

Aseguramiento de la Calidad del Software:

Avance 3, proyecto semestral

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

17 de octubre de 2017

En el presente avance, se desarrollarán las funcionalidades críticas del proyecto, usando criterios de aseguramiento de la calidad.

Fecha de entrega: Martes 15 de Noviembre.

Entrega: A través del TEC digital.

Valor: 15 % del proyecto I semestral.

Parte I

Avance 3 del proyecto

En el último avance del proyecto, se integrarán actividades de verificación dinámica, y se mantendrán las tareas prioritarias definidas en el avance anterior.

1. Necesidades prioritarias para el proyecto

1.1. Modo de entrenamiento

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.

| Clase de equivalencia | Válida |
|-----------------------|--------|
| | |

Figura 1: Tabla de clases de equivalencia por componente.

- 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.

1.2. 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, usando el algoritmo del **centroide más cercano**. El sistema debe desplegar la etiqueta correspondiente a tal sujeto.

2. Requerimientos para el «sprint»

1. Diseñe al menos cuatro pruebas unitarias, cuatro de integración y dos de sistema.
 - a) De los módulos o componentes a probar, intente definir sus «clases de equivalencia», y defina si es una clase de equivalencia válida o no válida. Puede usar la siguiente tabla:
 - b) Si alguna prueba necesita de un módulo no terminado, reemplacelo por un *stub* y documente su funcionamiento.
 - c) Use la siguiente tabla para realizar el diseño de las pruebas:
 - d) Cada prueba debe ser asociada a una clase de equivalencia. Para ello use la siguiente tabla:

| Id. de la prueba | Tipo de prueba | Descripción | Precondiciones | Resultados esperados |
|------------------|----------------|-------------|----------------|----------------------|
| | | | | |

Figura 2: Tabla de de diseño de pruebas.

| Clase de equivalencia | Id. de la prueba |
|-----------------------|------------------|
| | |

Figura 3: Tabla de de diseño de pruebas.

- e) Al menos una prueba de sistema debe utilizar la herramienta *Selenium*, por lo cual debe probarse la integración total del sistema con la interfaz gráfica.
- f) Todas las pruebas deben escribirse usando un paquete para pruebas unitarias . Adjunte la dirección del repositorio *git* el cual contenga la implementación de las pruebas.
- g) Adjunte evidencias de que las pruebas fueron ejecutadas correctamente.