

# Aseguramiento de la Calidad del Software: Proyecto Semestral I

M. Sc. Saúl Calderón Ramírez  
Instituto Tecnológico de Costa Rica,  
Escuela de Ingeniería en Computación,  
PAttern Recognition and MACHine Learning Group (PARMA-Group)

3 de agosto de 2017

El presente proyecto pretende ser desarrollado en grupos de tres personas, a lo largo del curso de aseguramiento de la calidad del software. El proyecto será insumo para aplicar distintos estándares de calidad en sus varias etapas, por lo que a lo largo del curso se realizarán tareas y agregados específicos al proyecto. El objetivo es desarrollar un sistema biométrico de identificación a partir de la imagen digital del rostro.

**Fecha de entrega: 11 de Noviembre.**

## 1. Introducción y motivación

La biometría (del griego *bios* vida y *metron* medida) estudia métodos y algoritmos para el reconocimiento inequívoco de personas basado en características conductuales o físicas de los sujetos. La construcción de sistemas de reconocimiento de rostros, huellas dactilares o iris oculares para fines como por ejemplo la autenticación de usuarios en diversos sistemas ha sido un campo de extenso estudio en el reconocimiento de patrones, para los cuales se han desarrollado distintos enfoques. La confiabilidad y precisión de estos sistemas a alcanzado grados considerables, lo que hace posible la implementación de por ejemplo el reconocimiento del usuario de un teléfono celular por huellas dactilares, el reconocimiento de los usuarios de Facebook por rostro. Otros ejemplos de sistemas biométricos son:

- El sistema de validación de frontera SmartGate, implementado en Australia y Nueva Zelanda utiliza el reconocimiento automático de rostros para identificar el portador del pasaporte. <https://www.border.gov.au/Trav/Ente/GoIn/Arrival/Smartgateor-ePassport>
- La aplicación Vision Applock usa la cámara digital del teléfono para identificar al usuario permitido para usar aplicaciones específicas <https://>



Figura 1: Los sistemas biométricos son utilizados con mayor frecuencia en distintos ámbitos.

`play.google.com/store/apps/details?id=visidon.AppLockPlus&hl=en.`

- El aeropuerto internacional de Tocumen en Panamá opera un sistema de identificación de individuos buscados por la justicia internacional <https://findbiometrics.com>.

Dada la eficacia lograda en muchos de estos sistemas, la preocupación de la sociedad civil por el acoso de la privacidad ciudadana es creciente, por lo que existen iniciativas para prohibir la aplicación de tales sistemas en algunos ámbitos, recientemente con éxito.

En el presente proyecto semestral se implementará un sistema de reconocimiento de sujetos a partir de imágenes digitales de rostros. El sistema identificará una imagen de rostro a partir del conocimiento inferido a partir de un banco de imágenes digitales de rostros de todos los usuarios del sistema. Para implementarlo se usará el Análisis de Componentes Principales (ACP), una técnica estadística que descarta la información poco relevante y preserva la más importante. A continuación se detalla el marco teórico de tal técnica, para posteriormente

## 2. Análisis de Componentes Principales: Marco teórico

A continuación se abordan los aspectos generales de la técnica de análisis de componentes principales, lo cuales se detallan con mayor profundidad en el documento de apéndice.

La técnica de *auto-caras* es una técnica de aprendizaje supervisado que básicamente conceptualiza la imagen digital de un rostro representada en una matriz  $U_i \in \mathbb{R}^{n \times m}$  como un vector  $\vec{v}_i \in \mathbb{R}^p$ , con  $p = n \times m$ . El problema de clasificación de rostros según el sujeto  $t = 1, \dots, S$  en una base de datos muestral de  $S$  sujetos definida en el conjunto  $M = \{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_r\}$  con  $r$  muestras en total, radica entonces para una muestra  $\vec{v}_i$  estimar su etiqueta correspondiente

$t_i$ . El enfoque de auto-caras para la estimación de tal etiqueta, utiliza un algoritmo de aprendizaje supervisado no paramétrico como por ejemplo K-vecinos más cercanos o el centroide más cercano, en un espacio de dimensionalidad reducida y con menor presencia de ruido, conformado por las *auto-caras*.

Para construir el espacio de dimensionalidad reducida, el enfoque de auto-caras calcula los auto-vectores con mayor auto-valor de la matriz de covarianza  $\Sigma$ . La matriz de covarianza refleja las dimensiones en las que los datos *varían más*, es decir, aportan la mayor información. Los auto-vectores de tal matriz indican la dirección de tales dimensiones, por lo que son útiles para construir un nuevo espacio de dimensionalidad reducida  $p' \ll p$ , el cual preserve la mayor cantidad de información posible, descartando entonces las dimensiones que aportan la menor información, usualmente asociada al ruido.

Una vez construido el nuevo espacio vectorial con las *auto-caras* definidas en el conjunto  $E = \{\vec{e}_1, \vec{e}_2, \dots, \vec{e}_{p'}\}$ , cada una de las muestras de entrenamiento  $M$  es proyectada en el espacio  $E$ , creando un nuevo conjunto de muestras  $M' = \{\vec{v}'_1, \vec{v}'_2, \dots, \vec{v}'_r\}$ .

Finalmente, para una nueva muestra a clasificar  $\vec{m} \in \mathbb{R}^p$ , se implementará el algoritmo de clasificación supervisado no paramétrico del centroide más cercano. Tal algoritmo consiste en calcular, por cada sujeto  $t = 1, \dots, S$  el rostro promedio de sus muestras en el nuevo conjunto de muestras  $M' = \{\vec{v}'_1, \vec{v}'_2, \dots, \vec{v}'_r\}$ , obteniendo el conjunto de *rostros promedio*  $A = \{\vec{\mu}_1, \vec{\mu}_2, \dots, \vec{\mu}_S\}$ . De esta forma, para clasificar una nueva muestra  $\vec{m} \in \mathbb{R}^p$ , se realiza primero la proyección al espacio de *auto-caras*  $E$ , generando la nueva muestra de dimensionalidad reducida  $\vec{m}' \in \mathbb{R}^{p'}$ , y posteriormente se calcula la muestra promedio  $\vec{\mu}_{\min}$  más cercana, usando la distancia euclidiana o norma  $\ell_2$ :

$$\vec{\mu}_{\min} = \arg \min_{\vec{\mu}_i} \|\vec{m}' - \vec{\mu}_i\|$$

el valor de la etiqueta más cercana mín será la etiqueta estimada por el clasificador.

En términos generales, las siguientes son las etapas del enfoque de auto-caras para el reconocimiento de sujetos por imágenes digitales de rostro:

1. A partir de la base de datos muestral  $M = \{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_r\}$ , construir la **matriz de covarianza**  $\Sigma$ .
2. Para la matriz de covarianza  $\Sigma$ , **calcular los  $p'$  auto-vectores más importantes** (según la magnitud del auto-valor).
3. **Proyectar** los datos en la base de datos muestral  $M$  para crear la base de datos muestral con muestras de dimensión reducida  $M' = \{\vec{v}'_1, \vec{v}'_2, \dots, \vec{v}'_r\}$ .
4. Implementar el enfoque de clasificación de rostros promedio o K-vecinos más cercano, para nuevas muestras, usando la **distancia euclidiana**.

### 3. Métricas

Las métricas se especificarán con detalle en el documento de apéndice.

## 4. Requerimientos de la aplicación

La aplicación debe proveer una interfaz gráfica de usuario amigable, con dos modos: entrenamiento e identificación de nuevos sujetos. Tales modos se explican a continuación.

### 4.1. Modo de entrenamiento

El modo de entrenamiento consiste en el cálculo de las auto-caras o *eigen-faces*, así como la proyección de todas las muestras en el nuevo espacio de dimensionalidad reducida de auto-caras.

1. Cargar un conjunto de imágenes almacenada en la dirección provista por el usuario para el entrenamiento. El conjunto de imágenes provisto debe contener una carpeta por cada sujeto, y se debe almacenar el nombre de tal carpeta como la etiqueta de tal sujeto o clase.
2. Entrenar el sistema, generando las *auto-caras* y la proyección de las muestras en el nuevo espacio formado por tales *auto-caras*, usando el algoritmo supervisado del centroide más cercano. Tanto las auto-caras como las proyecciones deben ser almacenadas usando algún esquema de almacenamiento de su elección.
  - a) El parámetro del entrenamiento será la cantidad de auto-vectores a conservar en el nuevo conjunto de vectores base.
3. Cargar un conjunto de muestras de prueba, para medir la precisión del sistema.
  - a) Deben generarse las métricas a estipularse en el curso para medir la precisión del sistema, con un informe en formato *.csv*. Las métricas a usarse como mínimo serán los falsos positivos, falsos negativos, la precisión y el *recall*. Presente un informe con los resultados obtenidos para distintos porcentajes de muestras usadas para el entrenamiento y pruebas.

### 4.2. Modo de identificación de sujetos

El modo de identificación de sujetos consiste en una fachada que emula la identificación de un sujeto usando el sistema de *auto-caras* pre-entrenado. Los requerimientos del sistema en este modo son los siguientes:

1. Cargar las auto-caras y las muestras proyectadas de un sistema entrenado previamente.
2. Permitir la carga de una imagen correspondiente a un rostro, y ejecutar la identificación del sujeto. El sistema debe desplegar la etiqueta correspondiente a tal sujeto.

### 4.3. Puntos extra

Los requerimientos extra de la aplicación consisten en:

1. Implementar el algoritmo de clasificación supervisado no paramétrico de los K-vecinos más cercanos. El valor K debe ser un parámetro modificable por el usuario. Comparar los resultados con el algoritmo de clasificación del centroide más cercano, para distintos valores de K.
2. Genere los datos de uno o más nuevos sujeto, e incluyalo en la base de datos.
  - a) Implemente un algoritmo que normalice automáticamente los datos (iluminación, desplazamiento).
  - b) Incorpore un módulo que permita la lectura de nuevas imágenes desde una cámara web y documente las pruebas para esos nuevos sujetos.

## 5. Implementación

Para el diseño e implementación del proyecto, es necesario usar la metodología de desarrollo Scrum, usando alguna herramienta para gestión de proyectos en línea como *zoho projects*. Como herramientas de diseño, se recomienda usar *StarUML*, *Altova*, *Visio*, etc. Para promover el atributo de mantenibilidad del proyecto, al menos un patrón de diseño debe ser implementado.

Respecto a la codificación del proyecto, es muy recomendado el uso del lenguaje Python, usando Eclipse como ambiente de desarrollo, pues existen herramientas y *plugins* relacionados con el cálculo de métricas y gestión de la calidad del proyecto, lo cual será un rubro importante a evaluar. **Los estándares de calidad a aplicar se agregarán a lo largo del curso.** Es conocido el soporte de Java con OpenCV además, librería de algoritmos de visión por computador y aprendizaje automático.

La documentación final del proyecto debe incluir todos los estándares implementados a lo largo del curso, incluyendo los realizados en tareas, de manera consistente. Es obligatorio el uso de LaTeX o algún editor basado en esa tecnología como LyX para la documentación del proyecto. Para facilitar la edición colaborativa, se recomienda el uso de la herramienta *Overleaf*.