

---

## **TERCERA PRUEBA DE EVALUACIÓN (PRÁCTICA)**

**Diseño e implementación de algoritmos de reconocimiento de objetos basados en estrategias de selección de características y clasificación**

---

### **CONVOCATORIA:**

- **DÍA Y HORA:** lunes, día 10 de junio, 16:00h
- **LUGAR:** aula 1.1 ISIDORO MORALES

### **FORMATO Y CRITERIOS DE EVALUACIÓN:**

- **Descripción:** prueba práctica con Matlab. Se debe tener disponible todo el material práctico generado en la práctica 5 y en los ejercicios de evaluación propuestos en este documento.
- **Contenido:** diseño e implementación de sistemas de reconocimiento de objetos de acuerdo a una serie de especificaciones.
- **Consideraciones:**
  - **Clases:** el problema de reconocimiento involucrará a un máximo de 8 clases.
  - **Descriptores matemáticos de partida:**
    - Se trabajará inicialmente con los 23 descriptores utilizados en la práctica 5.
    - Se especificará el número de descriptores con los que se debe trabajar (dimensión del vector de características).
    - Se podrá establecer que se trabaje con unos descriptores determinados o que se elijan mediante una metodología de selección de características.



Universidad  
de Huelva

## SISTEMAS DE PERCEPCIÓN

Grado en Ingeniería Informática – itinerario Computación

**Dpto. de Ingeniería Electrónica, Sistemas Informáticos y Automática**

**EVALUACIÓN CONTINUA – PRIMERA CONVOCATORIA – 2023/24**

- **Datos de entrada para la resolución de estos ejercicios:** se facilitarán datos ya generados de los 23 descriptores (dispuestos en columnas en el orden que se indica a continuación; las filas serán sus cálculos sobre los objetos disponibles) o bien las imágenes que permiten su generación.

Orden de Descriptores:

```
% Compacticidad: 1
% Excentricidad: 2
% Solidez_CHull(Solidity): 3
% Extension_BBox(Extent): 4
% Extension_BBox(Invariante Rotacion): 5
% Hu1-Hu7: 6-12
% DF1-DF10: 13-22
% NumEuler: 23
```

- Se podrá requerir que se representen histogramas o diagramas de caja de los valores de algún descriptor en las muestras disponibles de las diferentes clases del problema.
- **Técnicas de clasificación para problemas genéricos de K clases:** se podrá establecer que se trabaje con una técnica de clasificación específica (LDA, QDA o KNN).
- **Diseño de clasificador LDA para clasificación binaria:** en el caso de clasificación binaria (2 clases), se podrá requerir que se genere la función discriminante entre las dos clases en lenguaje simbólico, así como los coeficientes que definen la ecuación del hiperplano de separación. En casos de vectores de predictores de dimensión 2 o 3, también se podrá requerir la representación de ese hiperplano de separación en el espacio de características junto con los datos de entrenamiento de cada clase.
- **Implementación del sistema de reconocimiento:** creación de un algoritmo que, dada una imagen de entrada, visualice en distintas ventanas de tipo figure cada objeto que hay en la imagen, especificando a través del título el reconocimiento de la clase a la que pertenece. Para ello, el algoritmo debe cargar la información necesaria para la aplicación de los clasificadores diseñados en las etapas anteriores.
- **Representación de datos:** en caso de trabajar en dimensión 2 o 3, se podrá requerir la representación de la descripción matemática de los objetos desconocidos en el espacio de características junto con los datos de entrenamiento de cada clase.



Universidad  
de Huelva

## SISTEMAS DE PERCEPCIÓN

Grado en Ingeniería Informática – itinerario Computación

**Dpto. de Ingeniería Electrónica , Sistemas Informáticos y Automática**

**EVALUACIÓN CONTINUA – PRIMERA CONVOCATORIA – 2023/24**

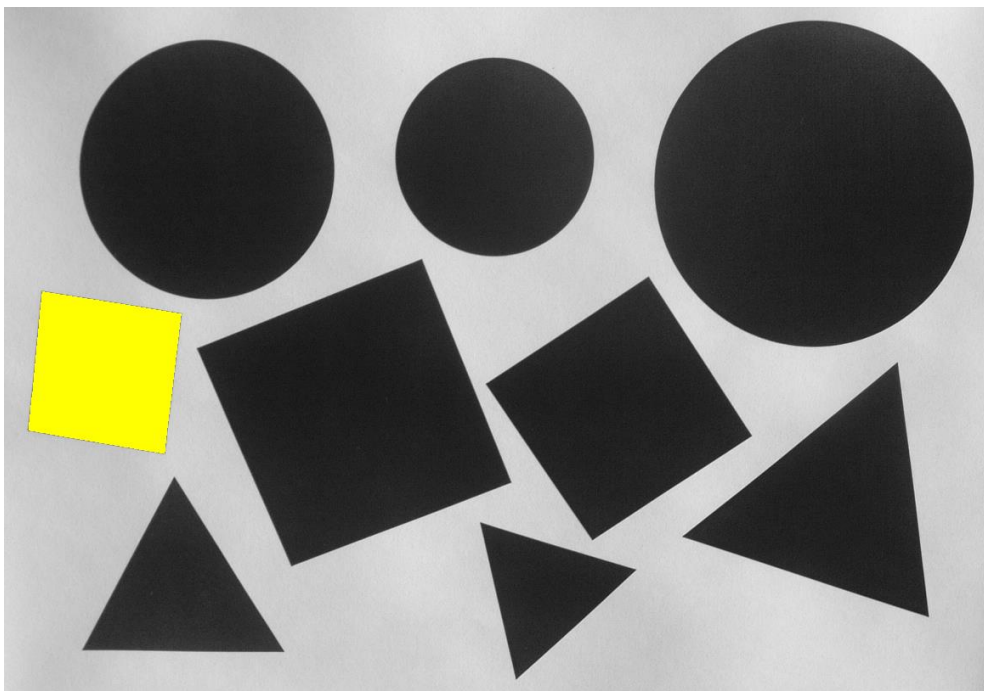
### **EXAMEN PRÁCTICO TIPO: EJEMPLO 1 (SOBRE PRÁCTICA 5)**

Implementar un algoritmo que reconozca la forma de cada uno de los objetos (circular, cuadrada o triangular) que componen las imágenes de test facilitadas, a partir de la siguiente función:

`Funcion_Reconoce_Formas (Nombre)`

- La función debe recibir como entradas la variable `Nombre` (cadena de texto con el nombre del archivo de la imagen a tratar).
- La función debe abrir una ventana tipo `figure` por cada objeto de la imagen, donde se visualice, sobre la imagen original de entrada, el resultado de la segmentación del objeto en cuestión y cuyo título muestre el resultado del reconocimiento de la forma del objeto: circular, cuadrada o triangular.

Forma Cuadrada





Universidad  
de Huelva

---

## SISTEMAS DE PERCEPCIÓN

Grado en Ingeniería Informática – itinerario Computación

**Dpto. de Ingeniería Electrónica , Sistemas Informáticos y Automática**

**EVALUACIÓN CONTINUA – PRIMERA CONVOCATORIA – 2023/24**

---

### MATERIAL DISPONIBLE:

- **Imágenes de Entrenamiento:** 2 imágenes por cada tipo de objeto, con 15 objetos cada una. Para diseñar el algoritmo
- **Imágenes de test:** 2 imágenes con 9 objetos cada una de distintos tipos (circulares, cuadrados o triangulares). Para evaluar el funcionamiento del algoritmo diseñado.

### ESPECIFICACIONES DE DISEÑO:

Estrategia de clasificación basada en dos etapas:

1. Reconocer entre Círculos-Cuadrados y Triángulos utilizando un clasificador KNN ( $K = 5$ ) que trabaje sobre los 3 descriptores que proporcionen la mayor separabilidad conjunta.
2. Reconocer entre Círculos y Cuadrados utilizando un clasificador LDA basado en la siguiente descripción matemática: **DF3, DF7, Hu3**.

### ANÁLISIS DE RESULTADOS:

Justificar los resultados obtenidos en las imágenes de test a partir de la representación en el espacio de características de los patrones de entrenamiento de los distintos objetos y del plano de separación utilizado para el reconocimiento entre las clases círculo y cuadrado.

### ENTREGA DE CÓDIGO MATLAB:

Se debe organizar y entregar el código siguiendo el criterio y estructura explicada en la práctica 5.

### **EXAMEN PRÁCTICO TIPO: EJEMPLO 2**

Implementar un algoritmo que reconozca entre los caracteres (A, B, C, D, E, F y G) que componen las imágenes de test facilitadas, a partir de la siguiente función:

`Funcion_Reconoce_Letras (Nombre_Imagen)`

- La función debe recibir como entradas las variables `Nombre_Imagen` (cadena de texto con el nombre del archivo de la imagen a tratar). Para evitar pequeños objetos ruidosos en la segmentación de objetos, se deben considerar únicamente las agrupaciones conectadas de píxeles que tengan un número mínimo de píxeles (establecido de acuerdo a algún criterio).
- La función debe abrir dos ventanas tipo `figure` por cada objeto de la imagen:
  1. Ventana donde se visualice, sobre la imagen original de entrada, el resultado de la segmentación del objeto en cuestión y cuyo título muestre el resultado del reconocimiento del objeto.
  2. Ventana donde se representen los valores de los descriptores del conjunto de entrenamiento utilizados para reconocer el objeto, los del objeto en cuestión, así como las fronteras de separación entre las clases cuando proceda (sólo cuando la clase del objeto a analizar NO sea Letra B). Cuando la dimensión sea superior a 3, el espacio de características de la representación estará definido por los 3 primeros descriptores.

### **ESPECIFICACIONES DE DISEÑO:**

1. Reconocimiento inicial por N° de Euler de los siguientes conjuntos de letras: A-D, B, y C-E-F-G.
2. Reconocimiento de las letras de cada conjunto:
  - 2.1. Para reconocer entre la A y la D, se deben aplicar los descriptores *Extensión (Inv. Rotación)*, *Hu1* y *Hu2* y aplicar un clasificador LDA.
  - 2.2. Para reconocer entre la C, E, F y G se deben buscar los 4 mejores descriptores y utilizar como técnica de reconocimiento el clasificador QDA.

### MATERIAL DISPONIBLE (Directorio *ImágenesLetras*):

- **Imágenes de Entrenamiento:** 2 imágenes por cada clase de salida. Para diseñar y entrenar los algoritmos de clasificación.
- **Imágenes de test:** 1 imagen por cada clase de salida. Para la evaluación visual la herramienta de clasificación diseñada.

### ENTREGA DE CÓDIGO MATLAB, ORGANIZADO EN LAS SIGUIENTES CARPETAS:

- 1. Carpeta 1: generación de datos de entrenamiento.** Se debe entregar el código matlab que genere los datos estandarizados (media 0 y desviación típica 1) de los descriptores considerados calculados sobre todos los objetos de las imágenes de entrenamiento.
- 2. Carpeta 2: Diseño Clasificador LDA para reconocer entre las letras A y D.** Se deben entregar dos scripts, cada uno de ellos realizando las siguientes acciones:
  - Diseño del clasificador LDA.
  - Representación en el espacio de características los datos de entrenamiento de las clases junto con el plano de separación definido por el clasificador utilizado. La gráfica generada debe especificar el nombre de los ejes y las leyendas correspondientes a la información mostrada.
- 3. Carpeta 3: Diseño Clasificador QDA para reconocer entre las letras C, E, F y G.**

Se deben entregar dos scripts, cada uno de ellos realizando las siguientes acciones:

  - Selección de los 4 descriptores que proporcionan la mayor separabilidad conjunta.
  - Diseño del clasificador QDA.
- 4. Carpeta 4: Algoritmo de reconocimiento.** Se debe entregar la función solicitada `Funcion_Reconoce_Letras (Nombre_Imagen)` preparada para ejecutarse sobre las imágenes de test.

### **EXAMEN PRÁCTICO TIPO: EJEMPLO 3 (VARIANTE DEL EJERCICIO 2)**

Implementar un algoritmo que reconozca entre los caracteres (A, B, C, D, E, F y G) que componen las imágenes de test facilitadas, a partir de la siguiente función:

`Funcion_Reconoce_Letras (Nombre_Imagen)`

- La función debe recibir como entradas las variables `Nombre_Imagen` (cadena de texto con el nombre del archivo de la imagen a tratar). Para evitar pequeños objetos ruidosos en la segmentación de objetos, se deben considerar únicamente las agrupaciones conectadas de píxeles que tengan un número mínimo de píxeles (establecido de acuerdo a algún criterio).
- La función debe abrir dos ventanas tipo `figure` por cada objeto de la imagen:
  1. Ventana donde se visualice, sobre la imagen original de entrada, el resultado de la segmentación del objeto en cuestión y cuyo título muestre el resultado del reconocimiento del objeto.
  2. Ventana donde se representen los valores de los descriptores del conjunto de entrenamiento utilizados para reconocer el objeto y los del objeto en cuestión (sólo cuando la clase del objeto a analizar NO sea Letra B). Cuando la dimensión sea superior a 3, el espacio de características de la representación estará definido por los 3 primeros descriptores.

### **ESPECIFICACIONES DE DISEÑO:**

1. Reconocimiento inicial por N° de Euler entre la letra B y el siguiente subconjunto de letras: A-C-D-E-F-G
2. Reconocimiento entre los siguientes subconjuntos de letras: A-C-D y E-F-G  
Se deben utilizar los descriptores Hu3, DF4 y DF6. Como técnica de clasificación, se debe utilizar LDA.

3. Reconocimiento de las letras del subconjunto A-C-D:

Se deben buscar los 3 mejores descriptores y utilizar como técnica de reconocimiento un clasificador k-NN con  $k = 5$ .

4. Reconocimiento de las letras del subconjunto E-F-G:

Se deben buscar los 4 mejores descriptores y utilizar como técnica de reconocimiento un clasificador QDA.

### MATERIAL DISPONIBLE:

- **Imágenes de Entrenamiento:** 2 imágenes por cada clase de salida. Para diseñar y entrenar los algoritmos de clasificación.
- **Imágenes de test:** 1 imagen por cada clase de salida. Para la evaluación visual la herramienta de clasificación diseñada.

### ENTREGA DE CÓDIGO MATLAB, ORGANIZADO EN LAS SIGUIENTES CARPETAS:

**1. Carpeta 1: generación de datos de entrenamiento.**

Se debe entregar el código matlab que genere los datos estandarizados (media 0 y desviación típica 1) de los descriptores considerados calculados sobre todos los objetos de las imágenes de entrenamiento.

**2. Carpeta 2: Diseño Clasificador LDA para reconocer entre los grupos de letras A-C-D y E-F-G.** Se deben entregar dos scripts, cada uno de ellos realizando las siguientes acciones:

- Diseño del clasificador LDA.





Universidad  
de Huelva

---

## SISTEMAS DE PERCEPCIÓN

Grado en Ingeniería Informática – itinerario Computación

**Dpto. de Ingeniería Electrónica, Sistemas Informáticos y Automática**

**EVALUACIÓN CONTINUA – PRIMERA CONVOCATORIA – 2023/24**

---

- Representación en el espacio de características los datos de entrenamiento de las clases junto con el plano de separación definido por el clasificador utilizado. La gráfica generada debe especificar el nombre de los ejes y las leyendas correspondientes a la información mostrada.

### 3. Carpeta 3: Diseño Clasificador KNN para reconocer entre las letras A, C y D.

Se deben entregar dos scripts, cada uno de ellos realizando las siguientes acciones:

- Selección de los 3 descriptores que proporcionan la mayor separabilidad conjunta.
- Generación de la información que requiere la aplicación del clasificador.

### 4. Carpeta 4: Diseño Clasificador QDA para reconocer entre las letras E, F y G.

Se deben entregar dos scripts, cada uno de ellos realizando las siguientes acciones:

- Selección de los 4 descriptores que proporcionan la mayor separabilidad conjunta.
- Generación de la información que requiere la aplicación del clasificador.

### 5. Carpeta 5: Algoritmo de reconocimiento.

Se debe entregar la función solicitada `Funcion_Reconoce_Letras(Nombre_Imagen)` preparada para ejecutarse sobre las imágenes de test.