

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITÉCNICO  
NACIONAL  
ROBÓTICA Y MANUFACTURA AVANZADA



**Cinvestav**  
Unidad Saltillo

Visión por Computadora

---

## PROYECTO

# Hand Gesture Detection and Recognition

---

Integrantes:  
Hilario Acuapan Gabriela  
Pineda Gómez Luis Alberto

RAMOS ARIZPE, COAHUILA, APRIL 2023

## Contents

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Fundamentos teóricos</b>	<b>3</b>
2.1	Análisis de componentes principales . . . . .	3
2.2	Reconocimiento de gestos de mano . . . . .	5
<b>3</b>	<b>Desarrollo de la metodología</b>	<b>6</b>
3.1	Preprocesamiento de las imágenes de entrada para la formación de la base de datos . . . . .	6
3.2	Adquisición de datos . . . . .	9
3.3	Training set . . . . .	9
3.4	Eigenhands con PCA . . . . .	10
3.5	Clasificación . . . . .	13
<b>4</b>	<b>Resultados y discusión</b>	<b>13</b>
4.1	K-ésima clase . . . . .	14
4.2	Nueva imagen de entrada . . . . .	14
<b>5</b>	<b>Conclusiones</b>	<b>14</b>

## 1 Introducción

El objetivo principal de este trabajo es detectar y reconocer los gestos de mano en 2D (en imágenes) a partir de su forma, comparándolos con gestos existentes en una base de datos. Para cumplir este objetivo se utiliza un algoritmo para la detección y reconocimiento de gestos basado en los componentes principales (PCA). El algoritmo a implementar se describe en el artículo de Turk y Pentland [3] (*Face Recognition Using Eigenfaces*), pero en lugar de aplicarlo a la detección de rostros humanos, se utilizará para la detección de gestos de mano.

El algoritmo de *Eigenfaces*, fue presentado por primera vez por Sirovich y Kirby en su artículo de 1987 [2], y posteriormente formalizado por Turk y Pentland en su artículo de 1991 [3]. El procedimiento general indica que cada cara se almacena en un vector de dimensión  $N^2$ . Y el Análisis de los Componentes Principales (PCA) se utiliza para encontrar un subespacio de dimensión  $M$  cuyos vectores de la base corresponden a las direcciones de máxima varianza en el espacio original de la imagen ( $N^2$ ). Este nuevo subespacio es normalmente de dimensión más baja ( $M \ll N^2$ ).

EL enfoque del algoritmo es descomponer las imágenes de los rostros en un conjunto más pequeño de rasgos característicos llamados "eigenfaces", que pueden considerarse como los componentes principales de la base de datos original. Para la parte del reconocimiento se proyecta una nueva imagen en el subespacio generado por las eigenfaces (llamado "face space"), para luego clasificar el rostro comparando su posición en el *face space* de las personas conocidas (o clases conocidas).

Este trabajo aquí presentado trata de replicar este procedimiento pero en lugar de reconocer rostros, se utilizará para reconocer y clasificar gestos de la mano. . . .

## 2 Fundamentos teóricos

### 2.1 Análisis de componentes principales

El objetivo de implementar un análisis por componentes principales es realizar una **reducción de dimensionalidad** ó del número de variables en una base datos. Sin embargo, el realizar este análisis viene acompañado de un precio a pagar y esto es la relación **precisión - simplicidad**. Por un lado, al reducir la dimensión de nuestra base de datos, el realizar cómputos con ella será mucho más rápido y eficiente, además de que la interpretación de los datos es más sencilla a expensas de perder precisión y exactitud de la base original.

El análisis por componentes principales puede ser implementado en 5 pasos:

#### Paso 1.- Estandarización

En este primer paso realizamos una estandarización de los datos, esto con el objetivo de evitar que valores grandes, por ejemplo, que se encuentren en un rango de (100 a 1000) no dominen entre valores menores, por ejemplo (0 a 10). Esto es para evitar que nuestra base de datos presente lo que conocemos como **sesgo o bias**.

La estandarización de los datos se logra aplicando la siguiente expresión 1:

$$Z = \frac{\text{valor} - \text{media}}{\text{desviacion estandar}} \quad (1)$$

#### Paso 2.- Obtención de la matriz de covarianza

El propósito por el cual se calcula la matriz de covarianza es para obtener la relación existente entre las vari-

ables de la base de datos. Se busca encontrar cómo es que varían entre ellas. En algunas ocasiones, las variables se encontrarán relacionadas de tal forma, que tendrán información **redundante**, esto es, que varias variables contengan información similar. Este tipo de relaciones son las que se encuentran en la matriz de covarianza.

La matriz de covarianza es una matriz **simétrica** de dimensiones  $p \times p$ . Por ejemplo, supongamos que tenemos una matriz de covarianza de  $3 \times 3$ :

$$\begin{bmatrix} Cov(x, x) & Cov(x, y) & Cov(x, z) \\ Cov(y, x) & Cov(y, y) & Cov(y, z) \\ Cov(z, x) & Cov(z, y) & Cov(z, z) \end{bmatrix}$$

Donde los elementos de la diagonal de la matriz  $Cov(x, x) = Var(x)$  y también es importante mencionar que  $Cov(x, y) = Cov(y, x)$ .

La información obtenida por la matriz de covarianza viene dada principalmente por el **signo**. Esto es:

- Si es **Positivo**: Las dos variables incrementan o decrementan de la misma forma (Correlacionadas).
- Si es **Negativo**: Una variable incrementa cuando la otra decremente (Inversamente correlacionadas).

### Paso 3.- Obtención de los eigenvalores y eigenvectores de la matriz de covarianza

Esto se realiza con el objetivo de obtener los **componentes principales** de los datos. En los componentes principales, se encuentra contenida la información principal o más significativa.

Por ejemplo, en la figura 1, se muestra una gráfica de la varianza dado el número de componentes principales. En este caso, en el componente principal de mayor importancia se encuentra codificada el 40% de la covarianza. Esto es que dicho componente contiene información *única*.

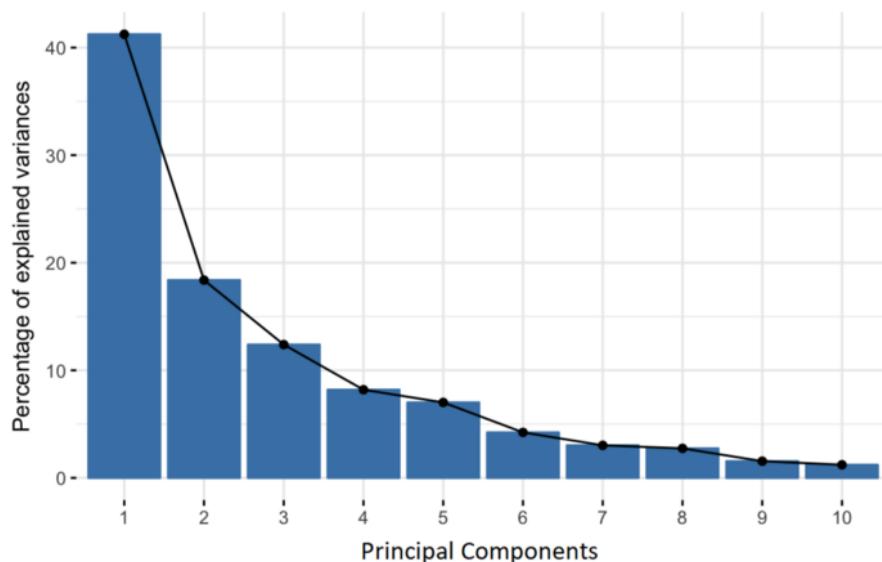


Figure 1: Comparación del Porcentaje de la varianza vs el número de componentes principales

Con este procedimiento, reducimos la dimensionalidad de la base de datos original sin tener un impacto sustancial en la pérdida de información. La parte sustancial del PCA, es que los componentes principales **carecen** de interpretación. Esto quiere decir que no se le puede atribuir cierto atributo o explicación a un componente principal.

Debido a que existe tantos componentes principales como número de variables en la base de datos, el PCA se construye de tal forma que el componente primario contenga la mayor cantidad de varianza posible de la base de datos. El componente secundario contendrá la mayor cantidad de varianza posible, pero que no se encuentra en la primer componente y así sucesivamente con el resto de componentes.

Los **eigenvectores** juegan el papel de indicar los ejes en **dónde** se encuentra la mayor cantidad de varianza, esto es, la mayor cantidad de información de la base de datos. Mientras que los **eigenvalores**, dan la **cantidad** de varianza que contiene cada componente principal.

#### Paso 4.- Construcción del Vector característico

En este paso, es donde se toma la decisión de conservar o descartar aquellos **eigenvectores** que no aporten información relevante de la base de datos. Con los vectores que deseemos conservar, se construirá una nueva matriz, conformada por los vectores columna de los eigenvectores más representativos.

#### Paso 5.- Reconformar la base de datos a lo largo de los ejes de los componentes principales

Para este último paso, tenemos que **reorientar** los datos a lo largo de los nuevos eigenvectores en donde se encuentra concentrada la mayor cantidad de información. Esto se obtiene de la siguiente forma:

$$\text{Final data set} = \text{Matriz caracteristica}^T * \text{Data set estandarizado}$$

## 2.2 Reconocimiento de gestos de mano

### 3 Desarrollo de la metodología

#### 3.1 Preprocesamiento de las imágenes de entrada para la formación de la base de datos

Antes de poder ingresar las imágenes a nuestro sistema, es necesario el "estandarizarlas". Para ello, se hace uso del la figura 2, en la cual se muestra de forma sencilla las etapas que conforman el preprocesamiento. El propósito por el cual se estandarizan las imágenes es para trabajar con frames cuadrados, esto tiene ventajas al momento de realizar operaciones y simplificar considerablemente el código.

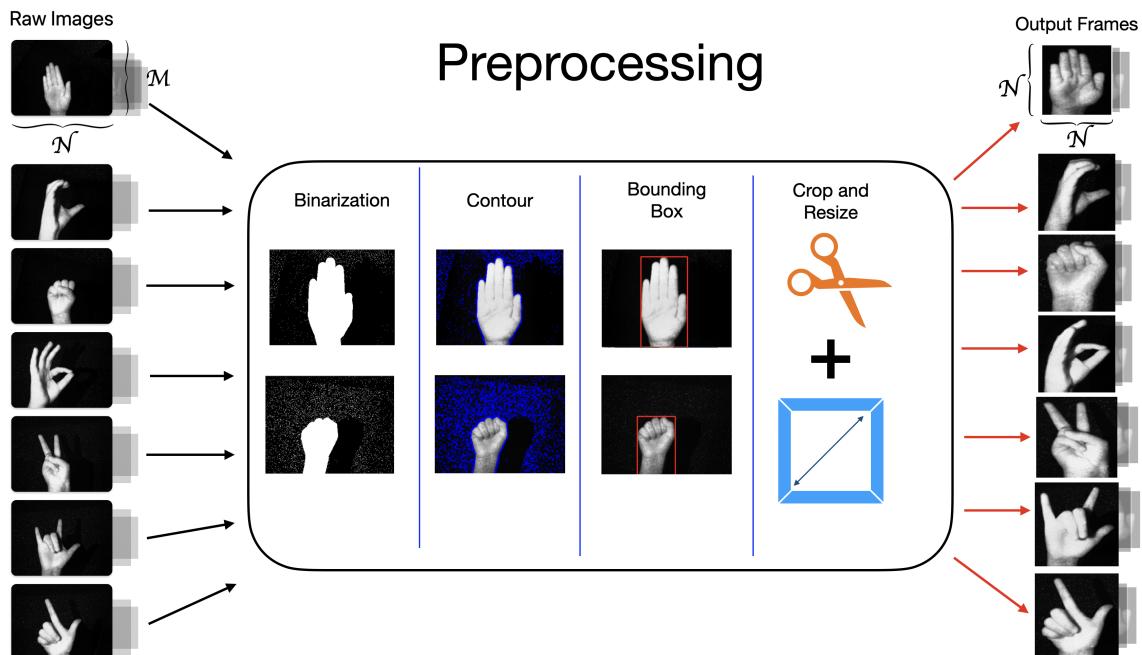


Figure 2: Preprocesamiento de las imágenes para conformar la base de datos del sistema.

Comenzamos con un conjunto de imágenes sin procesar, las cuales denotamos como **Raw Images**. En dicho conjunto, las imágenes tienen dimensiones  $M \times N$ . Como primer paso, se realiza una **binarización**, esto con el objetivo de encontrar los **contornos de la imagen**. Posteriormente, ya con los contornos detectados, procedemos a delimitar el contorno más importante mediante un **bounding box**. Para finalmente recortar la imagen que se encuentra contenida dentro del bounding box y realizar un **escalamiento**. Esto con el propósito de obtener imágenes cuadradas y enfocarnos únicamente en el gesto.

##### Paso 1.- Binarización

La operación de binarización juega un papel fundamental en el preprocesamiento, debido a que se tiene que elegir un valor de umbral o **threshold** para permitir conservar la mayor cantidad de información de cada frame pero filtrando el mayor ruido posible. En la figura 3, se muestra la binarización de un frame. Es importante mencionar que el valor del threshold o umbral fue elegido de manera experimental.

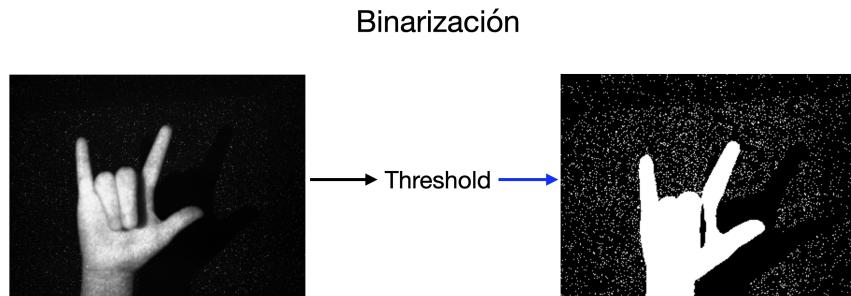


Figure 3: Binarización de las imágenes de entrada

#### Paso 2.- Contornos

En la siguiente etapa, encontramos los contornos tomando como base la imagen binarizada. Es importante mencionar que el contorno más relevante será aquel que contenga la mayor cantidad de elementos. En la figura 4, se muestran los contornos de la imagen binarizada.

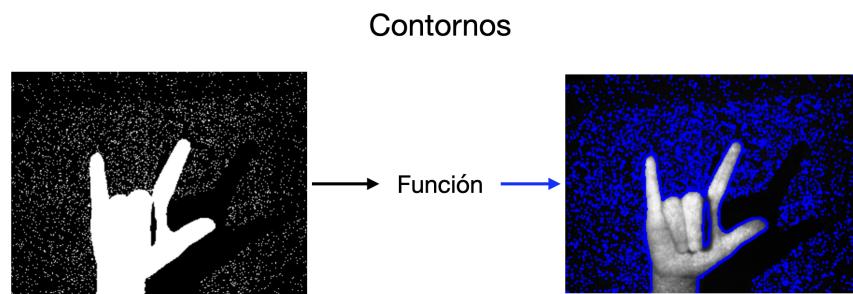


Figure 4: Contornos de la imagen binarizada

#### Paso 3.- Bounding Box

En la penúltima etapa, se implementa una función para generar un bounding box el cual contiene la información del contorno más relevante. Esto se muestra en la figura 5.

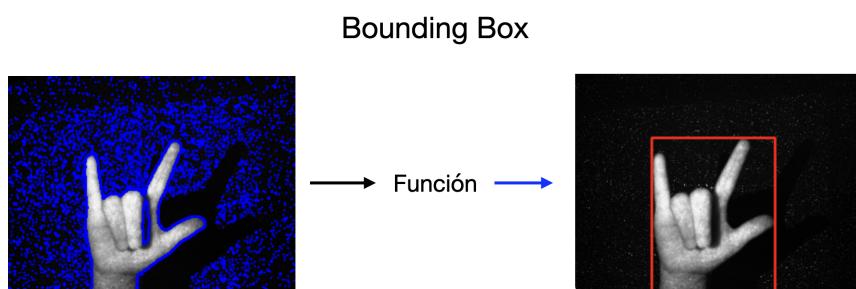


Figure 5: Contornos de la imagen binarizada

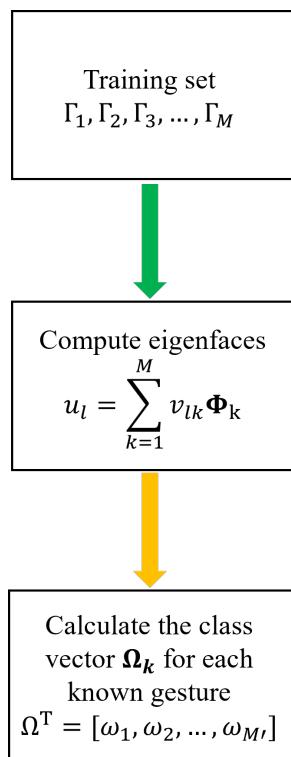


Figure 6: Operaciones de inicialización.

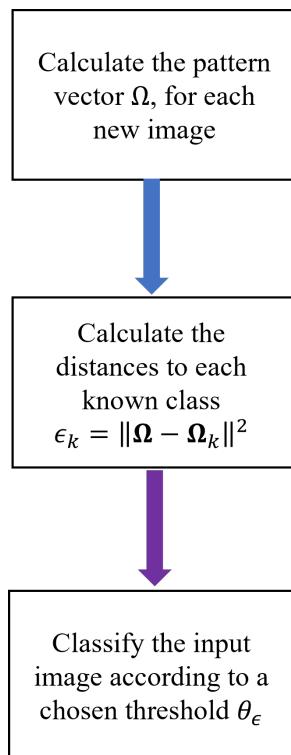


Figure 7: Operaciones para clasificación.

### 3.2 Adquisición de datos

La base de datos utilizada para este proyecto se creó a partir de la captura de imágenes en infrarrojo, utilizando el Sensor mostrado en la figura 8.



Figure 8: Structure sensor 3D [1].

### 3.3 Training set

Se utilizaron  $M = 70$  imágenes para el conjunto de entrenamiento, donde se incluyeron los 7 gestos de mostrados en la figura 9.



Figure 9: Gestos para el set de entrenamiento.

Para seguir con el proceso es necesario realizar un preprocesamiento de las imágenes de la base de datos de la figura 9. Este consiste en realizar una binarización de las imágenes, para encontrar el contorno de la mano. Posteriormente dibujar el bounding box sobre la imagen original para recortarla y finalmente realizar un reescalamiento de la imagen a 200x200 píxeles.

Denotando al conjunto de entrenamiento como

$$\Gamma_1, \Gamma_2, \dots, \Gamma_M \quad (2)$$

Cada imagen de dimensiones  $N \times N$  se convierte en un vector  $\Gamma$  de dimensión  $N^2$

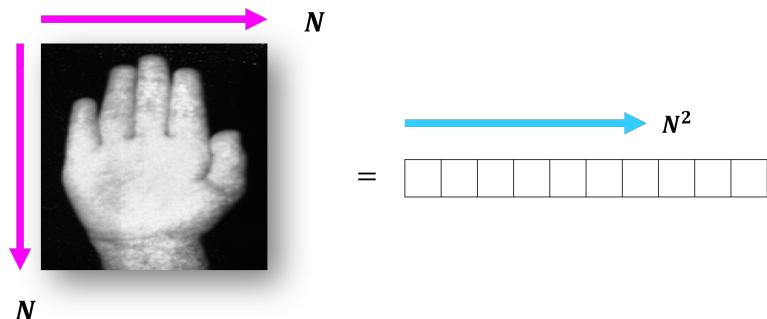


Figure 10: Vector imagen.

Estos vectores forman el conjunto de entrenamiento (2),

Figure 11: Conjunto de entrenamiento.

### 3.4 Eigenhands con PCA

Teniendo los vectores del conjunto (2), se obtiene la mano promedio (figura 12) mediante la siguiente operación

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \quad (3)$$

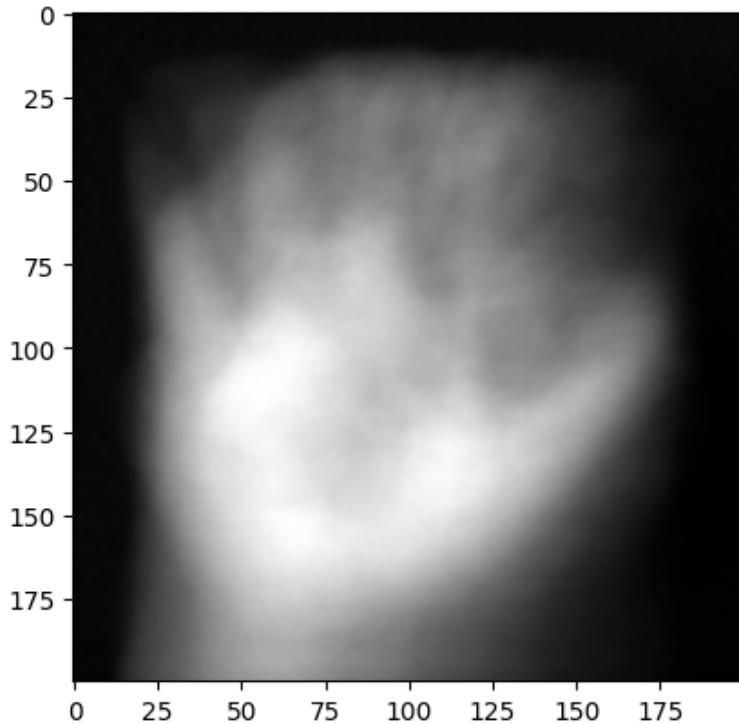


Figure 12: Mano promedio.

Posteriormente se calcula la desviación  $\Phi$  que tiene cada  $\Gamma$  con respecto al promedio  $\Psi$

$$\Phi = \Gamma_i - \Psi \quad (4)$$

Este conjunto de vectores  $\Phi$  se somete a un análisis de componentes principales (PCA), que busca un conjunto de  $M$  vectores ortonormales,  $u_n$ , que mejor describen la distribución de los datos.

Para esto definimos una matriz  $A$  que contiene a los vectores de las desviaciones  $\Phi$

$$A = [\Phi_1, \Phi_2, \dots, \Phi_M] \quad (5)$$

Para construir una matriz  $L$  de dimensiones  $M \times M$

$$L = A^T A \quad (6)$$

Y encontrar los eigenvalores  $\mu_i$  y eigenvectores  $v_i$  de la matriz  $L$ , tal que

$$Av_i = \mu_i v_i \quad (7)$$

Se deben encontrar los  $M'$  eigenvectores  $v_i$  asociados a los eigenvalores  $\mu_i$  más representativos. Para esto obtenemos la suma acumulada y observamos que con 40 eigenvectores reconstruimos casi el 90% de la información (figura 13)

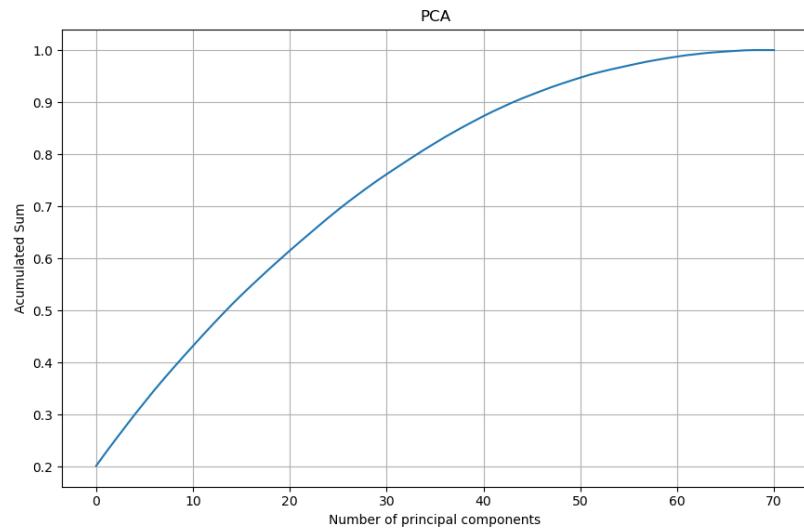


Figure 13: Suma acumulada de eigenvalores de  $L$ .

Estos vectores determinan combinaciones lineales de las imágenes del conjunto de entrenamiento  $M$ , y se utilizan para formar a las eigenhands  $u_i$ .

$$u_i = \sum_{k=1}^M v_{lk} \Phi_k, \quad l = 1, \dots, M \quad (8)$$

En la figura 16 se muestran 10 de las 40 eigenhands principales derivadas de las imágenes de entrenamiento.

### Primeras 10 eigenhands obtenidas

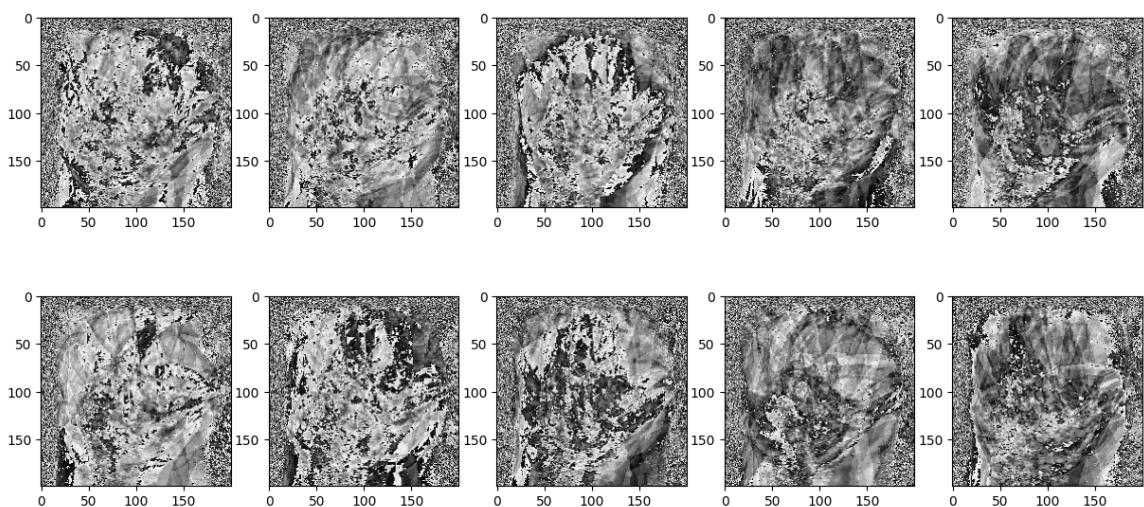


Figure 14: Primeras 10 eigenhands más representativas.

### 3.5 Clasificación

Las imágenes de eigenhands calculadas a partir de los vectores propios de  $L$  abarcan un conjunto básico (de dimensión  $M' \times M'$ ) con el que describir el conjunto de imágenes inicial.

Para realizar la clasificación es necesario encontrar los patrones  $\Omega_K$  que describen a la  $k$ -ésima clase de gestos.

$$\Omega^T = [\omega_1, \omega_2, \dots, \omega_{M'}] \quad (9)$$

Los pesos que forman al vector  $\Omega$  describen la contribución de cada eigenfaces en la representación de la imagen de la cara de entrada, y se calculan mediante la siguiente operación

$$\omega_k = u_k^T (\Gamma - \Psi) \quad (10)$$

Las clases de los gestos se calculan promediando los patrones de pesos  $\Omega$  de un pequeño número de imágenes de cada gesto.

Para determinar a qué clase pertenece una nueva imagen de entrada se calcula su patrón de pesos  $\Omega$ , para luego utilizar la distancia euclídea

$$\epsilon_k = \|\Omega - \Omega_k\|^2 \quad (11)$$

La imagen se clasifica como perteneciente a la clase  $k$  cuando  $\epsilon_k$  está por debajo de algún umbral elegido  $\theta_k$

## 4 Resultados y discusión

En esta sección se muestran los resultados obtenidos de la etapa de clasificación

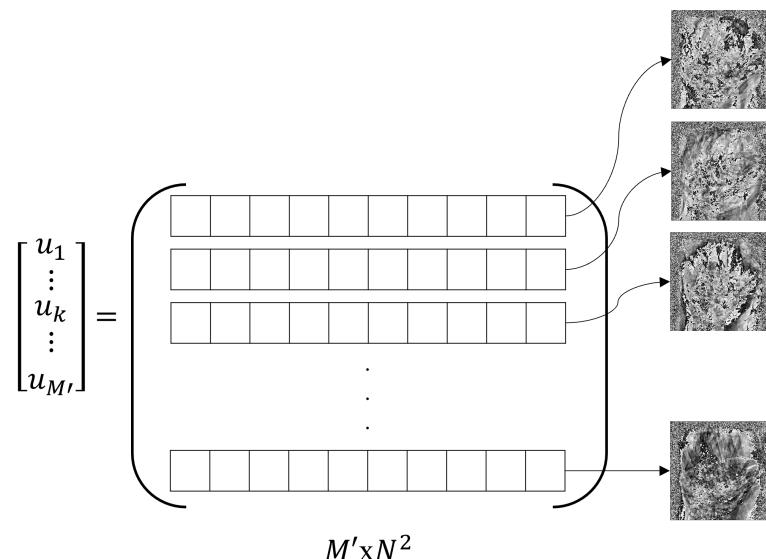


Figure 15: Eigenvectores.

#### 4.1 K-ésima clase

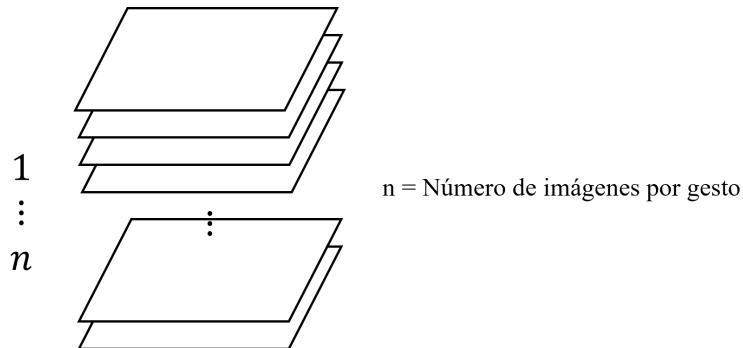


Figure 16: Imágenes por gesto.

$$\begin{aligned}\Omega_1 &= [\omega_1, \omega_2, \dots, \omega_{M'}] \\ \Omega_2 &= [\omega_1, \omega_2, \dots, \omega_{M'}] \\ &\vdots \\ \Omega_n &= [\omega_1, \omega_2, \dots, \omega_{M'}]\end{aligned}\tag{12}$$

#### 4.2 Nueva imagen de entrada

Obtener su vector de pesos

$$\Omega = [\omega_1, \omega_2, \dots, \omega_{M'}]\tag{13}$$

Obtener la distancia euclíadiana mínima según la ecuación (11)

$$\Omega_k = \frac{\Omega_1 + \Omega_2 + \dots + \Omega_n}{n}\tag{14}$$

## 5 Conclusiones

## References

- [1] Structure - the world's leading healthcare 3d scanning platform. Consultado el 10 de abril de 2023. [Online]. Available: <https://structure.io/>
- [2] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Josa a*, vol. 4, no. 3, pp. 519–524, 1987.
- [3] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.