

# Reconocimiento facial usando SVD: Aplicación al Registro Nacional de Identificación y Estado Civil (RENIEC)

Carlos Mariños, Deysi Zavaleta, Paula Ramos

Departamento de Ingeniería, Universidad del Pacífico

Lima, Perú

cf.marinoso@alum.up.edu.pe

dy.zavaletar@alum.up.edu.pe

pf.ramoss@alum.up.edu.pe

## I. INTRODUCCIÓN

En abril de 2025, Perú enfrentó un escándalo político de gran magnitud debido a la detección de más de 300,000 firmas falsas utilizadas por al menos 32 partidos políticos para su inscripción ante el Jurado Nacional de Elecciones (JNE) de las cuales todas debieron pasar por la verificación del Registro Nacional de Identificación y Estado Civil (RENIEC). Entre las agrupaciones implicadas se encuentran Fuerza Popular, Avanza País, Perú Primero y Primero la Gente. Estas irregularidades incluyen la suplantación de identidad y afiliaciones sin consentimiento de miles de ciudadanos, lo que generó una crisis de desconfianza en el sistema electoral peruano según Perú, R. I. [1]

Hoy en día, la validación de firmas para inscribir partidos políticos se hace principalmente revisando la firma manuscrita y la huella digital del ciudadano. Pero estas técnicas no están exentas de vulnerabilidades que podrían ser aprovechadas para hacer fraudes o suplantaciones. Para combatir eso, se propone crear un sistema de reconocimiento facial que compare automáticamente la foto en el Documento Nacional de Identidad (DNI) con una toma en tiempo real. La idea es hacer que el proceso de inscripción sea más seguro y confiable [2][3].

Para lograr esto, se planea usar técnicas de procesamiento de imágenes, como Singular Value Decomposition (SVD) y Análisis de Componentes Principales (PCA). Ambas sirven para reducir las imágenes a un espacio de menor tamaño, guardando las características más importantes para facilitar la comparación. En este proyecto, se evaluarán y compararon ambas metodologías: usar SVD directamente sobre imágenes individuales, y aplicar PCA usando SVD sobre una base de datos con muchas fotos. Esto ayudará a decidir cuál método funciona

mejor, con más precisión y rapidez, en condiciones reales. La idea es ofrecer una solución tecnológica confiable que ayude a prevenir la suplantación en procesos electorales.

## II. DESARROLLO DE CONCEPTOS

### 1. Verificación biométrica

La RENIEC describe la verificación biométrica como el uso de características físicas únicas, como las huellas o el rostro, para confirmar quién eres. En estas elecciones, se usará el reconocimiento facial como una capa adicional de seguridad, ayudando a prevenir fraudes y suplantaciones, ya que convierte las imágenes en datos matemáticos que comparan rasgos faciales con una base de datos o una foto de referencia, como el DNI. [3]

### 2. Reconocimiento facial

El reconocimiento facial automatizado funciona comparando las características de una imagen con los datos existentes, utilizando técnicas como PCA (Análisis de Componentes Principales) o SVD (Descomposición en Valores Singulares). Estas técnicas permiten extraer los rasgos más importantes para hacer comparaciones rápidas y precisas. [2]

### 3. Singular Value Decomposition (SVD)

Gonzalez y Woods detallan que la descomposición en valores singulares (SVD) es una técnica matemática que descompone una matriz (como una imagen digital) en tres componentes: U,  $\Sigma$  y  $V^t$ . El componente  $\Sigma$  contiene los valores singulares de la información. Así, al proyectar 2 imágenes en el espacio definido por las columnas de  $V$  (espacio de eigenfaces), es posible comparar las proyecciones para estimar qué tan

similares son las imágenes entre sí. [4]

#### 4. Principal Component Analysis (PCA)

El Análisis de Componentes Principales, conocido como PCA, es una técnica que ayuda a simplificar datos complejos como las imágenes, reduciéndose a un espacio de menor dimensión más sencillo y fácil de manejar. Lo que hace PCA es transformar las imágenes originales en un conjunto de nuevas variables que no están correlacionadas entre sí, llamadas componentes principales. Estos componentes capturan las direcciones de máxima variación de los datos originales. Es importante mencionar que PCA se puede calcular mediante la SVD, que permite encontrar las direcciones principales de cambio en las imágenes, conocidas como autovectores, sin necesidad de crear de manera explícita la matriz de covarianza.

#### 5. Cálculo del error

Una vez obtenidos los valores singulares de las imágenes, según Gonzalez y Woods se calcula un error de aproximación, después de extraer los vectores de características, ya sea usando los valores singulares en SVD o los componentes principales en PCA, se procede a calcular una medida de similitud entre las imágenes comparadas. Para SVD, se puede optar por la distancia euclídea o por el error cuadrático medio (ECM) entre los vectores truncados de los valores singulares. En el caso de PCA, una métrica que se utiliza habitualmente es la similitud de coseno, que mide el ángulo entre los vectores proyectados. En ambos métodos, un valor bajo de error o una alta similitud indican que las imágenes probablemente corresponden a la misma persona. [4]

#### 6. Umbral de decisión

Gonzalez y Woods recomiendan establecer un umbral para decidir si dos imágenes son lo suficientemente similares. Si la similitud (por ejemplo, similitud coseno en PCA o diferencia mínima en valores singulares en SVD) supera este umbral, se acepta la verificación como válida; en caso contrario, se rechaza. Este umbral puede ser calibrado empíricamente para balancear entre falsos positivos y falsos negativos. [4]

#### 7. Aplicaciones en procesos electorales

La implementación de sistemas de reconocimiento

facial basados en PCA o SVD en el proceso de inscripción de partidos políticos puede detectar suplantaciones de identidad y reducir el uso de firmas falsas. Según el JNE, al incorporar múltiples métodos biométricos (firma, huella digital y rostro), se fortalece la fiabilidad del sistema electoral, promoviendo así la transparencia democrática. La integración de herramientas matemáticas de reconocimiento facial podría, por tanto, restaurar la confianza ciudadana y mejorar la eficiencia de la verificación de identidad. [2]

### III. ESTADO DEL ARTE

En la literatura, se encuentran diversos trabajos que abordan la técnica biométrica de reconocimiento facial; sin embargo, la diferencia entre estas investigaciones radica en la técnica que utilizan para llevar a cabo este análisis dependiendo de los requisitos que requiera cada proyecto, ya sea la prioridad reducir la dimensionalidad de la imagen, reconocer edad, poses o analizar la imagen a pesar de las variaciones de luz y sombras. Este tema ha sido utilizado para aplicaciones en su mayoría de control de accesos, seguridad o vigilancia. Es importante recalcar, de acuerdo con Crosswhite et al. que existen dos enfoques: la **verificación** de rostro (uno a uno) que implica hacer una comparación de una imagen de tipo consulta con una imagen ya almacenada en la base de datos para verificar si hay similitud entre ambas. Mientras que, la **identificación** (uno a muchos) se refiere a la comparación de un rostro consulta contra numerosos rostros almacenados en una base de datos, con el fin de asociar este nuevo rostro a alguno de los que ya se tienen guardados [5].

En el caso de la RENIEC, donde se debería validar si el propietario del DNI corresponde con la persona presente físicamente, se requiere un sistema de verificación facial uno a uno, el cual sea preciso, interpretable (para el público) y eficiente.

En cuanto a las técnicas utilizadas para el reconocimiento facial, se tiene como base al álgebra lineal. En primer lugar, Turk y Pentland de la reconocida *Massachusetts Institute of Technology* (MIT) introdujeron el método de *Eigenfaces*, aplicando el Análisis de Componentes Principales (PCA) para el reconocimiento facial. Su estudio consistió en primero convertir cada imagen de los rostros en un vector de alta dimensión. Luego, con ese conjunto de entrenamiento, se calculó la imagen

promedio, de ahí, se le restó este promedio a las imágenes y se construyó una matriz de covarianza con las imágenes resultantes (imágenes centradas, con media 0) ello con el fin de evitar sesgos de la iluminación u otras características y garantizar que el primer componente capture la máxima variabilidad. A continuación, se calcularon los vectores propios (*eigenvectors*) y valores propios (*eigenvalues*) de la matriz de covarianza. Finalmente, se calcula la proyección de las imágenes en el espacio de *eigenfaces*, reduciendo así la dimensionalidad y permitiendo una comparación más eficiente. Como resultados se obtuvo que el método fue efectivo; no obstante, su rendimiento disminuye significativamente ante variaciones de iluminación y expresión facial. A pesar de ello, la investigación sentó las bases para trabajos posteriores en el área [6].

Más de una década después, el método de *eigenfaces* se logró reproducir y optimizar. Prueba de ello, es el artículo de Hsu y Chen, el cual propone realizar reconocimiento facial usando Descomposición en Valores Singulares (SVD), técnica que se puede entender como un 2DPCA, que busca mejorar la eficiencia y precisión en comparación a estudios anteriores. A diferencia del estudio anterior, cuando se usa PCA se convierten las imágenes a vectores unidimensionales (cada imagen de tamaño  $m \times n$  se convierte a un vector de dimensión  $mn$ ), pero en este caso , se mantuvo la estructura bidimensional de las imágenes (ej. 100 x 100 píxeles), preservando mejor la información. Luego, se aplicó directamente la técnica de SVD a cada imagen A, de modo que:

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

Teniendo:

- U: vectores singulares "verticales" (filas)
- $\Sigma$ : valores singulares
- $V^T$  : vectores singulares "horizontales" (columnas)

Obteniendo matrices de características que capturan la variación significativa de las imágenes. Finalmente, se realiza la proyección al espacio generado por las matrices características para luego comparar usando métricas de similitud para el reconocimiento. Como resultado se tuvo una tasa de reconocimiento del 97.5% utilizando 12 vectores singulares izquierdos y 12 derechos representando cada rostro. Este método demostró optimizar el almacenamiento, procesamiento

y la precisión a comparación de PCA. Mejorando también cuando hay condiciones de variabilidad en iluminación y expresión facial, tal como las variaciones que se muestran en la Figura 1[7].

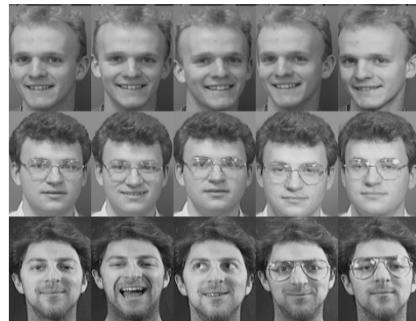


Figura 1: Ejemplo de Imágenes de 3 sujetos con diferentes expresiones faciales de la base de datos ORL utilizada en el estudio de Hsu y Chen [7]

Otro autor que utiliza SVD para el reconocimiento facial es Zeng, quien en su investigación desarrolló un sistema que tuviese la habilidad de aprender a reconocer nuevas caras en tiempo real y de manera automática. Zeng creó un set de *training* con caras conocidas de  $m \times n$  píxeles formando así un subespacio llamado "face space" dentro de todo el espacio de las imágenes. Este subespacio contenía a la variación de las caras conocidas. Su base se definió con la matriz de vectores singulares del set de caras conocidas, el cual no necesariamente contiene detalles como orejas, ojos o nariz. Luego, para identificar una nueva imagen, se realizaba la proyección de esta en el subespacio "face space", para compararlo con las proyecciones de las imágenes ya existentes utilizando distancia euclídea. Ello se realizó debido a que es mucho más fácil y rápido hacer la comparación solo con el subespacio de caras y no con toda la base de datos de imágenes que tiene una dimensión considerablemente más grande al comparar pixel por pixel. Finalmente, con imágenes de *test* (con variaciones de iluminación, expresión y orientación) se evaluó el rendimiento del sistema. Así, se obtuvo que el sistema demostró ser efectivo en el reconocimiento de rostros pero bajo condiciones controladas y variaciones moderadas de las imágenes. El uso de SVD permitió reducir significativamente la dimensionalidad, mejorando la eficiencia computacional sin comprometer la precisión de reconocimiento [8].

Como se puede observar, los métodos clásicos de reconocimiento facial como PCA y SVD son sólidos para el estudio de verificación de identidad, se sabe

que estos presentan limitaciones frente a variaciones grandes de iluminación, pose o envejecimiento. Según Hernández, durante la última década, se han propuesto métodos de aprendizaje profundo como Redes Neuronales Convolucionales (CNN) especialista en condiciones no controladas y para aprender condiciones jerárquicas, o modelos de Análisis Discriminante (LDA) - Fisherfaces, el cual es más robusto frente a variaciones de iluminación y expresión. Estos métodos superan en precisión a los métodos clásicos [9]. Sin embargo, dichos modelos requieren grandes volúmenes de datos, alta capacidad técnica de computación y suelen ser menos intuitivos de interpretar. Se elegirá el método de SVD para el presente estudio debido a su claro procedimiento matemático, eficiencia y claridad computacional, bajo requerimiento de almacenamiento y mayor interpretabilidad.

#### IV. METODOLOGÍA PROPUESTA

El presente trabajo busca implementar un programa en Python que pueda verificar la identidad de un votante al comparar la imagen de su rostro en RENIEC con una foto capturada en tiempo real al inscribirse como ciudadano que apoya al partido. Esto, con el fin de brindar un nivel mayor de seguridad al proceso inscripción de partidos de forma eficiente y reducir la posibilidad de suplantación de identidad. A continuación, se presentan los pasos que se seguirán de acuerdo a lo investigado previamente:

##### Paso 1: Preprocesamiento de imágenes

Se leen las 60 imágenes del conjunto de entrenamiento, convirtiéndolas a escala de grises y redimensionándolas a matrices de  $100 \times 100$  píxeles. Luego, cada matriz es transformada en un vector de 10,000 elementos mediante un proceso de aplanamiento. Los 60 vectores se agrupan en la matriz  $X$ , que tiene una dimensión de 60 columnas y 10,000 filas, donde cada columna representa una imagen aplanada.

$$X \in R^{60 \times 1000}$$

Finalmente, se centran los datos restando con la media para utilizar  $X_{\text{centrada}}$  en el proceso de SVD

$$X_{\text{mean}} = \frac{1}{m} \sum_{i=1}^m X_i$$

$$X_{\text{centrada}} = X - X_{\text{mean}}$$

##### Paso 2: Aplicación de SVD

Se realiza la descomposición SVD utilizando `np.linalg.svd()` sobre la matriz de datos centrada, obteniendo:

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

donde  $U \in \mathbb{R}^{60 \times 60}$ ,  $\Sigma \in \mathbb{R}^{60 \times 60}$  y  $V \in \mathbb{R}^{60 \times 10000}$ . La matriz  $\Sigma$  contiene los valores singulares ordenados de forma descendente, representando varianza de la imagen. Mientras que  $U$  y  $V$  son matrices ortogonales, con  $U$  conteniendo los valores singulares izquierdos de  $A$  y  $V$  conteniendo los valores singulares derechos de  $A$ .

Siendo los valores singulares las raíces cuadradas de los autovalores, representando la magnitud de la transformación lineal. Los valores singulares más grandes capturan las características más significativas del rostro, mientras que los valores menores corresponden al ruido o detalles menos relevantes.

##### Paso 3: Reducción de dimensionalidad

Con el fin de conservar sólo la información más relevante, se seleccionan los primeros  $k$  valores singulares de  $\Sigma$ . Reduciendo así la cantidad de datos a procesar.

$$X_{\text{aprox}} = U_k \Sigma_k V_k^T$$

##### Paso 4: Proyección en el subespacio de características

Para cada imagen, se proyecta su vector aplanado en el subespacio de características definido por los primeros  $k$  vectores de  $V$ . Estos serían los Eigenfaces con más varianza explicada

$$\text{Firma} = X[i] \cdot V_k^t$$

El resultado es un vector de  $k$  elementos que representa la codificación compacta de la imagen

##### Paso 5: Calcular similitud

Para poder evaluar si dos imágenes son de la misma persona, se compararán los dos vectores ( $x_1, x_2$ ) utilizando la distancia euclídea entre sus

proyecciones.

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^k (x_{1i} - x_{2i})^2}$$

Cuando  $d$  sea menor al umbral definido (3500) se considera que las imágenes pertenecen a la misma persona.

#### Paso 6: Validación del sistema

Se evalúa el sistema utilizando un conjunto de imágenes nuevo Xtest compuesto por 5 imágenes de la persona utilizada para el entrenamiento así como otras 5 imágenes diferentes a la persona.

Se calculan métricas como la exactitud (Accuracy), se genera una matriz de confusión, se mide el tiempo de ejecución y el consumo de memoria.

#### Funcionamiento de Numpy (np.linalg.svd)

Esta función calcula directamente los valores de las matrices  $U$ ,  $\Sigma$ ,  $V$ , que contienen lo mencionado previamente. Internamente devuelve la transpuesta de  $V$ . Para ello, se le tienen que ingresar una matriz que ya esté pre procesada, es decir que esté centrada o normalizada según sea el caso. En el preprocessamiento que haremos, centraremos la matriz antes de ingresar a la función.

#### V. PROCEDIMIENTO Y RESULTADOS

En la implementación se realizaron varias pruebas con 2 casos diferentes. En el primero se parte de una sola imagen para generar la data de entrenamiento y en el segundo se opta por una muestra con 60 imágenes diferentes de la misma persona para la data de entrenamiento.

En primer lugar, se instalan las siguientes librerías necesarias para el funcionamiento del código:

- OpenCV: se utiliza para capturar imágenes con la cámara local, detectar rostros humanos y procesar las imágenes capturadas.
- NumPy: es utilizado para la manipulación de matrices y para aplicar SVD.
- Matplotlib: es usado para mostrar las imágenes originales, las imágenes reconstruidas a partir de K componentes de SVD, y el gráfico de disminución de los valores singulares.
- OS: Para guardar imágenes capturadas por el programa como archivos .jpg

#### Prueba preliminar

Luego de instalar las librerías, se utiliza la clase VideoCapture de OpenCV para capturar la imagen del ciudadano en tiempo real. Esto abre una ventana con la cámara del dispositivo, como se muestra en la Figura 2, donde mediante un clasificador de CV2, haarcascade\_frontalface\_default.xml, se reconoce el rostro en la imagen. Esto captura solamente la cara del ciudadano para realizar SVD más adelante.



Figura 2: Captura del rostro del ciudadano usando OpenCV, con detección facial utilizando un clasificador Haar.

Esto se realizó para la prueba piloto donde se utilizó una única imagen de RENIEC para el entrenamiento. En esta prueba piloto se cometió el error de usar como matriz de entrenamiento a la matriz de la imagen como tal. La imagen de reniec era de 100 x 100 pixeles y se descompuso directamente con SVD. De esta forma se determinó que con 15 componentes se capturaba el 90% de la varianza. Luego con Matplotlib, se generan gráficos para observar la imagen original preprocessada redimensionada y en escala de grises, la imagen reconstruida con los 15 valores singulares, y un gráfico que muestra la caída en información aportada por los valores singulares como se muestra en la Figura 3. Este último se utiliza para determinar la validez del número de valores singulares ( $k$ ) escogido por el programa.

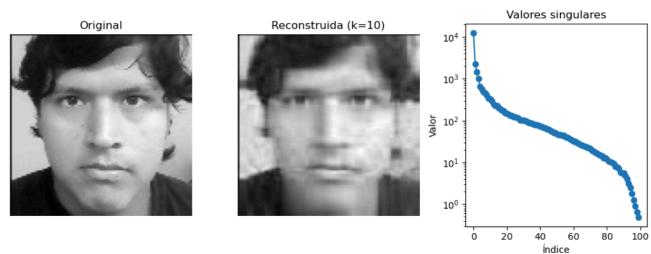


Figura 3: Rostro original, rostro reconstruido con 10 valores singulares y gráfico de caída de valores

singulares.

Este método se descartó debido a que descomponer la imagen directamente no brinda valores singulares interpretables para hacer reconocimiento facial, además que al capturar la imagen en tiempo real se dificulta centrar completamente el rostro de la forma en la que se tenía la imagen de RENIEC. A continuación presentamos los 2 casos principales donde SVD tanto como PCA fueron utilizados y validados.

### 1. Análisis generando Xtrain de una sola imagen

Durante la etapa de desarrollo, se exploró una segunda opción que se basaba en el Análisis de Componentes Principales (PCA), como una alternativa al método directo de descomposición SVD. Es importante destacar que PCA y SVD están muy relacionados, ya que PCA puede realizarse a través de la SVD de una matriz de datos que ha sido centrada. De hecho, muchas librerías en Python, como `sklearn.decomposition.PCA`, usan internamente `numpy.linalg.svd` para calcular los componentes principales de manera eficiente.

Si tenemos una matriz de datos  $X \in \mathbb{R}^{n \times m}$ , donde cada fila es una imagen "aplanada" (vectorizada), el procedimiento clásico de PCA consiste en:

1. **Centrar los datos:** restar la media de cada característica (columna) para que la matriz tenga media cero. Ello, con el fin de que los componentes capturen direcciones de máxima varianza y no solo grandes valores absolutos.
2. **Calcular la SVD de la matriz centrada:**

$$X_{\text{centrada}} = U \Sigma V^T$$

3. **Obtener los componentes principales:** las columnas de  $V$  representan las direcciones de máxima varianza, es decir, los *eigenfaces* o "rostros característicos".

Además, explicándolo aún más matemáticamente, partiendo de una matriz de datos centrada  $X$ , si hacemos su descomposición en SVD se obtiene que los vectores en  $V$  corresponden a los autovectores de la matriz de covarianza  $\frac{1}{n-1} X^T X$ , es decir, son precisamente los componentes principales que se buscan en PCA. Además, los valores singulares en  $\Sigma$  están directamente relacionados con la varianza explicada por cada componente, es decir con los

autovalores de la matriz de covarianza. Entonces, SVD ayuda con la obtención de direcciones principales y también con la magnitud de varianza explicada. Por lo tanto, SVD es un fundamento algebraico de PCA.

En este trabajo, utilizamos PCA para comparar la foto del DNI con una foto reciente del ciudadano. Hay que destacar que en esta prueba inicial, al considerar que entidades como Reniec solo mantienen una única foto correspondiente a la persona, se tuvo que generar artificialmente más fotos para así formar una matriz  $X$  útil para el entrenamiento. Esta matriz contiene 30 imágenes que provenían de la original pero con alteraciones en el brillo y exposición. No obstante, luego de implementar la metodología inicial, se obtuvo que los resultados no eran los esperados, pues al comparar únicamente una imagen del DNI contra una fotografía reciente de la persona, el sistema tenía dificultades para identificar correctamente los rostros, dando constantemente falsos negativos. Factores como cambios en la iluminación, variaciones de la pose o expresión facial, afectaron al rendimiento del sistema. Además, se encontró que la varianza explicada se mantenía en un solo componente como se muestra en la Figura 4:

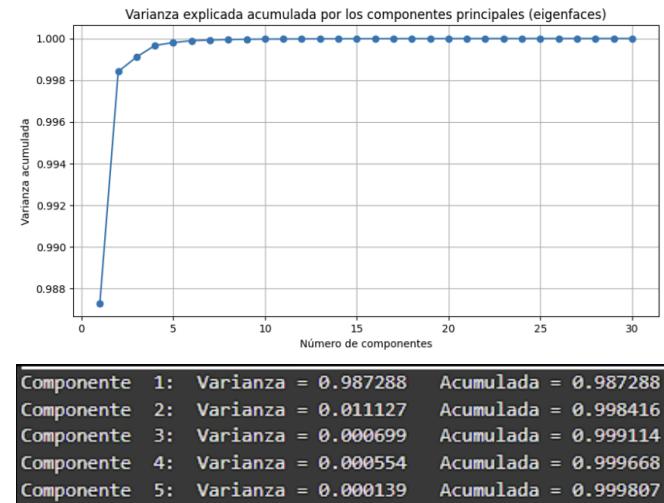


Figura 4: Gráfico de variabilidad explicada acumulada de los componentes principales o eigenfaces junto con la varianza por cada componente.

Esto se puede interpretar como que el primer componente está capturando casi toda la variación de los datos. Es posible intuir que este componente recoge las diferencias totales en iluminación o contraste entre rostros ya que no existe otra diferencia significativa entre las imágenes de entrenamiento para este caso en específico. Se intentó ampliar la

variabilidad de la imagen de entrada, cambiando la iluminación para que se puedan obtener imágenes ‘diferentes’ no obstante, ello no mejoró el resultado, pues todas las poses eran iguales.

Por lo tanto, se decidió aplicar el método de SVD y PCA en un conjunto de varias imágenes por persona, lo que aumenta la variedad de datos disponibles. Esto ayuda a la técnica de PCA a identificar patrones más relevantes para reconocer rostros. A diferencia del método original, donde solo comparábamos una imagen del DNI con una captura en tiempo real lo cual tenía poca variación y hacía que PCA y SVD fueran menos eficientes. En este enfoque creamos una base de datos con unas 60 fotos por persona, tomadas en diferentes condiciones de luz, postura y expresión. Primero, preprocesamos cada imagen convirtiéndola a escala de grises, redimensionándola y convirtiéndola en un vector de 10,000 elementos. Luego, armamos una matriz con todas estas imágenes y aplicamos PCA para extraer los componentes más importantes del rostro promedio, así cada rostro nuevo se puede representar en este espacio reducido de datos.

## 2. Análisis usando SVD y PCA realizando comparación de la foto del DNI contra un conjunto de entrenamiento de 60 imágenes:

A partir de esta observación, se propuso una segunda solución al problema: Utilizar un conjunto de aproximadamente 60 imágenes por persona, donde se tengan diferentes posiciones y expresiones faciales, con el objetivo de aumentar la variabilidad del conjunto de datos. De modo que, se pueda entrenar un modelo más robusto y capaz de reconocer patrones faciales hasta cuando hay variaciones.

Antes de aplicar SVD, se centraron las imágenes a utilizar:

$\text{CaraCentrada} = \mathbf{X} - \mathbf{X}_{\text{mean}}$ . En la Figura 5, se puede observar que la imagen centrada tiene líneas mucho más definidas a comparación de la cara no centrada, pues aquí se busca encontrar los factores más relevantes.

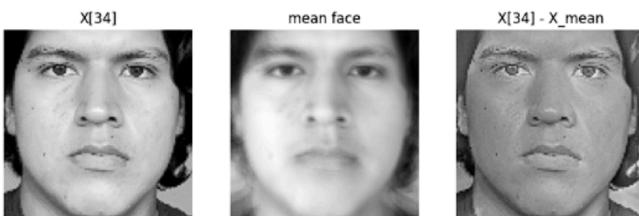


Figura 5: Cara número 34 junto con la cara promedio y

la cara 34 centrada.

Luego, se hallaron los eigenfaces y la variabilidad que representa cada uno de los componentes principales.

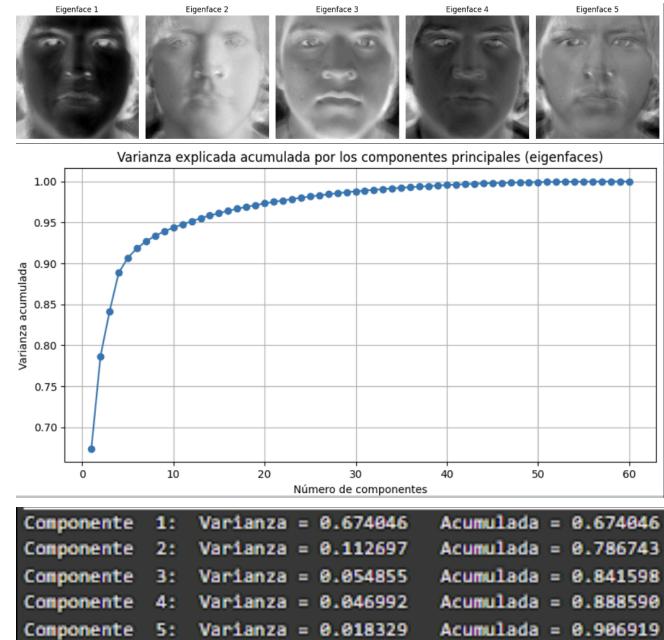


Figura 6: Se muestran los 5 eigenfaces hallados junto con un gráfico de varianza explicada acumulada por cada componente.

En la Figura 6, se puede observar que la varianza explicada no solo se mantiene en el primer componente, sino que se esparce entre el primer, segundo y hasta tercer componente. Conceptualmente, esto indica que se están capturando rasgos característicos reales de la persona. Después de tener las varianzas por componente se realizó la comparación de las imágenes entrenadas contra nuevas imágenes usando distancia euclíadiana con un umbral de 3500.



Figura 7: Cuatro comparaciones de imágenes de entrenamiento contra imágenes nuevas.

Como se observa en la Figura 7, el sistema sí está reconociendo a nuestro sujeto de prueba (hasta cuando hay variaciones en los gestos faciales) y también determinó que es una persona diferente cuando se le comparó con una persona de cabello más largo. No obstante, ocurrió un falso positivo en la última comparación, ello pudo ocurrir debido a que el contorno de la cara de la imagen de la persona nueva es muy similar a la imagen de entrenamiento.

Finalmente, se obtuvo la matriz de confusión del sistema, como se observa en la Figura 8:

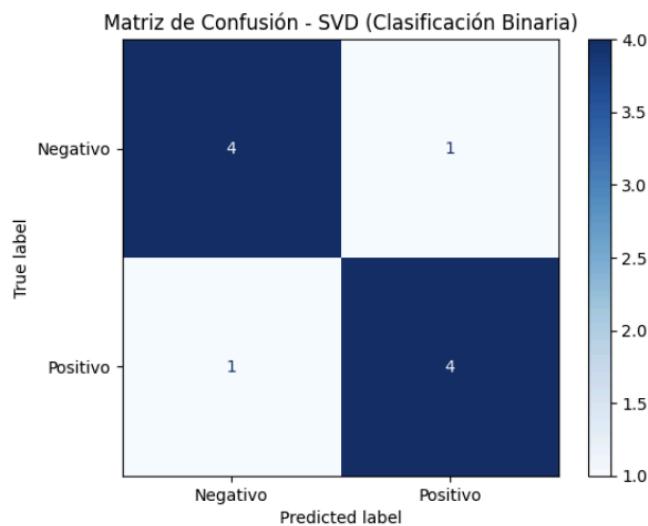


Figura 8: Matriz de confusión para SVD cuando se utilizan 60 imágenes de entrenamiento.

El modelo de clasificación basado en SVD logra un 80% de precisión, lo cual indica un desempeño sólido. La tasa de verdaderos positivos y verdaderos negativos es alta (4 de 5). Solo se comete un error por tipo, por lo que el bajo número de falsos positivos y falsos negativos sugiere que el modelo está bien calibrado y sin sesgos a una clase.

#### Benchmark:

Con el fin de comparar ambas metodologías y también el uso de PCA contra solo SVD, se calcularon indicadores como la exactitud (accuracy) que indica qué tan cerca estuvieron las predicciones del valor real. También se calculó el tiempo total de ejecución, la memoria aproximada utilizada y la varianza explicada con cinco componentes.

```
--- SVD ---
Accuracy: 0.80
Tiempo total: 0.00 s
Memoria total aprox: 1175.17 KB
Varianza explicada (k=5): 0.9069
```

Figura 9: Indicadores de SVD usando 60 imágenes

```
--- PCA ---
Accuracy: 0.80
Tiempo total: 0.08 s
Memoria total aprox: 48466.31 KB
Varianza explicada (k=5): 0.9069
```

Figura 10: Indicadores de PCA usando 60 imágenes

Como se puede observar en las Figuras 9 y 10, el accuracy de ambas es el mismo pues teóricamente el uso de PCA y SVD es indiferente ya que PCA usa SVD para su ejecución. Sin embargo, el cambio se visualiza en el tiempo y la memoria utilizada, donde se concluye que PCA puede llegar a demorar 8 veces más en la ejecución que SVD. Si bien ahora la diferencia en segundos no parece mucha, cuando estos se trasladan a pruebas más grandes, con más ciudadanos, la diferencia sí será considerable. También la diferencia en espacio en memoria es muy considerable pues en PCA es 41 veces más grande que en SVD.

## VI. CONCLUSIONES

Del presente estudio se pueden obtener 2 conclusiones fundamentales:

Primero, aplicar SVD en el caso de reconocimiento facial es más eficiente que aplicar PCA, pues como se vio en los resultados, al aplicar solo PCA el tiempo de trabajo del sistema es 8 veces mayor al tiempo de trabajo de SVD. Mientras que la memoria utilizada también incrementa en gran magnitud cuando se usa PCA.

Segundo, usar una base de entrenamiento con un número considerable de imágenes (ej. 60) que contenga diferentes poses y muecas de la persona es mejor para que el sistema pueda aprender de toda esa variabilidad, ya que si solo se le da una imagen de

comparación (ej. solo la del DNI) esta no contendrá mayor información más que solo una pose y el sistema solo reconocerá un contorno facial en una sola posición, dificultando mucho más el reconocimiento facial.

## VII. RECOMENDACIONES

Finalmente, se le recomendaría a RENIEC que utilice una mayor cantidad de fotos si es que quiere aplicar reconocimiento facial en los ciudadanos para poder identificarlos. Lo que se podría hacer, es que cuando cada ciudadano vaya a sacar su DNI, a este no solo se le tome una foto de frente, sino que se le pida voltear de un lado al otro, sonreir, etc. Puede que sea un procedimiento que consuma más tiempo a la hora de la inscripción y almacenamiento inicial, pero al usar SVD todas las imágenes de cada ciudadano se comprimirán (a 100 x 100 pixeles) y se obtendrá una imagen promedio, aligerando así el peso. Asimismo, el sistema de reconocimiento facial será más efectivo y no se incurrirá en falsos negativos o falsos positivos que podrían después dejar en cuestionamiento a la institución.

## VIII. REFERENCIAS

[1] R. I. Perú, "Fuerza Popular, Avanza País, APP y otros 28 partidos presentaron firmas falsas para inscribirse en elecciones, según Reniec," *Infobae*, 29 de abril de 2025. [En línea]. Disponible en: <https://www.infobae.com/peru/2025/04/29/fuerza-popular-avanza-pais-app-y-otros-28-partidos-presentaron-firmas-falsas-para-inscribirse-en-elecciones-segun-reniec/>

[2] Jurado Nacional de Elecciones (JNE). *Procedimientos de inscripción de organizaciones políticas*. Recuperado de: <https://www.jne.gob.pe>

[3] Registro Nacional de Identificación y Estado Civil (RENIEC). *Verificación biométrica y autenticación de identidad*. Recuperado de: <https://www.reniec.gob.pe>

[4] Gonzalez, R. C., & Woods, R. E. (2008). *Digital Image Processing* (3.<sup>a</sup> ed.). Pearson Education.

[5] Crosswhite, N., Byrne, J., Stauffer, C., Parkhi, O., Cao, Q., & Zisserman, A. (2017). Template adaptation for face verification and identification. 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017) (págs. 1-8).

Washington, DC: IEEE.

[6] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[7] C.-H. Hsu and C.-C. Chen, "SVD-based projection for face recognition," *Applied Mathematics and Computation*, vol. 185, no. 1, pp. 176–185, Mar. 2007. doi: 10.1016/j.amc.2006.06.088.

[8] G. Zeng, "Facial Recognition with Singular Value Decomposition," in *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, K. Elleithy, Ed. Springer, 2007, pp. 145–148.

[9] J. C. Hernández, "Reconocimiento facial en sub-espacios: lineales y no lineales," *ARJE: Revista de Investigación*, vol. 22, no. 2, pp. 1–12, 2019. [En línea]. Disponible en: <https://arje.bc.uc.edu.ve/arje22e/art41.pdf>

### \*Uso de Inteligencia Artificial

- Nombre: Chat GPT
- Versión: 4o
- Prompts empleados:
  - Ordename el código que te estoy adjuntando y en base a ello dame la matriz de confusión y la matriz de singularidad
  - Dame un código de python que aplique la librería OS para guardar una imagen tomada en tiempo real
- Comentario: El uso de la IA nos permitió poder ordenar nuestras ideas e implementar gráficos que den mayor interpretabilidad a los resultados que obtuvimos con los métodos de PCA y SVD. También nos ayudó a usar librerías que no conocíamos para mejorar el uso técnico de nuestra solución (usando imágenes en tiempo real).

