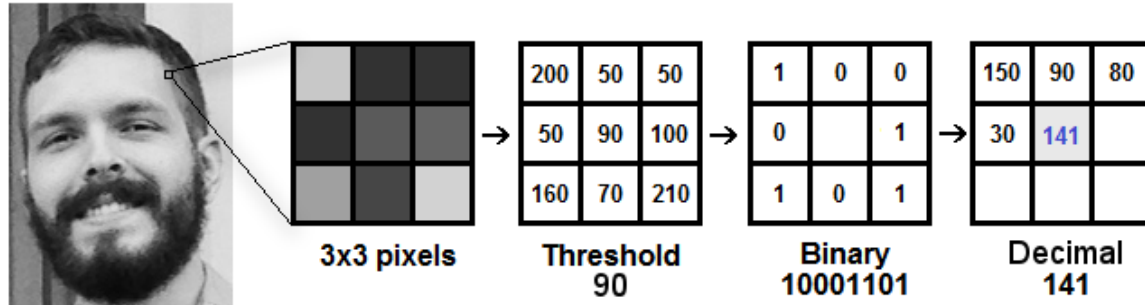


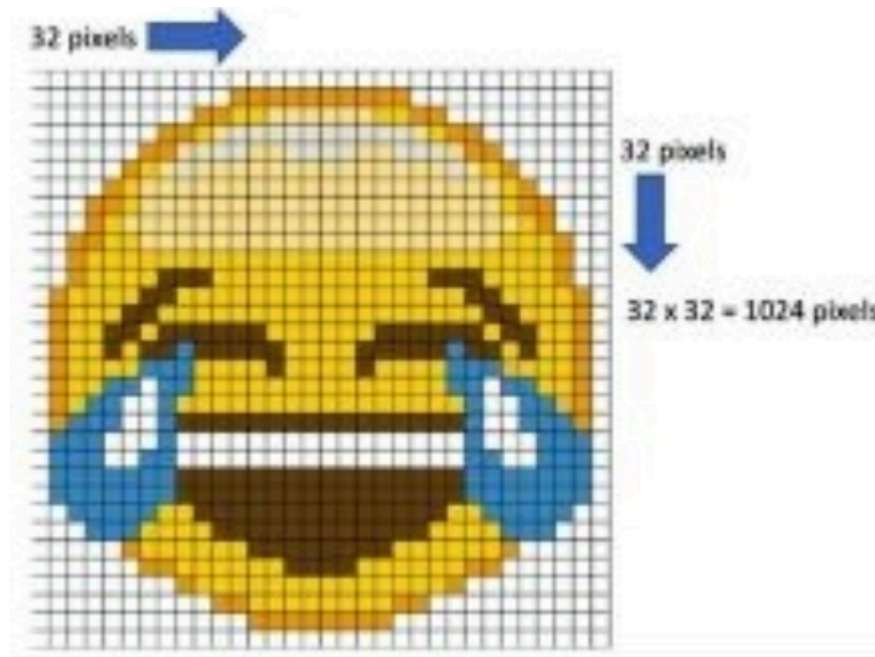
# Comprendiendo el reconocimiento facial mediante el algoritmo LBPH



LBPH (Histograma de Patrón Binario Local) es un algoritmo de reconocimiento facial utilizado para reconocer el rostro de una persona. Se destaca por su rendimiento y su capacidad para reconocer el rostro de una persona tanto en el frente como de perfil.

Antes de adentrarnos en la intuición detrás del algoritmo LBPH, primero entendamos un poco sobre los conceptos básicos de imágenes y píxeles para comprender cómo se representan las imágenes antes de abordar el contenido sobre el reconocimiento facial. Así que empecemos a entender imágenes y píxeles.

## Imágenes y Píxeles



Todas las imágenes se representan en formatos de matriz, como se puede observar aquí, y están compuestas por filas y columnas. El componente básico de una imagen es el píxel.

Una imagen está formada por un conjunto de píxeles, siendo cada uno de ellos pequeños cuadrados. Al colocarlos uno al lado del otro, podemos formar la imagen completa. Un solo píxel se considera la información mínima posible en una imagen. Para cada imagen, el valor de los píxeles oscila entre 0 y 255.

La imagen aquí tiene 32 píxeles de ancho y 32 píxeles de alto.

Y cuando multiplicamos 32 por 32, el resultado es 1024, que es el número total de píxeles en la imagen. Cada píxel está compuesto por tres valores: R, G y B, que son los colores básicos rojo, verde y azul. La combinación de estos tres colores básicos creará todos los colores presentes en la imagen, por lo que concluimos que un solo píxel tiene tres canales, uno para cada uno de los colores básicos.

Ahora que tenemos cierto entendimiento de las imágenes y los píxeles, será más fácil comprender la intuición detrás del algoritmo LBPH. Así que comencemos con la intuición del algoritmo LBPH (Histogramas de Patrones Binarios Locales).

## LBPH (Histogramas de Patrones Binarios Locales)

12	15	18
5	8	3
8	1	2

Vamos a empezar analizando una matriz que representa una parte de la imagen. Como aprendiste anteriormente, una imagen se representa en estos formatos. En este ejemplo, tenemos tres filas y tres columnas, y el número total de píxeles es nueve. Seleccionemos el píxel central aquí, con un valor de ocho, y apliquemos una condición. Si el valor es mayor o igual a 8, el resultado es '1'; de lo contrario, si el valor es menor que ocho, el resultado es cero. Después de aplicar la condición, la matriz se verá así:

1	1	1
0	8	0
1	0	0

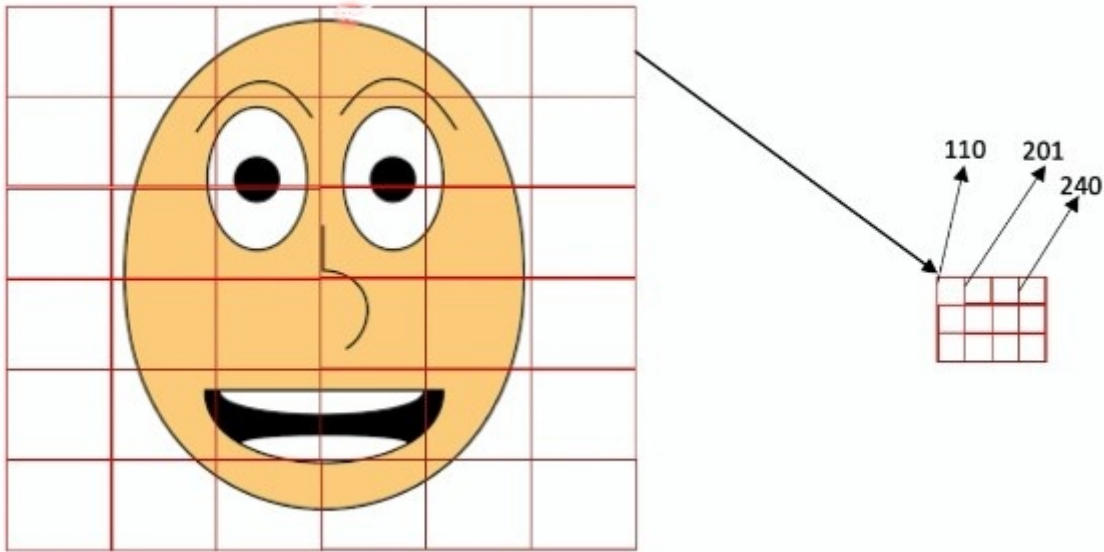
La base del cálculo de este algoritmo consiste en aplicar esta condición, seleccionando el elemento central de la matriz. Ahora necesitamos generar un valor binario. El valor binario sería 11100010. El algoritmo comenzará a aplicar la condición desde el elemento de la esquina superior izquierda, avanzando hasta el primer elemento de la segunda fila, como si estuviera formando un círculo de esta manera.

1	1	1
0	8	0
1	0	0

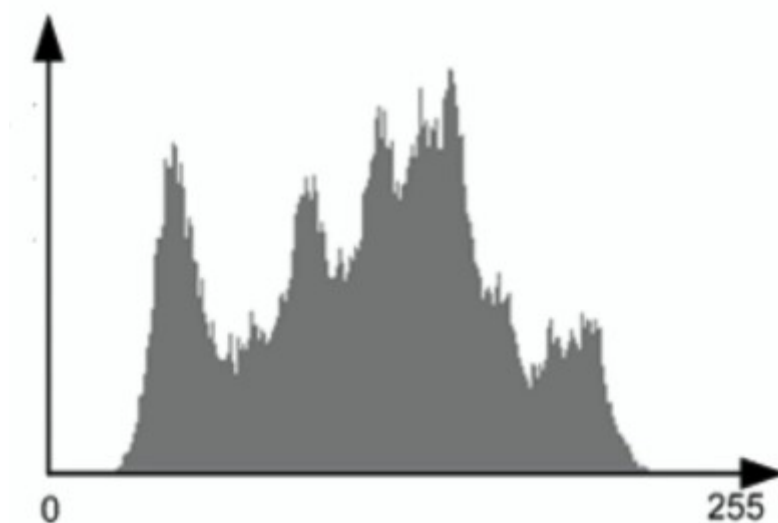
Después de convertir el valor binario al valor decimal, obtenemos un valor decimal de 226. Esto indica que todos estos píxeles alrededor del valor central son iguales a 226. Este algoritmo es robusto en términos de iluminación. Si se coloca una linterna sobre la imagen, el valor de los píxeles aumentará. Cuanto mayores sean los valores, más brillante será la imagen, y cuando los valores sean más bajos, la imagen será más oscura. Por esta razón, este algoritmo arroja buenos resultados tanto en imágenes con iluminación intensa como en imágenes oscuras, ya que cuando la imagen se vuelve más clara o más oscura, todos los píxeles en el vecindario cambiarán. Después de iluminar la imagen, la matriz se verá así. Después de aplicar la condición mencionada anteriormente, obtendremos el mismo valor binario que antes, es decir, 11100010.

42	55	48
35	38	33
38	30	32

Vamos a considerar otra imagen aquí para entender mejor cómo el algoritmo reconocerá el rostro de una persona.



Tenemos la imagen de un rostro aquí, y lo que hará el algoritmo es crear varios cuadrados, como puedes ver aquí. Y dentro de cada uno de estos cuadrados, tenemos la representación de lo anteriormente mencionado. Por ejemplo, este cuadrado aquí no representa solo un píxel, sino que está configurado con varios píxeles que son tres filas por cuatro columnas. Tres por cuatro es igual a doce píxeles en total en estos cuadrados aquí, en cada uno de estos cuadrados que son doce píxeles. Y luego aplicamos esa condición a cada uno, considerando el píxel central. El siguiente paso es crear un histograma, que es un concepto de estadística que contará cuántas veces aparece cada color en cada cuadrado. Esta es la representación del histograma.



Por ejemplo, si el valor 110 aparece 50 veces, se creará una barra en el histograma con un tamaño igual a 50; si el valor 201 aparece 110 veces, se creará otra barra en el histograma con un tamaño igual a 110. Basándose en la comparación de los histogramas, el algoritmo

podrá identificar los bordes y las esquinas de las imágenes. Por ejemplo, en este primer cuadrado aquí, no tenemos información sobre el rostro de la persona. Por lo tanto, el histograma será diferente al de este otro cuadrado que tiene el borde del rostro. En resumen, el algoritmo sabe qué histogramas representan los bordes y cuáles representan las características principales de la persona, como el color del ojo, la forma de la boca, y así sucesivamente. Entonces, esta es la teoría básica de este algoritmo, que se basa en la creación y comparación de histogramas.

## **Realización del reconocimiento facial**

Cada histograma creado se usa para representar cada imagen del conjunto de datos de entrenamiento. Dada una imagen de entrada, realizamos los pasos nuevamente para esta nueva imagen y creamos un histograma que representa la imagen. Entonces, para encontrar la imagen que coincida con la imagen de entrada, solo tenemos que comparar dos histogramas y devolver la imagen con el histograma más cercano. Podemos usar varios enfoques para comparar los histogramas (calcular la distancia entre dos histogramas), por ejemplo: distancia euclídea, chi-cuadrado, valor absoluto, etc. La salida del algoritmo es la ID de la imagen con el histograma más cercano. El algoritmo también debe devolver la distancia calculada, que puede usarse como una medida de confianza”.