



## RECONOCIMIENTO DE EMOCIONES MEDIANTE SENSOR KINECT RECOGNITION OF EMOTIONS THROUGH KINECT SENSOR

Yeison Guillermo Muñoz Pérez

Universidad Autónoma del Estado de México. Centro Universitario UAEM  
Nezahualcóyotl, México. yeisonguillermomunozperez@gmail.com  
ORCID:0000-0002-8829-3252

### RESUMEN

El estudio de las emociones ha crecido exponencialmente en los últimos años a medida que los equipos informáticos se han vuelto más sofisticados. Por ello, se han desarrollado sistemas que intentan interactuar con el usuario de una forma más natural mediante el reconocimiento de emociones a través de una red neuronal convolucional utilizando "Deep learning" para extraer las características del rostro detectado. Para este caso utilizaremos 3 modelos de reconocimiento diferentes, "EigenFaces", "FisherFaces" y "LBPH", han sido entrenados para un experimento de 200 fotos y se han obtenido resultados de una comparación entre ellos para determinar qué modelo resulta más adecuado para detectar emociones. Para el modelo "EigenFaces" se obtuvieron valores que oscilan en un rango de 2500 y 3200 (entre el valor siendo mayor significa que el valor del error es mayor), para "FisherFaces" se obtuvieron valores que oscilan en un rango de 75 y 45 en comparación con

"EigenFaces" es mucho mejor elegir usar este modelo para realizar los

experimentos, teniendo en cuenta que aun así el tiempo de entrenamiento podría ser elevado. Y finalmente tenemos "LBPH" que arrojó valores menores a 45, en comparación a los otros 2 modelos el resultado del modelo "LBPH" es "prácticamente" exacto al realizar el reconocimiento de la emoción. En conclusión "LBPH" es sin duda la mejor opción para la detección de emociones no solo por su bajo índice de error sino también por su bajo tiempo de entrenamiento.

**Palabras clave:** Emociones; reconocimiento; deep learning; modelo.

### ABSTRACT

The study of emotions has grown exponentially in recent years as computer equipment has become more sophisticated. For this reason, systems have been developed that try to interact with the user in a more natural way by recognizing emotions through a convolutional neural network using "Deep learning" to extract the characteristics of the detected face. For this case we will use 3 different recognition models, "EigenFaces", "FisherFaces" and



"LBPH", they have been trained for an experiment of 200 photos and the results of a comparison between them have been obtained to determine which model is more suitable to detect emotions. For the "EigenFaces" model, values were obtained that oscillate in a range of 2500 and 3200 (between the value being greater it means that the value of the error is greater), for "FisherFaces" values that oscillate in a range of 75 and 45 were obtained. compared to "EigenFaces" it is much better to choose to use this model to perform the experiments, considering that the training time could still be high. And finally we have "LBPH" that yielded values less than 45, compared to the other 2 models, the result of the "LBPH" model is "practically" accurate when performing emotion recognition. In conclusion "LBPH" is undoubtedly the best option for emotion detection not only because of its low error rate but also because of its low training time.

**Key words:** Emotions; recognition; deep learning; model.

Fecha de envío: 10/06/2022

Fecha de aprobación: 11/07/2022

Fecha de publicación: 01/01/2023

## INTRODUCCIÓN

Todos sabemos por experiencia propia qué son las emociones y cuán importantes son en nuestra vida, de hecho, las personas solo podemos vivir la vida emocionalmente: Siento, luego existo (Bericat Alastuey, 2012).

En el mundo de los datos y aprendizaje automático. Este concepto cae bajo la noción de los sistemas cognitivos. ¿Qué es exactamente lo que intenta lograr un algoritmo cognitivo de reconocimiento de emociones? La idea es recrear el proceso de pensamiento humano a partir de datos de entrenamiento (en forma de imágenes capturadas en tiempo real) e intentar segmentar las emociones contenidos en esos datos (Rahul, 2020).

La visión por computadora es una parte de la inteligencia artificial que se ocupa de los datos visuales. Con la llegada de uso del aprendizaje automático y los modelos de aprendizaje profundo, los sistemas informáticos actuales pueden trabajar con imágenes y videos digitales para comprender e identificar las características de su contenido (Briega, 2019).

Actualmente la inteligencia artificial ha alcanzado alturas y longitudes que antes eran inimaginables los programas y sistemas informáticos ahora pueden imitar el comportamiento, las reacciones y las respuestas humanas con gran precisión (Rahul, 2020).

Los factores responsables de hacer posible el análisis de sentimientos son el procesamiento del lenguaje natural, la lingüística computacional, la minería de textos y el análisis biométrico. Por otra parte, la ciencia cognitiva es un campo



interdisciplinario de base empírica que se ocupa del estudio de la naturaleza de la mente humana (Medina, 2008).

El ojo humano, en comparación con los sensores de que se dispone para el análisis de imágenes, presenta una resolución espacial pobre, su rango dinámico es pequeño y el rango de longitudes de onda de los rayos luminosos que es capaz de percibir es menor (Hueso, 2001).

En cualquier sistema de visión el primer elemento presente es la luz. Se estudian los diversos tipos de iluminación que existen con sus ventajas e inconvenientes (Hueso, 2001).

## MATERIALES Y MÉTODOS

En este trabajo, se procesan los resultados obtenidos mediante una aplicación de escritorio realizada en Python bajo las librerías de OpenCV, en concreto la librería *cv::CascadeClassifier* (Open Source Computer Vision, s.f.) para la detección de rostros (*haarcascade\_frontalface\_default.xml*) disponible en (Pisarevsky, 2010).

Es un enfoque basado en el aprendizaje automático en el que la función de cascada se entrena a partir de 200 imágenes positivas. Luego se usa para detectar objetos en otras imágenes.

Para este trabajo utilizaremos la detección de rostros. Inicialmente, el algoritmo necesita como mínimo 200 imágenes positivas (imágenes de rostros) para entrenar el clasificador.

Luego necesitamos extraer características de él, para esto, se utilizan las características de Haar (Figura 1) de uso y distribución libre por OpenCV.

Son como nuestro Kernel Convolutivo. Cada característica es un valor único obtenido al restar la suma de píxeles debajo del rectángulo blanco de la suma de píxeles debajo del rectángulo negro.

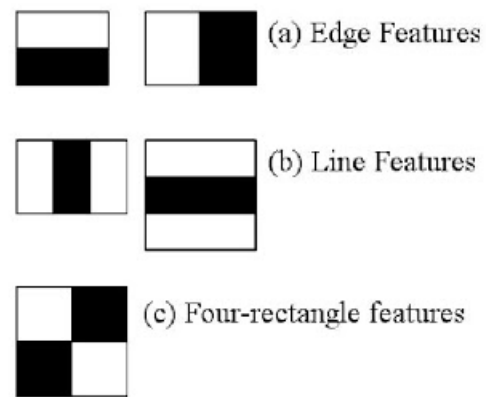


Fig1. Características Haar.  
Fuente: (Paul Viola, 2001)

Usaremos 3 tipos diferentes de métodos para detectar la emoción en el rostro retransmitiéndola en directo con el sensor Kinect. (Fretz, 2011).

Dichos métodos son:

### **EIGENFACES**

Es una técnica que permite utilizar la ortogonalidad dimensional para determinar qué vectores proporcionan más información sobre un conjunto de datos de dimensión N. Sin embargo, la información N-dimensional obtenida de *Eigenfaces* contiene datos redundantes que solo encarecen un sistema de clasificación (Franco, et al., 2015). Para minimizar



esto, se aplica análisis de componentes principales (PCA) (Gottumukkal, 2003).

### **FISHERFACES**

Es una técnica de reconocimiento facial consciente de la luz y la expresión facial. Este se encarga de clasificar y reducir el tamaño de los rostros según el método FLD (Discriminant Linear Fisher). (Franco, et al., 2015).

### **LBPH**

El método de patrones binarios locales fue diseñado para la descripción de texturas (Silva, et al., 2012). El uso de descripciones locales en algunas áreas de la cara aporta más información que otras, por lo que los descriptores de textura tienden a promediar la información que describen, lo que no es práctico cuando se describen caras, ya que es importante retener información sobre relaciones espaciales (Alvarado & Fernandez, 2012).

### **DESARROLLO**

Esta aplicación fue codificada en 3 partes: la primera de registro, la segunda de entrenamiento y la tercera de detección. El programa está cargado en una laptop con sistema operativo Windows 11 y se usa el sensor Kinect (Fretz, 2011) como cámara web.

### **ETAPA DE REGISTRO**

En esta etapa, el individuo se presenta frente al sensor Kinect el cual actúa como una “cámara virtual” generada gracias al programa OBS

Studio (versión 27. 2) y una luz artificial en posición direccional con el fin de crear sombras en el objeto (individuo) y aumentar el contraste de las partes tridimensionales con respecto al fondo (Hueso, 2001).

Una vez colocado de esta forma, se procede a ejecutar el programa una vez por cada emoción para este caso; Felicidad, Tristeza, Ira y Sorpresa, para cada emoción se creó un registro en una carpeta de 200 fotos para la creación de la base de datos con la cual posteriormente en la fase de reconocimiento se hará la comparación, las imágenes capturadas se clasifican con por medio de “haarcascade\_frontalface\_default.xml” que contiene los datos necesarios para procesar las caras. Utilizando un clasificador Haar proporcionado por las librerías de OpenCV, el cual es un método de detección de objetos (Paul Viola, 2001), el clasificador funciona con un enfoque basado en el aprendizaje automático de las máquinas donde la función de cascada se entrena a partir de una gran cantidad de imágenes positivas y negativas y se utiliza para reconocer objetos en las imágenes para nuestro caso, reconocimiento de emociones (Julian, et al., 2017)

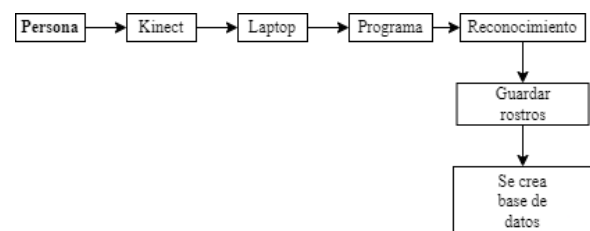


Fig2. Etapa de Registro



Fuente: elaboración propia

## ETAPA DE ENTRENAMIENTO

La ejecución del programa para la segunda etapa utiliza algunas bibliotecas de Python (Os, Numpy y time). Esta última la utilizamos para medir el tiempo que necesita cada reconocedor para su respectivo entrenamiento, se van tomando las 200 imágenes que tenemos en nuestra carpeta de “Data” y así obtener nuestro modelo para realizar la comparación posteriormente, en la etapa 3, estos 3 modelos son almacenados en nuestra carpeta raíz.

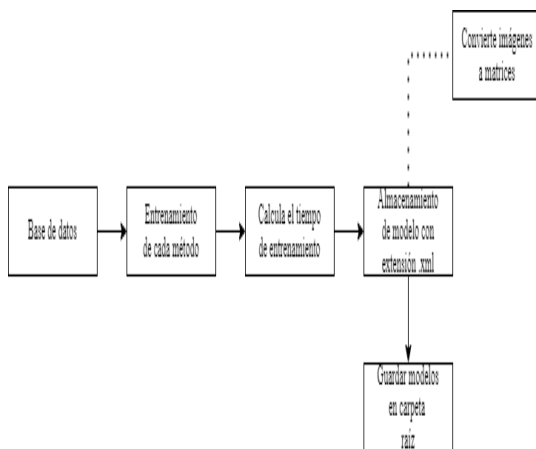


Fig3. Etapa de Entrenamiento  
Fuente: elaboración propia

## ETAPA DE RECONOCIMIENTO

En esta tercera y última etapa se realiza el reconocimiento de las emociones en la persona y se realiza con los 3 modelos obtenidos en la etapa anterior:

- Se abre el modelo obtenido.
- Se va a comparar con las imágenes de la base de datos, las cuales se convierten a escala de grises para trabajar con menos información de los

píxeles y sobre todo reconocer más detalles (Julian, et al., 2017).

Después el sensor Kinect entra en acción y comienza a capturar las imágenes en retransmisión en vivo y el algoritmo comienza a comparar estas imágenes con los modelos obtenidos entregando un número de fidelidad para cada uno de ellos.

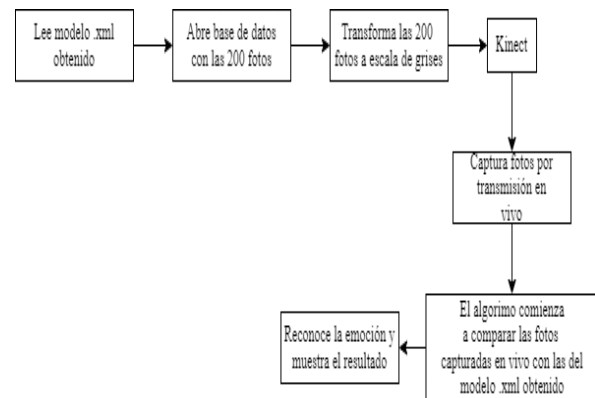


Fig4. Etapa de Reconocimiento  
Fuente: Elaboración propia

## RESULTADOS

### Funcionamiento del programa

Se realiza el experimento con las siguientes características:

- Iluminación direccional (Más contraste y detalle en la persona)
- Kinect, tiene una cámara de captura de movimiento VGA con una resolución de 640 x 480 píxeles a 30 FPS y una cámara dual de profundidad 3D de 640 x 480 píxeles a 30 FPS. (Fretz, 2011)
- Python 3.8.2
- OpenCV 4.5.5
- Laptop Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz 2.30 GHz



RAM: 8GB

Presentación de la primera parte del programa (cuando esté en funcionamiento), para realizar este paso es necesario verificar que el sensor Kinect esta en línea y listo para su uso, en la figura 5 vemos que se extraen las 200 fotos para la primera emoción a detectar (Enojo) y así generar nuestra base de datos.

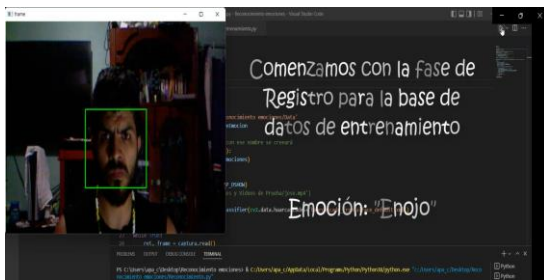


Fig5. Captura de las 200 fotos para “Enojo”

Segunda emoción para detectar (Felicidad)

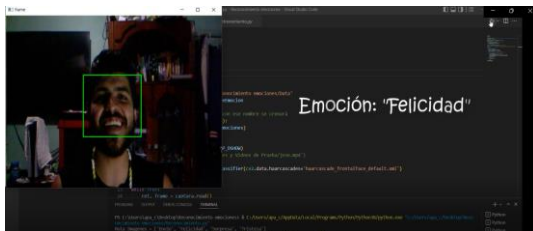


Fig6. Captura de las 200 fotos para “Felicidad”

Fuente: elaboración propia

Tercera emoción para detectar (Sorpresa)

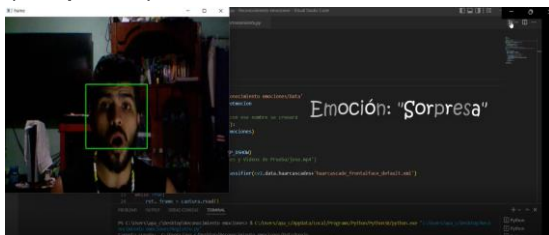


Fig7. Captura de las 200 fotos para “Sorpresa”

Cuarta emoción para detectar (Tristeza)

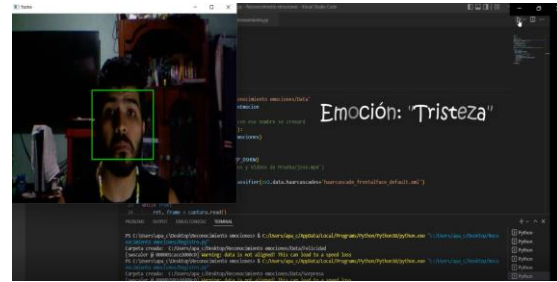


Fig8. Captura de las 200 fotos para “Tristeza”

En la segunda parte del programa se observa que sirve para el entrenamiento de y creación de los modelos .xml con la ayuda de características haar (Paul Viola, 2001) los cuales utilizaremos para realizar la comparación en la última fase con las imágenes obtenidas en tiempo real, así como el tiempo que tardo cada método para la obtención de dichos modelos .xml.

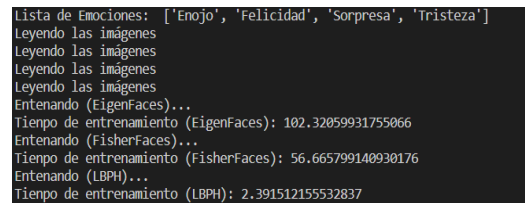


Fig9. Entrenamiento de modelos y tiempo(s) de demora para la obtención del modelo .xml

En la última fase de nuestro programa lo ejecutamos con cada uno de los modelos obteniendo así, diferentes valores de fidelidad.

Para “EigenFaces”





Fig10. Fidelidad baja para el modelo "EigenFaces"

### Para "FisherFaces"



Fig11. Fidelidad media para el modelo "FisherFaces"

### Para "LBPH"

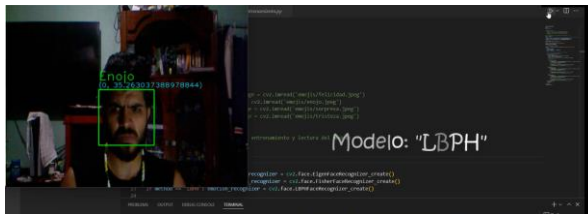


Fig12. Fidelidad alta, casi perfecta para el modelo "LBPH"

## DISCUSIÓN

En el presente trabajo, se realizó un experimento con 200 fotos para entrenar a cada uno de los modelos no se trabajó con más debido al alto consumo de recursos que 2 de los modelos utilizan (EigenFaces y FisherFaces) el tiempo de entrenamiento de dichos modelos es demasiado alto y si no se cuenta con un equipo lo suficientemente potente para llevar a cabo el entrenamiento con más fotos, por ejemplo 800 fotos el resultado podría ser el siguiente:

```
Lista de Emociones: ['Enojo', 'Felicidad', 'Sorpresa', 'Tristeza']
Leyendo las imágenes
Leyendo las imágenes
Leyendo las imágenes
Leyendo las imágenes
Entrenando (EigenFaces)...
Tiempo de entrenamiento (EigenFaces): 3748.965019464493
Entrenando (FisherFaces)...
Tiempo de entrenamiento (FisherFaces): 4867.105513572693
Entrenando (LBPH)...
Tiempo de entrenamiento (LBPH): 6.389079809188843
PS C:\Users\yapa_c\Desktop\Reconocimiento emociones> █
```

Fig13. Entrenamiento de modelos y tiempo(s) de demora para la obtención del modelo .xml para un experimento de 800 fotos

El tiempo está expresado en segundos, por lo tanto, no hay un número de fotos adecuado para la obtención de mejores resultados para estos dos modelos.

Podemos observar que, si varía mucho el resultado entre los tres modelos de reconocimiento, para "EigenFaces" se tiene como fidelidad valores que oscilan en un rango de 2500 y 3200, lo cual nos indica que la detección de emociones no es tan preciso para este experimento de 200 fotos.

Para el segundo modelo de reconocimiento "FisherFaces", se tiene como valores de fidelidad que oscilan en un rango de 75 y 45, lo que sugiere que la detección es casi "perfecta" en comparación a "EigenFaces", sin embargo, para un experimento con más fotos y diferentes tipos iluminación, la realidad sería diferente, por lo que tal vez se puedan lograr mejores resultados, incluso si el periodo de entrenamiento fuera largo.

Por último, tenemos el tercer modelo de reconocimiento "LBPH", se tiene como valores de fidelidad de <45 lo cual para este experimento de 200 fotos y la iluminación direccional concluye que la detección de la emoción es en realidad "perfecta" y bastante exacta, casi no se tiene un valor de error.



Por lo que se concluye que el modelo de reconocimiento “LBPH” es la mejor opción para este y cualquier experimento, debido a su mínimo valor de error y tiempo de entrenamiento.

## REFERENCIAS

Alvarado, J. & Fernandez, J., (2012). Análisis de textura en imágenes a escala de. ENGI Revista Electrónica De La Facultad De Ingeniería, 1(1).  
Bericat Alastuey, E., (2012). Emociones. sociopedia.isa, pp. 1-13.

Briega, R., (2019). La Era de las máquinas inteligentes. [En línea] Available at: <https://iaarbook.github.io/vision-por-computadora/> [Último acceso: 1 Junio 2022].

Franco, C. H., Ospina, C., Cuevas, E. & Capacho, D., (2015). Reconocimiento facial basado en eigenfaces, lbhp y fisherfaces en la beagleboard. Revista Colombiana de Tecnologías de Avanzada.

Fretz, M., (2011). Ventana Interactive Shop Window. IAD Interaction Desing.  
Gottumukkal, R. a. A. V., (2003). Real time face detection from color video stream based on PCA method. IEEE, pp. 146-150.

Hueso, A. d. I. E., (2001). Visión por Computador. Madrid: Prentice Hall.  
Julian, F. G., V. Reyes, M., L. Sánchez, A. & J. Ríos, C. A., (2017). Reconocimiento facial por el método de Pistas Educativas, 39(127).

Medina, N., (2008). La ciencia cognitiva y el estudio de la mente. IIPSI, 11(1), pp. 183-198.  
Open Source Computer Vision, s.f. Cascade Classifier. [En línea] Available at: [https://docs.opencv.org/3.4/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html) [Último acceso: 2 Junio 2022].

Paul Viola, M. J., (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE, Volumen 1, pp. I-I.  
Pisarevsky, V., (2010). HAARcascades. [En línea] Available at: [https://github.com/kipr/opencv/blob/master/data/haarcascades/haarcascade\\_frontalface\\_default.xml](https://github.com/kipr/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml) [Último acceso: 3 Junio 2022].

Rahul, (2020). La guía definitiva para el reconocimiento de emociones a partir de expresiones faciales con Python. [En línea] Available at: <https://ichi.pro/es/la-guia-definitiva-para-el-reconocimiento-de-emociones-a-partir-de-expresiones-faciales-con-python-110937080276223>

Silva, E., Esparza, C. & Mejía, Y., (2012). POEM-based facial expression recognition, a new approach. IEEE, pp. 162-167