenido de
https://globaljournals.org/GJCST_Volume13/1-Face-Recognition-using-Local.pdf
- [Hennessy J., Patterson D. Computer Architecture: A Quantitative Approach (5ta edición) Morgan Kaufmann Publishers; 2012