



## Tarea 3: Parte 2

Ingeniería en Computadores

CE 5201: Procesamiento y Análisis de Imágenes Digitales

Profesor: Juan Pablo Soto

Nasser Santiago Brown Aparicio 2019043776

Kenichi Hayakawa Bolaños 2020044884

Jose Antonio Retana Corrales 2020144743

**Pregunta 1:**

**Artículo Científico 1: An SVD-based grayscale image quality measure for local and global assessment de A. Shnayderman, A. Gusev y A. Eskicioglu.**

**Aplicación:**

En este primer artículo científico, el uso específico de la descomposición en valores singulares (SVD) en el procesamiento de imágenes se centra en la evaluación de la calidad de las imágenes en escala de grises, tanto a nivel local como global. Este trabajo propone una medida de calidad de imagen que utiliza la SVD para analizar la distorsión de imágenes sometidas a diferentes tipos de ruido y compresión.

La SVD se emplea aquí para evaluar la diferencia entre los valores singulares de bloques específicos de la imagen original y sus versiones distorsionadas. Esta técnica permite identificar y cuantificar la distorsión en la imagen, generando mapas de distorsión en los cuales se puede visualizar el tipo, la cantidad y la distribución del error. Esta medida de calidad, basada en SVD, también permite obtener un valor numérico que resume el grado de distorsión en la imagen completa.

**Formulación Matemática:**

La formulación matemática del método presentado en el artículo para evaluar la calidad de una imagen en escala de grises se basa en la descomposición en valores singulares (SVD) de matrices que representan bloques de la imagen.

Primero se explica la descomposición en valores singulares (SVD) de la siguiente forma: para cualquier matriz real  $A$  de dimensiones  $m \times n$ , su descomposición en valores singulares se expresa como:

$$A = USV^T$$

donde  $U$  y  $V$  son matrices ortogonales, y  $S$  es una matriz diagonal de  $m \times n$  con valores singulares en la diagonal (ordenados de mayor a menor).

En el contexto de imágenes, cada imagen en escala de grises se representa como una matriz  $A$  donde cada elemento corresponde a la intensidad de un píxel en una posición específica. Al aplicar SVD, los valores singulares en la matriz  $S$  representan la energía o importancia de la información contenida en la imagen; los valores mayores suelen corresponder a las características dominantes de la imagen, mientras que los menores reflejan detalles más finos o ruido.

A partir de lo anterior, surge una medida de calidad basada en SVD. Para evaluar la calidad de una imagen distorsionada en comparación con su versión original, el método propuesto descompone ambas imágenes en bloques más pequeños (típicamente de  $8 \times 8$  píxeles) y calcula la SVD de cada bloque. La razón para dividir en bloques es obtener una evaluación más detallada y localizada de la distorsión, dado que los artefactos de compresión o ruido afectan áreas específicas de la imagen en lugar de afectar uniformemente.

Para medir la distorsión en cada bloque, se compara la diferencia entre los valores singulares de la imagen original y la imagen distorsionada, específicamente:

$$D_i = \text{Sqrt} \left[ \sum_{i=1}^n (s_i - \hat{s}_i)^2 \right]$$

Donde  $s_i$  y  $\hat{s}_i$  son los valores singulares i-ésimos del bloque original y el bloque distorsionado respectivamente, y  $n$  es el número de valores singulares considerados (correspondiente al tamaño del bloque).

Este cálculo para cada bloque proporciona una medida gráfica de la calidad de la imagen, llamada mapa de distorsión. Al visualizar estos valores en un mapa, es posible ver las áreas de la imagen donde la distorsión es más severa y el tipo de artefactos introducidos.

El artículo también menciona una medida numérica de distorsión global, en donde para obtener una medida numérica general que resuma la distorsión de toda la imagen, los valores de distorsión de todos los bloques se combinan en una media.

$$MSVD = \frac{\sum_{i=1}^{(k/n) \times (k/n)} |D_i - D_{mid}|}{(\frac{k}{n}) \times (\frac{k}{n})}$$

donde  $D_{mid}$  representa el punto medio de todos los  $D_i$  sorteados,  $k$  es el tamaño de la imagen, y  $n$  es el tamaño de bloque.

A partir de lo anterior, se concluye que la medida gráfica consistentemente muestra el tipo y cantidad de distorsión, además de la distribución de error en imágenes. La medida numérica o global obtiene una estimación global de la distorsión en la imagen, y ninguno de los dos requiere un modelo simplificado de HVS, es decir, de un sistema de visualización humana (human visual system), por lo que no requieren de asumir temas como distancia de visualización.

La medida basada en SVD es robusta frente a distintos tipos de distorsiones y niveles de ruido, ya que el análisis de los valores singulares permite capturar tanto las características globales como los detalles específicos de la imagen. Además, al realizar el cálculo de manera independiente en cada bloque, el método logra una alta precisión en la identificación y cuantificación de los artefactos visuales en áreas específicas.

### Pseudocódigo:

Pseudocódigo del Algoritmo de Evaluación de Calidad de Imagen Basado en SVD

Fase 1: Cálculo de la Medida Gráfica de Calidad de Imagen

# Entradas:

# I\_original: Imagen original en escala de grises

# I\_distorsionada: Imagen distorsionada en escala de grises

# n: Tamaño del bloque (por ejemplo, 8 para bloques de 8x8)

1. Dividir I\_original e I\_distorsionada en bloques de tamaño n x n.
2. Para cada bloque (i, j) en I\_original e I\_distorsionada:
  - a. Obtener el bloque original B\_original y el bloque distorsionado B\_distorsionado.
  - b. Aplicar SVD a ambos bloques:  
$$B_{\text{original}} = U_{\text{original}} * \Sigma_{\text{original}} * V_{\text{original}}^T$$
  
$$B_{\text{distorsionado}} = U_{\text{distorsionado}} * \Sigma_{\text{distorsionada}} * V_{\text{distorsionado}}^T$$
  - c. Extraer los valores singulares de cada bloque:  
 $s = \text{valores singulares de } \Sigma_{\text{original}}$   
 $s_{\text{roof}} = \text{valores singulares de } \Sigma_{\text{distorsionada}}$
  - d. Calcular la medida gráfica  $\sqrt{\sum (s_i - s_{\text{roof}_i})^2}$  del bloque usando la fórmula:  
$$D_{\{i, j\}} = \sqrt{\sum (s_i - s_{\text{roof}_i})^2} \quad \# \text{Para } i = 1 \text{ hasta } n \text{ (tamaño del bloque)}$$
  - e. Almacenar  $\sqrt{\sum (s_i - s_{\text{roof}_i})^2}$  como la medida de distorsión para el bloque (i, j).
3. Generar el mapa de distorsión utilizando los valores  $\sqrt{\sum (s_i - s_{\text{roof}_i})^2}$  para cada bloque.

# Salida:

# D: Mapa de distorsión de la imagen, donde cada entrada  $D_{\{i, j\}}$  representa la medida gráfica de distorsión para el bloque (i, j)

La imagen se divide en bloques, y se aplica SVD a cada bloque tanto en la imagen original como en la distorsionada.

Se calcula la distancia entre los valores singulares de los bloques originales y distorsionados usando la métrica especificada, lo que da como resultado un valor  $D_{i,j}$  que representa la distorsión de cada bloque.

Estos valores se usan para construir un mapa de distorsión, visualizando las áreas afectadas de la imagen.

#### Fase 2: Cálculo de la Medida Numérica de Calidad de Imagen (M-SVD)

# Entrada:

# D: Mapa de distorsión calculado en la Fase 1

# k: Tamaño total de la imagen (por ejemplo, 512 para una imagen de 512x512)

# n: Tamaño del bloque (por ejemplo, 8)

1. Convertir el mapa de distorsión D en una lista de valores  $D_i$  para cada bloque.
2. Ordenar la lista de valores  $D_i$  en orden ascendente.
3. Calcular el valor medio  $D_{mid}$  de la lista ordenada de  $D_i$  (es decir, el valor en la posición central de la lista).
4. Calcular la medida numérica M-SVD utilizando la fórmula:

$$M_{SVD} = (1 / (k / n)^2) * \sum |D_i - D_{mid}| \quad \# \text{ Para } i = 1 \text{ hasta } (k / n)^2$$

# Salida:

#  $M_{SVD}$ : Medida numérica de distorsión para la imagen

La lista de valores  $D_i$  del mapa de distorsión se ordena y se calcula el valor medio  $D_{mid}$ .

La medida numérica M-SVD representa el promedio de las desviaciones absolutas de  $D_i$  con respecto a  $D_{mid}$ , normalizado según el tamaño de la imagen y el bloque.

## **Experimentos Numéricos:**

Imagen	SSIM	HQI
imMod1.jpg	0.8111	0.9172
imMod2.jpg	0.7510	0.9105
imMod3.jpg	0.7542	0.8323
imMod4.jpg	0.7503	0.7832
imMod5.jpg	0.9824	0.6596
imMod6.jpg	0.9553	0.5579
imMod7.jpg	0.7194	0.2127

## ***Artículo Científico 2: A Watermarking Algorithm for Digital Image Based on DCT and SVD de X. Shen y X. Ma.***

### **Aplicación:**

En el segundo artículo científico, la descomposición en valores singulares se utiliza en conjunto con la Transformada Discreta del Coseno (DCT) para la inserción de marcas de agua digitales en imágenes. Esta técnica permite crear una marca de agua robusta y casi imperceptible, lo cual es esencial para la protección de derechos de autor y la autenticidad de contenido multimedia.

La aplicación específica del SVD en este contexto implica modificar los valores singulares de ciertos bloques de la imagen para incrustar la información de la marca de agua. La resistencia de los valores singulares frente a manipulaciones comunes de la imagen, como compresión con pérdida, ruido, y filtrado, permite que la marca de agua permanezca detectable incluso después de estas transformaciones. De este modo, el método asegura la durabilidad de la marca de agua, permitiendo su extracción y verificación sin afectar la calidad visual de la imagen de manera significativa.

### **Formulación Matemática:**

La formulación matemática central del método de marcado de agua en imágenes digitales presentado en este artículo combina la transformada discreta del coseno y la descomposición en valores singulares. Este enfoque tiene como objetivo crear un método robusto y de alta calidad para incrustar marcas de agua en imágenes, permitiendo que esta permanezca legible incluso después de someter la imagen a diversas manipulaciones.

La DCT convierte los valores de píxeles de una imagen del dominio espacial al dominio de la frecuencia. Para una imagen dividida en bloques de 8×8, la DCT en 2D se define como:

$$F(u, v) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(u) \cdot C(v) \cdot f(i, j) \cdot \cos\left[\frac{\pi(2i+1)u}{2N}\right] \cdot \cos\left[\frac{\pi(2j+1)v}{2N}\right]$$

donde  $f(i,j)$  representa la intensidad del píxel en la posición  $(i,j)$  del bloque, y  $N$  es el tamaño del bloque (en este caso,  $N=8$ ). Adicionalmente,  $C(u)$  y  $C(v)$  se definen como:

$$\begin{aligned} C(u), C(v) &= \left\{ \sqrt{1/N}, u, v, = 0 \right\} \\ C(u), C(v) &= \left\{ \sqrt{2/N}, u, v, = 1, 2, \dots, N - 1 \right\} \end{aligned}$$

Además, la DCT inversa se define como:

$$f(u, v) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(u) \cdot C(v) \cdot F(i, j) \cdot \cos\left[\frac{\pi(2i+1)u}{2N}\right] \cdot \cos\left[\frac{\pi(2j+1)v}{2N}\right]$$

La DCT permite concentrar la mayor parte de la información en los coeficientes de frecuencia baja, lo cual es útil para la compresión. En el contexto del marcado de agua, se trabaja con estos coeficientes porque modificaciones en ellos tienden a ser menos perceptibles, preservando la calidad visual de la imagen.

La SVD descompone cualquier matriz  $A$  de tamaño  $m \times n$  en tres matrices:

$$A = U\Sigma V^T$$

donde  $U$  y  $V$  son matrices ortogonales de tamaños  $m \times m$  y  $n \times n$  respectivamente, y  $\Sigma$  es una matriz diagonal  $m \times n$  con los valores singulares de  $A$  en la diagonal, ordenados de mayor a menor. Estos valores singulares representan la "fuerza" o "energía" de las características en la imagen.

Para el marcado de agua, se aprovecha la estabilidad de los valores singulares de una imagen frente a manipulaciones comunes, lo cual permite que la marca de agua permanezca incluso tras la compresión o el filtrado.

#### **Procedimiento de Inserción de la Marca de Agua:**

1. División y DCT: Se divide la imagen original en bloques de 8×8 y se aplica la DCT a cada bloque. Posteriormente, se extraen los coeficientes de frecuencia baja de cada bloque, los cuales se organizan en una nueva matriz  $A$ .
2. Aplicación de SVD a  $A$ : La matriz  $A$ , que representa los coeficientes de frecuencia baja de los bloques de la imagen, se descompone mediante SVD:

$$A = U_A \Sigma_A V_A^T$$

3. Incrustación de la Marca de Agua: Se representa la marca de agua como una matriz  $W$  de tamaño apropiado. Luego, se ajusta la matriz de valores singulares  $\Sigma_A$  para incorporar la marca de agua, obteniendo:

$$\Sigma_W = \Sigma_A + \alpha W$$

donde  $\alpha$  es un parámetro de ganancia que controla la intensidad de la marca de agua en la imagen. A continuación, se calcula una nueva matriz  $A_W$  usando las matrices  $U_A$ ,  $\Sigma_W$  y  $V_A^T$ :

$$A_W = U_A \Sigma_W V_A^T$$

4. Reconstrucción de la Imagen con Marca de Agua: Finalmente, se reemplazan los coeficientes de frecuencia baja originales de cada bloque con los nuevos valores en  $A_W$ , y se aplica la DCT inversa para obtener la imagen marcada.

### **Extracción de la Marca de Agua**

Para extraer la marca de agua de una imagen marcada:

1. Se divide la imagen marcada en bloques, se aplica DCT y SVD a los coeficientes de frecuencia baja, obteniendo la matriz  $\Sigma_{Aw}$ .
2. La marca de agua se recupera calculando:

$$W = \frac{\Sigma_{Aw} - \Sigma_A}{\alpha}$$

Las ventajas del método surgen tomando en cuenta que este enfoque es robusto, ya que los valores singulares son estables frente a manipulaciones como compresión y ruido, y la combinación con la DCT permite incrustar la marca de agua en componentes frecuenciales menos perceptibles, manteniendo la calidad de la imagen.

Este sistema asegura una alta resistencia contra manipulaciones comunes, al tiempo que mantiene la marca de agua casi imperceptible para el observador.

### **Pseudocódigo:**

Pseudocódigo del Algoritmo de Marcado de Agua Basado en DCT y SVD

Fase 1: Inserción de la Marca de Agua

# Entradas:

# I: Imagen original de tamaño 512x512

# W: Marca de agua de tamaño 32x32

#  $\alpha$ : Factor de ganancia para controlar la intensidad de la marca de agua ( $\alpha = 0.1$ )

1. Dividir la imagen original I en bloques de 8x8 píxeles.

2. Para cada bloque de 8x8 en I:
  - a. Aplicar la Transformada Discreta del Coseno (DCT) al bloque.
  - b. Extraer el valor de la componente DC (coeficiente de baja frecuencia) del bloque y almacenarlo en una nueva matriz A de tamaño 64x64.
  
3. Aplicar SVD a la matriz A:  

$$A = U * \Sigma * V^T$$
  
4. Incrustar la marca de agua W en los valores singulares de A:
  - a. Aplicar SVD a W para obtener su descomposición:  $W = UW * \Sigma_W * VV^T$
  - b. Modificar la matriz de valores singulares de A:  $\Sigma_{A\_W} = \Sigma + \alpha * \Sigma_W$
  
5. Reconstruir la matriz marcada A\_W:  

$$A\_W = U * \Sigma_{A\_W} * V^T$$
  
6. Reemplazar los valores DC de cada bloque de 8x8 en I con los valores correspondientes en A\_W.
  
7. Aplicar la DCT inversa a cada bloque de 8x8 para obtener la imagen marcada I\_W de tamaño 512x512.

# Salida:  
# I\_W: Imagen marcada con la marca de agua

Se aplica DCT a bloques de 8x8 para obtener la matriz de baja frecuencia (DC) de la imagen, luego se emplea SVD para descomponer esta matriz y modificar sus valores singulares con la marca de agua escalada por el factor de ganancia  $\alpha$ .

La DCT inversa se usa para reconstruir la imagen marcada sin afectar significativamente la calidad visual.

#### Fase 2: Extracción de la Marca de Agua

# Entrada:  
# I\_W: Imagen marcada de tamaño 512x512  
#  $\alpha$ : Factor de ganancia utilizado en la inserción de la marca de agua ( $\alpha = 0.1$ )

1. Dividir la imagen marcada I\_W en bloques de 8x8 píxeles.
  
2. Para cada bloque de 8x8 en I\_W:

- a. Aplicar la Transformada Discreta del Coseno (DCT) al bloque.
- b. Extraer el valor de la componente DC de cada bloque y almacenarlo en una nueva matriz A\_W de tamaño 64x64.

3. Aplicar SVD a la matriz A\_W:

$$A_W = U * \Sigma A_W * V^T$$

4. Extraer la marca de agua W a partir de los valores singulares de A\_W:

$$\Sigma W_{extraída} = (\Sigma A_W - \Sigma) / \alpha$$

$$W_{extraída} = UW * \Sigma W_{extraída} * VW^T$$

# Salida:

# W\_extraída: Marca de agua extraída

# Entrada:

# I\_W: Imagen marcada de tamaño 512x512

# α: Factor de ganancia utilizado en la inserción de la marca de agua ( $\alpha = 0.1$ )

1. Dividir la imagen marcada I\_W en bloques de 8x8 píxeles.

2. Para cada bloque de 8x8 en I\_W:

- a. Aplicar la Transformada Discreta del Coseno (DCT) al bloque.

- b. Extraer el valor de la componente DC de cada bloque y almacenarlo en una nueva matriz A\_W de tamaño 64x64.

3. Aplicar SVD a la matriz A\_W:

$$A_W = U * \Sigma A_W * V^T$$

4. Extraer la marca de agua W a partir de los valores singulares de A\_W:

$$\Sigma W_{extraída} = (\Sigma A_W - \Sigma) / \alpha$$

$$W_{extraída} = UW * \Sigma W_{extraída} * VW^T$$

# Salida:

# W\_extraída: Marca de agua extraída

Se repite el proceso de DCT y SVD en la imagen marcada. La marca de agua se extrae comparando los valores singulares de la imagen marcada con los originales, escalados por  $\alpha$ .

## Experimentos Numéricos y/o resultados:

Imagen Original



Marca de Agua Extraída

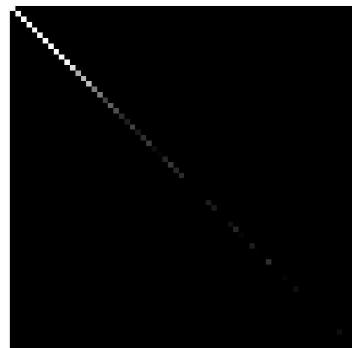


Imagen con Marca de Agua



Diferencia entre Imagen Original y Marcada



La imperceptibilidad de la marca de agua en la imagen marcada se debe a la utilización de técnicas de incrustación en el dominio de la frecuencia (DCT) y modificación de valores singulares (SVD), las cuales permiten introducir cambios sutiles en los componentes de baja frecuencia, que son menos perceptibles para el ojo humano. Aunque la diferencia entre la imagen original y la marcada revela la presencia de la marca de agua, estos cambios son tan pequeños que quedan por debajo del umbral de percepción humana, logrando así un equilibrio entre transparencia y robustez. De este modo, la imagen marcada mantiene su apariencia visual idéntica a la original, mientras que la marca de agua sigue siendo detectable y extraíble a través de técnicas específicas de análisis.

## **Artículo Científico 3: Facial Recognition with Singular Value Decomposition de G. Zeng.**

### **Aplicación:**

En el tercer artículo, la descomposición en valores singulares se utiliza para el reconocimiento facial en tiempo real mediante la construcción de un espacio sub-dimensional conocido como "espacio de caras" o "face space". Este espacio se crea a partir de un conjunto de imágenes faciales conocidas y sirve como una representación de las variaciones en los rostros.

La SVD permite reducir la dimensionalidad de las imágenes faciales, proyectándolas en este "face space". Luego, cualquier nueva imagen de rostro puede proyectarse en el mismo espacio y compararse con las proyecciones de las imágenes conocidas. De este modo, en lugar de comparar píxel por píxel, el sistema evalúa las coordenadas en el espacio de caras, lo que reduce significativamente el esfuerzo computacional y permite clasificar los rostros de manera rápida. Esta técnica resulta particularmente eficaz debido a que las características faciales se pueden representar de forma compacta mediante las singularidades derivadas de la SVD, lo cual proporciona una base robusta para la identificación y clasificación de personas.

### **Formulación Matemática:**

La formulación matemática central en el artículo de reconocimiento facial se basa en la construcción de un espacio de características faciales ("face space") mediante descomposición en valores singulares (SVD), lo que permite reducir la dimensionalidad y facilitar la comparación de rostros en un espacio sub-dimensional.

**Representación de Imágenes Faciales en el Espacio de Imágenes:**  
Para representar una imagen facial, cada imagen con  $m \times n$  píxeles se convierte en un vector columna de dimensión  $M = m \cdot n$ . Dado un conjunto de entrenamiento con  $N$  imágenes de rostros conocidos, formamos una matriz  $S$  de tamaño  $M \times N$ , donde cada columna representa un rostro:

$$S = [f_1, f_2, \dots, f_N]$$

### **Cálculo de la Imagen Promedio:**

Para eliminar las características comunes y concentrarse en las variaciones individuales, se calcula la imagen promedio  $\bar{f}$  del conjunto de entrenamiento:

$$\bar{f} = \frac{1}{N} \sum_{i=1}^N f_i$$

Luego, cada rostro en el conjunto se centra restándole la imagen promedio:

$$a_i = f_i - \bar{f}$$

Con esto, se forma una nueva matriz A, de tamaño M×N, que contiene las variaciones individuales de cada imagen respecto a la media:

$$A = [a_1, a_2, \dots, a_N]$$

Descomposición en Valores Singulares (SVD):

La SVD se aplica a la matriz A, lo que permite expresar A en términos de tres matrices:

$$A = U\Sigma V^T$$

donde:

- U es una matriz ortogonal de M×M cuyas columnas son los vectores singulares izquierdos de A.
- Σ es una matriz diagonal M×N que contiene los valores singulares de A, ordenados de mayor a menor.
- V es una matriz ortogonal de N×N cuyas columnas son los vectores singulares derechos de A.

Construcción del Espacio de Caras:

La descomposición en valores singulares permite identificar un subespacio de menor dimensionalidad (el "face space") que captura las características esenciales de las caras. La dimensión del espacio de caras, representado por el rango r de A, es típicamente mucho menor que M (la dimensión original de las imágenes), ya que los primeros r valores singulares suelen capturar la mayor parte de la variación en los rostros.

Los primeros r vectores singulares izquierdos de U se convierten en las "caras base" o "base-faces". Cada rostro en el conjunto de entrenamiento puede representarse ahora como una combinación lineal de estas caras base, lo que reduce la cantidad de datos necesarios para describir cada rostro.

Proyección de Nuevas Imágenes y Clasificación:

Para clasificar una nueva imagen facial f, se proyecta en el espacio de caras:

1. Primero, se centra la imagen restando la imagen promedio  $\bar{f}$ :

$$\hat{f} = f - \bar{f}$$

2. Luego, se calcula el vector de coordenadas x, que contiene los pesos de la combinación lineal de las caras base necesarias para aproximar f:

$$x = [u_1, u_2, \dots, u_r]^T \hat{f}$$

Este vector x se considera la representación de f en el espacio de caras.

## Comparación y Clasificación:

La clasificación de  $f$  se realiza comparando su vector de coordenadas  $x$  con los vectores de coordenadas  $x_i$  de los rostros conocidos en el conjunto de entrenamiento.

Se calcula la distancia euclídea entre  $x$  y cada  $x_i$ :

$$\epsilon_i = \|x - x_i\|$$

Si el valor mínimo de  $\epsilon_i$  es menor que un umbral predefinido  $\epsilon_0$ , la imagen se clasifica como el rostro correspondiente al mínimo  $\epsilon_i$ . Si ninguna distancia es suficientemente pequeña, la imagen se clasifica como "rostro desconocido".

## Identificación de No-Caras:

Si una imagen no es una cara, su proyección en el espacio de caras generará una distancia elevada con respecto al subespacio. La distancia de  $\hat{f}$  a su proyección en el espacio de caras  $p_f$  se utiliza para determinar si una imagen corresponde a una cara:

$$\epsilon_f = \|\hat{f} - p_f\|$$

Si  $\epsilon_f$  es mayor que un umbral  $\epsilon_1$ , entonces  $f$  se clasifica como "no rostro".

## Ventajas del Método Basado en SVD

El método SVD en reconocimiento facial tiene varias ventajas:

- Reducción de Dimensionalidad: La SVD reduce significativamente la dimensionalidad, lo cual permite realizar comparaciones rápidas y eficientes.
- Generalización: Las "caras base" generadas no dependen de características específicas como ojos o nariz, sino de variaciones generales en los datos de los rostros.
- Robustez: La SVD es menos sensible al ruido y a variaciones menores en las imágenes faciales, lo cual mejora la precisión en condiciones variadas.

Este método permite un reconocimiento facial eficiente y en tiempo real al reducir la representación de cada rostro a un conjunto reducido de componentes esenciales, logrando una clasificación rápida y efectiva.

## Pseudocódigo:

Pseudocódigo del Algoritmo de Reconocimiento Facial Basado en SVD

Fase 1: Construcción del Espacio de Caras

# Entradas:

# S: conjunto de imágenes de entrenamiento, donde cada imagen tiene m x n píxeles

# N: número de imágenes en el conjunto de entrenamiento

1. Calcular la imagen promedio:

$$f_{\text{promedio}} = (1 / N) * \text{SUM}(f_i \text{ for } i \text{ in } 1 \text{ to } N)$$

2. Centrar las imágenes de entrenamiento:

Para cada imagen  $f_i$  en  $S$ :

$$a_i = f_i - f_{\text{promedio}} \# a_i \text{ es la versión centrada de } f_i$$

3. Formar la matriz A:

$$A = [a_1, a_2, \dots, a_N] \# \text{Matriz A de tamaño } M \times N, \text{ donde } M = m * n$$

4. Aplicar la descomposición en valores singulares (SVD) a A:

$$A = U * \Sigma * V^T$$

5. Construir el espacio de caras:

Seleccionar los primeros  $r$  vectores singulares izquierdos de  $U$ ,  $\{u_1, u_2, \dots, u_r\}$ , donde  $r$  es el rango de  $A$ .

Cada  $u_j$  representa una "cara base" en el espacio de caras.

6. Calcular las coordenadas de cada imagen de entrenamiento en el espacio de caras:

Para cada imagen  $f_i$  en  $S$ :

$$x_i = [u_1, u_2, \dots, u_r]^T * (f_i - f_{\text{promedio}})$$

Guardar  $x_i$  como la representación de  $f_i$  en el espacio de caras

Esta fase convierte cada imagen en el conjunto de entrenamiento en un vector de características en un espacio de menor dimensionalidad. La SVD permite identificar las características clave que definen las variaciones entre los rostros.

Fase 2: Clasificación de una Nueva Imagen

# Entrada:

#  $f$ : nueva imagen a clasificar (con  $m \times n$  píxeles)

#  $f_{\text{promedio}}$ : imagen promedio calculada en la fase de entrenamiento

#  $\{u_1, u_2, \dots, u_r\}$ : vectores de base en el espacio de caras

#  $\{x_i\}$ : coordenadas de cada imagen conocida en el espacio de caras

#  $\epsilon_0$ : umbral de clasificación para rostros conocidos

#  $\epsilon_1$ : umbral de clasificación para diferenciar rostros y no rostros

1. Centrar la nueva imagen:

$$f_{\text{centrada}} = f - f_{\text{promedio}}$$

2. Proyectar la nueva imagen en el espacio de caras:

```
x = [u_1, u_2, ..., u_r]^T * f_centralizada  
# x representa las coordenadas de f en el espacio de caras
```

3. Calcular la distancia al espacio de caras:

```
p_f = SUM(x_j * u_j for j in 1 to r) # Proyección de f en el espacio de caras  
epsilon_f = ||f_centralizada - p_f|| # Distancia de f al espacio de caras
```

4. Determinar si f es una cara:

Si  $\epsilon_f > \epsilon_1$ :  
Clasificar f como "no rostro" y TERMINAR

5. Comparar f con rostros conocidos:

Para cada coordenada  $x_i$  en el conjunto de entrenamiento:  
 $\epsilon_i = ||x - x_i||$  # Distancia entre f y cada rostro conocido

Si el mínimo de  $\epsilon_i$  es menor que  $\epsilon_0$ :

Clasificar f como el rostro conocido más cercano (asociado al mínimo  $\epsilon_i$ )

De lo contrario:

Clasificar f como "rostro desconocido"

Para una nueva imagen, el algoritmo proyecta la imagen en el espacio de caras y calcula su distancia a este espacio. Si está dentro de un umbral de proximidad, se considera una cara. Luego, se compara con las proyecciones de rostros conocidos para encontrar el rostro más cercano o clasificarlo como "rostro desconocido".

### Experimentos Numéricos:

La implementación en Octave selecciona cada una de las imágenes del archivo compare, e identifica cuál es la imagen que identifica de la base de datos training. El resultado se muestra en la consola a continuación.

Command Window

```
Imagen 1 en compare se identifica como Persona 1 en training
Imagen 2 en compare se identifica como Persona 2 en training
Imagen 3 en compare se identifica como Persona 3 en training
Imagen 4 en compare se identifica como Persona 4 en training
Imagen 5 en compare se identifica como Persona 40 en training
Imagen 6 en compare se identifica como Persona 6 en training
Imagen 7 en compare se identifica como Persona 7 en training
Imagen 8 en compare se identifica como Persona 8 en training
Imagen 9 en compare se identifica como Persona 9 en training
Imagen 10 en compare se identifica como Persona 38 en training
Imagen 11 en compare se identifica como Persona 11 en training
Imagen 12 en compare se identifica como Persona 12 en training
Imagen 13 en compare se identifica como Persona 13 en training
Imagen 14 en compare se identifica como Persona 14 en training
Imagen 15 en compare se identifica como Persona 15 en training
Imagen 16 en compare se identifica como Persona 16 en training
Imagen 17 en compare se identifica como Persona 17 en training
Imagen 18 en compare se identifica como Persona 18 en training
Imagen 19 en compare se identifica como Persona 19 en training
Imagen 20 en compare se identifica como Persona 20 en training
Imagen 21 en compare se identifica como Persona 21 en training
Imagen 22 en compare se identifica como Persona 22 en training
Imagen 23 en compare se identifica como Persona 23 en training
Imagen 24 en compare se identifica como Persona 24 en training
Imagen 25 en compare se identifica como Persona 25 en training
Imagen 26 en compare se identifica como Persona 26 en training
Imagen 27 en compare se identifica como Persona 27 en training
Imagen 28 en compare se identifica como Persona 28 en training
Imagen 29 en compare se identifica como Persona 29 en training
Imagen 30 en compare se identifica como Persona 30 en training
Imagen 31 en compare se identifica como Persona 31 en training
Imagen 32 en compare se identifica como Persona 32 en training
Imagen 33 en compare se identifica como Persona 33 en training
Imagen 34 en compare se identifica como Persona 34 en training
Imagen 35 en compare se identifica como Persona 35 en training
Imagen 36 en compare se identifica como Persona 36 en training
Imagen 37 en compare se identifica como Persona 37 en training
Imagen 38 en compare se identifica como Persona 38 en training
```

< >

Command Window Editor

```
Imagen 39 en compare se identifica como Persona 39 en training
Imagen 40 en compare se identifica como Persona 40 en training
```

< >

Command Window Editor Profiler

## Bibliografía:

- GeeksforGeeks (2024) *Singular Value Decomposition (SVD)* Obtenido de: <https://www.geeksforgeeks.org/singular-value-decomposition-svd/>
- Liu F, Liu Y. (2008) *A Watermarking Algorithm for Digital Image Based on DCT and SVD.*
- s.a. (2018) *Singular Value Decomposition as Simply as Possible*. Obtenido de: <https://gregorygundersen.com/blog/2018/12/10/svd/>
- ScienceDirect (2014). *Watermarking Algorithm*. Obtenido de <https://www.sciencedirect.com/topics/computer-science/watermarking-algorithm#:~:text=A%20watermarking%20algorithm%20is%20a,and%20robustness%20of%20the%20watermark>.
- Shnayderman A., Gusev A., Eskicioglu A. (2006) *An SVD-Based Grayscale Image Quality Measure for Local and Global Assessment.*
- Strang G. *The Discrete Cosine Transform*. Obtenido de: <https://epubs.siam.org/doi/pdf/10.1137/S0036144598336745>
- Zeng, G. (2007) *Facial Recognition with Singular Value Decomposition.*