

Manual de usuario

Contenido

Introducción3
Manual de usuario4
Ventana principal5
Configuración de la cámara8
Calibración del sistema11
Catálogo20
Configuración campaña27
Lances
Enviar resultados
Estructura del directorio de la aplicación35
Modos de funcionamiento40
Modo tosteo do
Modu (esteu
Parámetros que modifican el modo de funcionamiento42
Parámetros que modifican el modo de funcionamiento42 Instalación43
Parámetros que modifican el modo de funcionamiento42 Instalación
Parámetros que modifican el modo de funcionamiento
Parámetros que modifican el modo de funcionamiento
Parámetros que modifican el modo de funcionamiento
Parámetros que modifican el modo de funcionamiento
Parámetros que modifican el modo de funcionamiento
Parámetros que modifican el modo de funcionamiento
Modo testeo

Introducción

El objetivo de *iSEAS* es disponer de una aplicación de escritorio para la clasificación de pescado, a su paso por una cinta transportadora, según un catálogo de especies conocidas y pre-caracterizadas.

Esta aplicación permite configurar el sistema en cuanto a:

- Identificar la instalación.
- Determinar los parámetros de la cámara para la captura de imágenes con la mayor calidad posible y dentro de la región de interés, la cinta transportadora.
- Calibrar el sistema para realzar los objetos de la imagen y aislarlos del fondo.
- Generar el catálogo con las diferentes especies de interés.
- Gestionar campañas, que agrupan las especies a reconocer en cada momento.

Con las especies a reconocer, definidas en la campaña seleccionada, el sistema clasifica el pescado a su paso por una cinta transportadora y genera un archivo de resultados con el detalle del tamaño y peso de los ejemplares de pescado reconocidos.

Manual de usuario

La principal utilización del sistema es su aplicación en la clasificación de pescado desde el módulo *Lances*, y es por ello que el programa se inicia en este módulo.

Desde ahí, a través del botón **Administración**, se accede a la ventana principal de la aplicación, que da acceso a los diferentes módulos que constituyen el sistema y que permiten su configuración antes de realizar el procesado de clasificación.

T ISEAS Barco INSTALAC	iSE4	AS - Lances Núme 1	2ro	
			Empezar lance	
image			Finalizar lance	
		Guardar una	de cada diez imágenes o de imágenes capturadas por minuto	Parar lance Reanudar lance

En todas las ventanas del sistema se visualizan los logotipos del proyecto y de la cofinanciación:

• Logotipo del proyecto:



• Logotipo de publicidad de cofinanciación de proyectos LIFE:



Ventana principal

En la ventana principal se identifica la instalación y se da acceso a los diferentes módulos del sistema.

3			ISEAS_		- 0 ×
1	iSEAS 7	alactin Barco INSTALACION_1		Número 1 v Modificar	<i>tite</i>
		Configuración climara	Calbración del sistema	Catiliogo	
		Configuración campaña	Lances	Errvier resultados	
				Sa	lir

• En la parte superior de la ventana se visualizan los datos de identificación de la instalación, nombre y número.

El nombre será el nombre del barco o identificación de la ubicación del sistema, y el número será el número de instalación dentro del mismo barco o ubicación.

La identificación se puede modificar pulsando el botón Modificar.

- En la parte inferior de la ventana se sitúa el botón **Salir**, que permite salir y cerrar la aplicación.
- En la parte central de la ventana se muestran los botones que dan acceso a los diferentes módulos del sistema:
 - **Configuración cámara**: da acceso al módulo de configuración de la cámara y captura de imágenes.

En este módulo se ajustan los parámetros de la cámara para capturar imágenes con la mayor calidad posible, y se determina la región de interés dentro del campo visual de la cámara.

• **Calibración del sistema**: da acceso al módulo de calibración del color, las distancias y el fondo de las imágenes.

A través de este módulo se determinan:

- Los niveles del blanco y el negro según las condiciones lumínicas de la instalación y con ayuda de una plantilla de calibración.
- La relación entre las distancias reales y las distancias dentro de la imagen.
- La caracterización del fondo de las imágenes, la cinta transportadora, para poder extraerlo de la imágenes a procesar durante la clasificación de especies.
- **Catálogo**: da acceso al módulo de generación y gestión del catálogo de especies.

A través de este módulo, se pueden capturar imágenes, asociarlas a una determinada especie, y generar el patrón que la caracteriza.

 Configuración campaña: da acceso al módulo de generación y gestión de campañas.

A través de este módulo se determinan las especies que se quieren reconocer en cada campaña, y se establece la campaña que estará activa durante el proceso de clasificación.

• Lances: da acceso al módulo que lanza el proceso de reconocimiento de especies.

A través de este módulo se puede iniciar, parar, reanudar y finalizar el reconocimiento.

Al finalizar el procesado de reconocimiento, el sistema genera el fichero de resultados con el detalle de las especies reconocidas.

• Enviar resultados: envía los ficheros de resultados a un determinado directorio de red o local.

Los módulos *Configuración cámara, Calibración del sistema, Catálogo* y *Lances,* utilizan la conexión con la cámara para la captura de imágenes.

Si al lanzar cualquiera de estos módulos, el sistema no puede abrir comunicación con la cámara, se muestra el siguiente mensaje informativo y no da acceso al módulo seleccionado.



Configuración de la cámara

En este módulo se ajustan los parámetros de la cámara para capturar imágenes con la mayor calidad posible, y determinar la región de interés dentro del campo visual de la cámara.

Estos parámetros deben ajustarse en el momento de la instalación del sistema y cuando se modifican la iluminación y/o se desplaza la cabina de captura de imágenes.

Se accede a este módulo a través del botón **Configuración cámara** de la ventana principal.

	ISEAS -	Configuración de la cámara	
ISEAS	Barco INSTALACION_1	Número 1. v	.16
		Ajustar los parámetros de la cámara Ganancia 0 +	ntos return ref so de
		Sali	

A la izquierda de la ventana se ve la imagen de salida, imagen que captura la cámara en tiempo real.

A la derecha de la ventana se accede a los controles para modificar los parámetros de la cámara y la región de interés.

Gariarida		-	CLARKE CHILDREN
0	+		Guardar cambior
Tiempo de es	rposición		Course our currents
0	+	-	Recuperar parámetros de fábrica
	i and		
marcar la región (de interés		
marcar la región o	de interés +	1	
marcar la región (le interés + -]	
marcar la región (de interés + + +]
marcar la región (le interés + + + -]	

• En la sección **Ajustar los parámetros de la cámara**, a través de los controles + y -, se pueden variar la **ganancia** y el **tiempo de exposición**.

Estos valores también se pueden modificar tecleando el valor requerido en el cuadro de edición correspondiente.

Con el botón **Probar cambios**, se puede ver cómo influyen las variaciones en la imagen de salida.

Con el botón **Guardar cambios**, se guardan los últimos valores comprobados con el botón *Probar* cambios.

Con el botón **Recuperar parámetros de fábrica**, se ajusta la cámara y se guardan los parámetros iniciales del sistema.

Variando estos parámetros se deben buscar los valores que permitan capturar las imágenes de la forma más nítida y clara posible, sin demasiados brillos ni zonas quemadas (blancas).

• En la sección **Enmarcar la región de interés**, a través de los controles + y -, se puede variar el marco que encuadra la zona de la imagen que va a ser procesada.

Los controles + y - desplazan la línea del marco hacia el margen de la imagen y hacia el centro de la imagen respectivamente.

Los controles superiores desplazan la línea superior del marco, los controles inferiores desplazan la línea inferior del marco, los controles de la izquierda desplazan la línea izquierda del marco, y los controles de la derecha desplazan la línea derecha del marco.

Con el botón **Guardar marco**, se guarda como marco para el procesado de la imágenes, el que se visualiza en ese momento en la imagen de salida de la zona de la izquierda de la ventana.

El objetivo es delimitar la zona de la imagen que va a ser procesada, y ésta sólo debe incluir la cinta transportadora, y dejar fuera cualquier objeto que entre dentro del campo visual de la cámara y que quede fuera de la cinta como el ejemplo siguiente.



Con el botón **Salir** de esta ventana, se sale del módulo *Configuración de la cámara*.

Calibración del sistema

Este módulo da acceso a los distintos procesos de calibración: calibración del color, calibración de las distancias, y calibración del fondo.

Debe realizarse la calibración del sistema cada vez que se modifican las condiciones de éste como la iluminación, posición o fondo de las imágenes capturadas.

Se accede a este módulo a través del botón **Calibración del sistema** de la ventana principal.

iseas	Barco INSTALACION_1	Número	.46e
anana anana		Calibrar color Última calibración: Antes de empezar la ca	03/05/2017 Ilbración, coloque la plantilla de forma
JANJAA JANJAA	mattatespart	que quede dentro del c Calibrar distancias Última calibración:	26/03/2017
		Medidas del marco blanc Alto	co (mm) Ancho
ananan ananan ananan	annananannan annananan annanananan	Calibrar fondo Última calibración: Antes de empezar la cal cuadro blanco no hay ni	07/05/2017 libración, asegúrese de que dentro del ngún objeto.

A la izquierda de la ventana se ve la imagen de salida, imagen que captura la cámara en tiempo real.

A la derecha de la ventana se accede a los controles para la calibración del sistema y se visualiza la última fecha en que se ejecutó cada tipo de calibración.

En la parte inferior izquierda, el botón **Prueba extraer fondo**, ayuda al cálculo de los coeficientes necesarios para el proceso de extracción del fondo de las imágenes.

Calibrar color	
Última calibración:	26/02/2017
Antes de empezar la calib que quede dentro del cua	ración, coloque la plantilla de forma adro blanco.
Calibrar distancias	
Última calibración:	26/02/2017
Medidas del marco blanco	(mm)
Alto	Ancho :
Calibrar fondo	
Última calibración:	26/02/2017
Antes de empezar la calibi cuadro blanco no hay ning	ración, asegúrese de que dentro del gún objeto.

• Calibración del color – con ayuda de la plantilla de calibración del Anexo I, calcula los niveles del blanco y el negro según las condiciones lumínicas de la instalación.

Última calibración indica la fecha en la que se registró la última calibración del color.

En este apartado se visualiza una leyenda que especifica las condiciones que debe cumplir la imagen que se va a capturar y sobre la que se ejecutará el proceso.

La plantilla de calibración (Anexo I) debe colocarse conforme quede dentro de la región de interés definida con el marco del módulo *Configuración cámara*.

Dentro de la región de interés sólo debe visualizarse la plantilla y la cinta.



El proceso de esta calibración empieza al pulsar el botón **Calibrar color**, y al comienzo de la misma, se muestra un mensaje recordando la leyenda anterior.



Pulsar el botón **Aceptar** para continuar, o el botón **Cancelar**, si en realidad no se quiere empezar la calibración.

Una vez finalizada la calibración del color, se visualiza un mensaje que informa de si la calibración se ha realizado correctamente o no.





El proceso actualiza la fecha de la Última calibración y registra los cálculos realizados, que se tendrán en cuenta en el resto de módulos del sistema.

La plantilla de calibración debe imprimirse en un material mate, como vinilo mate, para evitar reflejos.

• Calibración de las distancias – determina la relación entre las distancias reales y las distancias dentro de la imagen, necesaria para el cálculo de la medida de los ejemplares clasificados en el reconocimiento de especies.

Última calibración indica la fecha en la que se registró la última calibración de las distancias.

En los cuadros **Medidas del marco blanco (mm)**, se deben especificar las dimensiones, en mm, del marco blanco que determina la región de interés de la imagen:

- En el **Alto**, se debe introducir la distancia, en mm, entre la línea superior y la línea inferior del marco blanco.
- En el **Ancho**, se debe introducir la distancia, en mm, entre la línea izquierda y la línea derecha del marco blanco.

Las medidas deben realizarse físicamente sobre el fondo de la imagen. Sobre la cinta se utiliza un metro, que a su vez se ve reflejado en la imagen de salida, en la que se puede ver cuánto mide cada distancia.

El proceso de esta calibración empieza al pulsar el botón Calibrar distancias.

Si no se especifican las dimensiones del marco, el sistema informa de la falta de estos parámetros.





Una vez indicadas estas distancias, se ejecuta el proceso de calibración correspondiente y, una vez finalizado, se visualiza un mensaje que informa de si la calibración se ha realizado correctamente o no.



El proceso actualiza la fecha de la Última calibración y registra los cálculos realizados, que se tendrán en cuenta en el módulo *Lances*.

• Calibración del fondo – calcula la caracterización del fondo de las imágenes, la cinta transportadora, para poder extraerlo en el procesado de la generación del catálogo y en el reconocimiento de las especies.

Última calibración indica la fecha en la que se registró la última calibración del fondo.

En este apartado se visualiza una leyenda que especifica las condiciones que debe cumplir la imagen que se va a capturar y sobre la que se ejecutará el proceso.

Antes de empezar la calibración, asegúrese de que dentro del cuadro blanco no hay ningún objeto.

Dentro de la región de interés sólo debe visualizarse la cinta.



El proceso de esta calibración empieza al pulsar el botón **Calibrar fondo**, y al comienzo de la misma, se muestra un mensaje recordando la leyenda anterior.



Pulsar el botón **Aceptar** para continuar, o el botón **Cancelar**, si en realidad no se quiere empezar la calibración.

Una vez finalizada la calibración del fondo, se visualiza un mensaje que informa de si la calibración se ha realizado correctamente o no.



El proceso actualiza la fecha de la Última calibración y registra los cálculos realizados, que se tendrán en cuenta en los módulos *Catálogo* y *Lances*.

Importante - Los diferentes tipos de calibración se deben ejecutar en orden según aparecen en la ventana, primero la calibración del color, después la calibración de las distancias y, por último, la calibración del fondo.

El sistema no cuenta con una cabina cerrada para la captura de imágenes, por lo que puede ser instalado sobre fondos muy diversos, tanto en el color como en la estructura de la cinta transportadora, es por ello que se deben seleccionar los parámetros adecuados para la extracción del fondo de las imágenes capturadas.

Importante - Estos parámetros deben calcularse justo después de realizar la primera calibración del sistema en cada instalación, y después de un posible cambio de cinta transportadora.

Para la elección y modificación de estos parámetros se deben seguir los siguientes pasos:

- 1. Si han cambiado las condiciones del sistema desde la última calibración (iluminación, posición...), calibrarlo según las indicaciones anteriores, primero la calibración del color, después la calibración de las distancias y, por último, la calibración del fondo.
- 2. Capturar unas imágenes desde el módulo *Catálogo* según las indicaciones del apartado siguiente. Para ello se puede simular la captura de imágenes con cualquier objeto en el campo de visión y dentro del marco blanco de la región de interés, como si fuera de una especie.
- 3. Cortar las imágenes capturadas en el punto anterior y pegarlas en el directorio "C:/iSEAS_test/testbg".
- 4. Ejecutar el proceso de pruebas de extracción del fondo pulsando el botón **Prueba extraer fondo** del módulo actual. Al iniciarlo muestra un mensaje que informa cuál es el directorio en el que deben estar las imágenes de la prueba.

iSEAS_
Las imágenes a procesar deben estar en el directorio C:/iSEAS_test/testbg.
Aceptar Cancelar

Pulsar Aceptar para iniciar el proceso, o Cancelar para salir sin ejecutarlo.

5. Cuando finaliza este proceso se muestra una ventana con el mensaje correspondiente, indicando dónde se pueden ver los resultados.



El resultado consta de una serie de carpetas que contienen las imágenes copiadas en el punto 3 a las que se les extrajo el fondo con diferentes combinaciones de los parámetros que se están buscando, y el valor de estos parámetros forma parte del nombre de cada una de estas carpetas.

Los parámetros son cuatro y cada uno de ellos puede tener el valor 0 o 1.

6. Comprobar, en cada una de las carpetas generadas por el proceso del punto anterior, cuál de ellas guarda las imágenes en las que se extrae mejor el fondo.

Una vez localizada esta carpeta, los coeficientes corresponden a los indicados en su nombre.

Por ejemplo - si la carpeta seleccionada es bg1000, esto quiere decir que los coeficientes necesarios son: parámetro_0 = 1, parámetro_1 = 0, parámetro_2 = 0, parámetro_3 = 0.

7. Modificar los parámetros en el fichero correspondiente y según la selección de la carpeta del punto anterior.

El fichero que se debe modificar es **parametersbg.xml** situado en el directorio de configuración del sistema, por defecto, *"C:/iSEAS/settings"*.

Según el ejemplo expuesto en el punto anterior, el fichero debe quedar de la siguiente forma:

		parametersbg.	.xml: Bloc de notas	-		×
Archivo	Edición	Formato Ver	Ayuda			
xml<br <openc <numbe <param <param <param <param <td>version v_stora r_param eter_0> eter_1> eter_2> eter_3> cv_stor</td><td>="1.0"?> ge> eters>41k/parameter 00age></td><td>nber_parameters> `_0> `_1> `_2> `_3></td><td></td><td></td><th>~</th></param </param </param </param </numbe </openc 	version v_stora r_param eter_0> eter_1> eter_2> eter_3> cv_stor	="1.0"?> ge> eters>41k/parameter 00age>	nber_parameters> `_0> `_1> `_2> `_3>			~
<						>
				Línea	1, co	lumr 🔡

Una vez actualizados estos parámetros ya se pueden capturar muestras de especies desde el módulo *Catálogo*, y ejecutar la clasificación desde el módulo *Lances*.

Con el botón Salir de esta ventana, se sale del módulo Calibración del sistema.

Catálogo

En este módulo se genera y gestiona el catálogo de las especies que podrán ser reconocidas.

Se accede a este módulo a través del botón **Catálogo** de la ventana principal.

iseas	Barco INSTALACION_1	Número 1. V			
hanna	tim of the start	Capturar muestras de especies Especie	Car	a	
		Gestionar catálogo de especie Especie Generar patrón Elimina	s Y	Parámetro a Parámetro b	
annor annor		ANF BOC BRF GUN HKE HOM			•
				Sal	ir

A la izquierda de la ventana se ve la imagen de salida, imagen que captura la cámara en tiempo real.

A la derecha de la ventana se accede a los controles para la generación y gestión del catálogo de especies.

Capturar muestras de especies			
Especie	~	Cara 🗸	Capturar muestra
Gestionar catálogo de especies Especie	*	Parámetro a	
Generar patrón Eliminar pat	rón	Parámetro b	
Especies catalogadas			
ANF BOC BRF			^
HKE HOM			~

• Capturar muestras de especies

Para generar el catálogo de especies, se debe empezar con la recopilación de imágenes de las especies que se quieren reconocer.

Para ello se capturan imágenes de muestras de especies de la siguiente manera:

1. En el cuadro **Especie**, se identifica la especie de la que se van a capturar imágenes.

El cuadro es un desplegable con las especies de las que ya se tiene alguna imagen.

Si se quiere añadir más imágenes a una especie ya existente, seleccionamos en este cuadro la especie necesaria.

Si se quiere capturar imágenes de una especie nueva, se teclea su nombre en este mismo cuadro.

Сар	turar muestras de especies		
	Especie	Cara	Coopt your musethe
Ges	I ANF BOC BRF GUN HKE HOM MAC MEG RJC RJN SYC WHB	Talla media (mm) Peso medio (gr)	

2. En el control **Cara**, se especifica a qué cara de la especie corresponde la imagen que se va a capturar.

Se diferencia entre una cara y otra para caracterizarlas por separado, ya que hay especies que difieren mucho en color y/o textura entre una cara y otra.

Se puede diferenciar entre las caras Ventral, Lateral, y Dorsal.

3. Se pulsa el botón Capturar muestra para capturar la imagen de la muestra deseada.

En el caso de que no se haya especificado la especie y/o la cara a la que corresponde la muestra, el sistema lanza el siguiente mensaje de advertencia.



Una vez que se especifique la especie y la cara, si se especifica una especie nueva, el sistema lanza otro mensaje para informar al usuario de que se va a crear una especie nueva.

iSEAS_	×
¿Quiere crear esta nueva especie?	
Sí No	

Al responder afirmativamente al mensaje anterior, o en el caso de especificar una especie ya existente, el sistema pide que el usuario se asegure de que el ejemplar esté bien colocado, completamente dentro del cuadro que limita la región de interés.



Correcto



Incorrecto



Si el ejemplar no está bien colocado, se debe responder **No** al mensaje anterior, colocar bien el ejemplar y volver a pulsar el botón **Capturar muestra**.

Si el ejemplar ya está bien colocado, y realmente se quiere capturar esa muestra, se debe responder **Sí** al mensaje anterior.

Una vez registrada la imagen de muestra, ésta se vincula a la especie y cara seleccionadas, y se muestra un mensaje que indica que la muestra fue capturada.



Pulsar el botón Aceptar para continuar con la gestión del catálogo de especies.

De esta forma se van capturando imágenes de muestra de todas las especies que se quieren incluir en el catálogo.

• Gestionar catálogo de especies

Una vez que se tienen las muestras necesarias de alguna especie, ya se puede generar el patrón que la caracteriza, para ello seguimos los siguientes pasos:

1. En los cuadros **Especie**, **Parámetro a** y **Parámetro b**, se especifican, respectivamente, la especie de la que se quiere generar el patrón y los valores de a y b utilizados para la estimación del peso de los ejemplares que se identifiquen en el módulo *Lances*.

El cuadro *Especie* es un desplegable con las especies de las que se tiene alguna muestra.

Los parámetros a y b son los utilizados en la fórmula W = a * L^b, con la que se estima el peso de un ejemplar a partir de su talla.

Importante - Los parámetros a y b deben corresponder a tablas que trabajen con las unidades cm y g, es decir, que partiendo de la longitud en cm, nos den el peso en g, como las tablas NAFO.

2. Pulsar el botón Generar patrón, para generar el patrón de la especie especificada.

Si no se selecciona ninguna especie, el sistema advierte de que debe seleccionarse.



Si no se especifican los parámetros a y b, el sistema también advierte de que deben especificarse.



Si se identifica bien la especie y se especifican los parámetros a y b, el sistema pasa a calcular el patrón correspondiente.

Si el patrón ya estuviese generado, el sistema muestra el siguiente mensaje de aviso.



Se debe responder al mensaje indicando si realmente se quiere volver a calcular un patrón ya existente.

Importante - Si se modifican las muestras de una especie que ya tiene su patrón generado, se añaden o eliminan muestras, se asignan a una cara diferente..., se debe volver a generar ese patrón para que pueda incluir las modificaciones realizadas.

Si durante el proceso de generación del patrón, el sistema detecta que no hay imágenes de la especie indicada, muestra el mensaje correspondiente.



No es necesario que todas las especies tengan muestras de todas las caras. Para las especies que no presentan diferencias entre sus caras se pueden capturar muestras y asignarlas todas a una cara.

Una vez generado el patrón, el sistema indica la finalización de este proceso, y si se trata de una nueva especie, la incluye en la lista **Especies catalogadas**.

Si se quiere eliminar una especie del catálogo, se selecciona la especie en el cuadro **Especie**, y se pulsa el botón **Eliminar patrón**.

Para realizar la eliminación, el sistema muestra un mensaje que advierte al usuario de que se va a eliminar una especie, **tanto su patrón como sus imágenes**.



Importante – Si se quiere eliminar una especie porque ésta no se quiere tener en cuenta en el reconocimiento, no hace falta eliminarla, simplemente se debe configurar, en el módulo *Campañas*, una campaña que no incluya esa especie y marcarla como *campaña activa*, las especies excluidas no entran en el reconocimiento.

Si se responde **Sí** al mensaje anterior, el sistema realiza la eliminación, elimina la especie también de la lista **Especies catalogadas**, y muestra un mensaje cuando acaba.



Importante – No se debe eliminar una especie en la misma sesión en la que se creó.

Si se crea una especie nueva con la captura de muestras, para poder eliminarla correctamente, el usuario debe salir del sistema, tanto del módulo *Catálogo* como de la ventana principal, y volver a lanzar la aplicación para eliminar la nueva especie creada.

Con el botón Salir de esta ventana, se sale del módulo Catálogo.

Configuración campaña

En este módulo se generan y gestionan las campañas con las especies a reconocer en cada una de ellas, y se puede establecer qué campaña estará activa en la ejecución de los lances.

Se accede a este módulo a través del botón **Configuración campañas** de la ventana principal.

	iSEAS	- Configuración de las campañas	- • ×
iseas	Barco INSTALACION_1	Número 1 v	Abe
	Campaña activa: Campaña	PORTOSANTO	
	Especies excluidas ANF BOC BRF GUIN HKE HOM MAC MEG RJC RJN SYC VVHB	>> <<	Guardar campaña Eliminar campaña Activar campaña
			Salir

En este módulo se pueden crear campañas nuevas y modificar o eliminar campañas existentes.

En el cuadro **Campaña** se especifica la campaña que se quiere crear, modificar o eliminar. En el caso de crear una campaña nueva se teclea su nombre en este cuadro.

En el caso de modificar o eliminar una campaña existente, se selecciona la campaña en el desplegable de este cuadro.

Para asignar las especies que se quieren identificar en la campaña indicada, se deben seleccionar una a una de la lista **Especies excluidas**, y pasarlas a la lista **Especies a reconocer** ______ con el botón

Para excluir >> especies a reconocer en la campaña indicada, se deben seleccionar una a una de la lista **Especies a reconocer**, y pasarlas a la lista **Especies excluidas** con el botón

<<

Las especies visualizadas en las listas, son especies de las que se tiene un patrón, generado a través del módulo *Catálogo*.

Pulsar el botón **Guardar campaña** para guardar la configuración de la campaña nueva o la modificación de una ya existente.

Una vez guardada, el sistema lanza un mensaje indicando que se ha guardado correctamente.



Si se quiere guardar una campaña sin identificarla en el cuadro *Campaña*, el sistema advierte al usuario con el siguiente mensaje.

iSEAS_
Debe especificar la campaña que quiere modificar.
Aceptar

Para eliminar una campaña, se debe seleccionar una campaña existente en el cuadro desplegable **Campaña**, y pulsar el botón **Eliminar campaña**.

Antes de eliminarla, el sistema muestra un mensaje para que el usuario confirme que realmente quiere eliminarla.



Pulsar el botón **Aceptar** del mensaje para eliminarla, o el botón **Cancelar** para cancelar la eliminación.

El sistema muestra un mensaje cuando ya se elimina la campaña.



Si no se identifica la campaña, el mensaje muestra un mensaje de advertencia.

iSEAS_
Debe especificar la campaña que quiere eliminar.
Aceptar

Importante – los patrones pueden ser eliminados del sistema por el administrador si se estima necesario, bien por espacio, bien porque no interesa guardar sus muestras o patrón en una determinada instalación.

Si antes de eliminar una especie, se generó una campaña que la incluía, esta especie sigue perteneciendo a la campaña, lo único que ocurre es que no estará en la lista de especies a reconocer en el módulo *Lances* por no estar disponible su patrón.

Campaña activa identifica la campaña que está activa en cada momento.

Para activar una campaña, se debe seleccionar una campaña existente en el cuadro desplegable **Campaña** o escribir el nombre de la campaña en ese mismo cuadro y pulsar el botón **Activar campaña**.

Una vez activada la campaña se muestra el mensaje siguiente.



La campaña activa es la que determina qué especies se van a reconocer en el procesado de clasificación durante la ejecución de un lance.

El sistema puede funcionar sin una *campaña activa* o con una campaña activa que no tiene especies asignadas.

Importante - Cuando se activa una campaña, se pone a cero el número del lance, que indica el número de lances ejecutados desde el módulo *Lances* desde que se activó la campaña.

Con el botón Salir de esta ventana, se sale del módulo Campañas.

Lances

En este módulo se puede iniciar, parar, reanudar y finalizar el reconocimiento de especies.

Se accede a este módulo a través del botón Lances de la ventana principal.

		iSEAS - Lances	- 0
iseas	Barco INSTALACION_1	Número 1 v	Life :
		Empezar lance	
	image	Finalizar lance	
		Parar lance Rean Número de imágenes capturadas por minuto	udar lance
Administración		Sa	lir

A la izquierda de la ventana se ve el marco de la imagen de salida, que mostrará la imagen que captura la cámara en tiempo real durante la ejecución de un lance.

A la derecha de la ventana se accede a los controles para manejar el proceso de reconocimiento.

La campaña activa, seleccionada en el módulo *Campañas*, determina las especies que deben ser reconocidas, el resto de ejemplares se agrupará en una clase denominada **Otros**.

Si se quiere guardar un muestreo de las imágenes procesadas, se debe marcar la casilla Guardar una de cada diez imágenes.

El reconocimiento puede funcionar en dos modalidades, con disparador, de forma que se captura una imagen cada vez que un imán, de los instalados en la cinta transportadora, pasa pasa delante del detector, y en modo continuo, de forma que por minuto se captura el número de imágenes especificado en el cuadro **Número de imágenes capturadas en un minuto**.

Si el *Número de imágenes capturadas en un minuto* está vacío o es igual a 0, el reconocimiento funciona en modo *con disparador*.

Se debe pulsar el botón **Empezar lance** para comenzar el reconocimiento de especies.

Si no hay una campaña activa, o ésta no tiene especies asignadas, el sistema avisa que se va a ejecutar un lance sin especies a clasificar.

iSEAS_				
La campaña activa no tiene especies asignadas.				
Aceptar Cancelar				

Con los botones **Parar lance**, **Reanudar lance** y **Finalizar lance**, se puede parar, reanudar o finalizar el lance respectivamente.

El proceso de reconocimiento se ejecuta mediante dos subprocesos, por una parte se registran las imágenes capturadas, y por otra parte, se procesan las imágenes registradas.

Cuando se pulsa la finalización del lance, el sistema muestra un mensaje que informa al usuario de que debe esperar a que finalice el procesado de las imágenes registradas antes de salir de la aplicación.



Una vez finalizado el procesado de las imágenes registradas, se muestra el mensaje correspondiente.



El proceso de reconocimiento genera un fichero de resultados según el formato especificado en el **Anexo II**.

Una vez finalizado el procesamiento, se envían los resultados al directorio configurado según se indica en el módulo siguiente *Enviar Resultados*.

En la parte inferior de la ventana hay un botón **Administración**, que da acceso a la ventana principal del sistema, desde la que se accede a los módulos, *Configuración cámara*, *Calibración del sistema*, *Catálogo*, *Configuración campaña*, *Enviar resultados*, y a este mismo módulo *Lances*.

Con el botón **Salir** de esta ventana, se sale del módulo *Lances*.

Enviar resultados

Envía los resultados, al directorio especificado en el fichero de configuración correspondiente, por defecto **redbox.xml** situado en el directorio de configuración del sistema, por defecto, ".

```
×
redbox.xml: Bloc de notas
Archivo Edición Formato Ver Ayuda
<?xml version="1.0"?>
                                                     ^
<opencv_storage>
<!-- ruta para un equipo de red:
        <path>\EQUIPO\carpeta</path>
        <unit>M:</unit>
        <user>usuario</user>
        <password>clave</password>
-->
<!-- ruta para un directorio en el mismo equipo:
        <path>C:/carpeta</path>
        <unit></unit>
        <user>usuario</user>
        <password>clave</password>
-->
<unit></unit>
<path>C:/redbox</path>
<user></user>
<password></password>
```

En el mismo fichero se especifica la forma de configurar el envío a un equipo de red o a un directorio local.

Al pulsar **Enviar resultados** se envían, al directorio indicado, todos los ficheros que resultaron de la ejecución de los lances y que no fueron enviados anteriormente.

Este proceso también se ejecuta al finalizar el procesado de las imágenes de un lance, para enviar el último fichero generado.

En el caso de que se produjeran fallos en la conexión con otro equipo de red, siempre se puede configurar el fichero *redbox.xml* para que se envíen los resultados a un directorio del mismo equipo en el que está instalado el sistema, ejecutar el envío de resultados, y llevar los resultados directamente al equipo del *RedBox* con una memoria USB.

En este último caso, para no enviar los ficheros más de una vez, se debe vaciar el directorio intermedio antes de la ejecución de cada envío.

Estructura del directorio de la aplicación

En este apartado se explica la estructura del directorio de la aplicación, que incluye los diferentes ficheros de configuración que utiliza el sistema, el lugar en el que se mantienen el catálogo de especies y las campañas, y el lugar en el que se vuelcan los resultados de los diferentes procesos de reconocimiento de especies ejecutados (lances).

El **directorio principal** de la aplicación es, por defecto, **C:/iSEAS**, que contiene los siguientes subdirectorios:

- app contiene el ejecutable de la aplicación, los logotipos e imágenes utilizadas, las librerías externas necesarias y el fichero de configuración principal, settingsapp.xml.
- **settings** contiene los demás ficheros de configuración utilizados por los diferentes módulos del sistema.
- log contiene los ficheros de seguimiento de la ejecución de la aplicación con fines de depuración y mantenimiento. En estos ficheros se registran los posibles errores o mensajes indicativos de comportamientos no habituales que pueden generar los diferentes módulos del sistema.
- **calibration** contiene las imágenes capturadas y utilizadas en la última calibración ejecutada de cada tipo, registradas en el módulo *Calibración del sistema*.
- **species** contiene las imágenes de muestra de cada una de las especies catalogadas y los ficheros con los patrones de estas especies, generados en el módulo *Catálogo*.
- **campaigns** contiene los ficheros de configuración de cada una de las campañas generadas en el módulo *Campañas*.

En los siguientes párrafos se desglosan estos subdirectorios.

<u>Subdirectorio</u> **app** – los principales ficheros que contiene son:

- EL ejecutable de la aplicación, **iSEAS_.exe**.
- **settingsapp.xml** contiene las rutas a los directorios y ficheros de configuración utilizados en los diferentes módulos del sistema.

🧧 settingsapp.xml: Bloc de notas – 🗖	x
Archivo Edición Formato Ver Ayuda	
<pre>k?xml version="1.0"?> <opencv_storage> <appfileinstallation>C:/iSEAS/settings/installation.xml</appfileinstallation> <applogfile>C:/iSEAS/log/logapp.txt</applogfile> <appdirspecies>C:/iSEAS/species/</appdirspecies> <appdircampaigns>C:/iSEAS/campaigns/</appdircampaigns><td>^</td></opencv_storage></pre>	^
<cameralogfile>C:/iSEAS/log/logcamera.txt</cameralogfile> <camerasettings>C:/iSEAS/settings/settings.txt</camerasettings> <camerasettingsinit>C:/iSEAS/settings/settingsinit.txt</camerasettingsinit>	
<processlogfile>C:/iSEAS/log/logprocessing.txt</processlogfile> <processroifile>C:/iSEAS/settings/imageroi.xml</processroifile> <processcalibrationfile>C:/iSEAS/settings/calibration.xml</processcalibrationfile> <processdircalibration>C:/iSEAS/calibration/</processdircalibration> <processparametersbg>C:/iSEAS/settings/parametersbg.xml</processparametersbg> <processdirresult>C:/iSEAS_resultado/</processdirresult>	
<redboxfile>C:/iSEAS/settings/redbox.xml</redboxfile>	
<filecalibration>C:/iSEAS/settings/calibration.xml</filecalibration> 	~
< >) .

Si se desea una estructura diferente de los directorios del sistema, se deben modificar las entradas correspondientes de este XML, y trasladar su contenido a las nuevas rutas establecidas.

Subdirectorio settings – contiene los siguientes ficheros de configuración.


- **activecampaign.xml** contiene el nombre de la campaña que está activa en un momento determinado, y el número de lances ya ejecutados desde que se activó.
- **calibration.xml** contiene los resultados y las fechas de las últimas calibraciones realizadas desde el módulo *Calibración del sistema*.
- imageroi.xml contiene los parámetros del marco que determina la región de interés dentro de las imágenes capturadas por la cámara. Marco que se define en el módulo *Configuración cámara*.
- **installation.xml** contiene el nombre y el número de la instalación, que se configuran desde la *Ventana principal*.
- parametersbg.xml contiene unos coeficientes necesarios en el proceso de eliminación del fondo de las imágenes, utilizado tanto en la generación de patrones como en el proceso de clasificación. Según la instalación, y las indicaciones del módulo *Calibración del sistema*, es posible que sea necesario variar estos coeficientes.
- redbox.xml contiene la ruta y las credenciales necesarias para enviar los resultados de los reconocimientos al directorio de red del sistema RedBox indicado. Estos resultados se envían al finalizar el procesado de imágenes capturadas en un lance, y también desde la opción *Enviar resultados* de la *Ventana principal*.
- **settings.txt** contiene los parámetros con los que se debe iniciar la cámara, alguno de ellos modificables desde el módulo *Configuración cámara*.
- **settingsini.txt** contiene los parámetros de fábrica de la cámara para este sistema, configuración a la que se puede volver desde el módulo *Configuración cámara*.
- test.txt contiene los parámetros necesarios para ejecutar ciertas funcionalidades del sistema en modo testeo, sin cámara.

<u>Subdirectorio</u> **log** – contiene los siguientes ficheros con el historial de mensajes de los diferentes módulos del sistema:

- **logapp.xml** contiene mensajes de comportamientos no habituales en la ejecución de la lógica del sistema.
- **logcamera.xml** contiene mensajes de comportamientos no habituales respecto a la comunicación y manejo de la cámara.
- **logprocessing.xml** contiene mensajes de comportamientos no habituales en el procesado de las imágenes, tanto en los procesos de calibración, como en los de generación de patrones y clasificación de especies.

<u>Subdirectorio</u> **calibration** – contiene las imágenes capturadas y utilizadas para el cálculo de las últimas calibraciones registradas en el módulo *Calibración del sistema*.

La imagen **image_color** corresponde a la imagen utilizada en la calibración del color, de los niveles blanco y negro.

La imagen **image_dist** corresponde a la imagen utilizada en la calibración de las distancias, cálculo de la relación entre las distancias reales y las distancias dentro de la imagen.

Las imágenes **image_bg** corresponden a las imágenes utilizadas para caracterizar el fondo, la cinta transportadora, que se extrae de las imágenes a procesar en los módulos *Catálogo* y *Lances*.

Sólo se guarda la parte de la imagen que queda enmarcada dentro de la última región de interés configurada en el módulo *Configuración cámara*, que es la porción de imagen procesada.

<u>Subdirectorio</u> **species** – para cada una de las especies registradas en el módulo *Catálogo* contiene:

- Un directorio con las imágenes de muestra capturadas para esa especie. Estas muestras se organizan en tres subdirectorios, **Dorsal**, **Lateral** y **Ventral**, que contienen las muestras capturadas para cada una de estas caras.

Sólo existirán los subdirectorios para los que se han capturado muestras, no es necesario que todas las especies tengan muestras de todas las caras.

Para la caracterización de una determinada especie, se clasifican las muestras en más de una cara cuando entre éstas hay diferencias significativas de color y/o textura.

Ejemplo de la estructura de carpetas para tres especies:



En el ejemplo se muestra que para la especie *BRF* sólo se tienen muestras de la cara lateral, para la especie *GUN* se tienen muestras para todas las caras, y para la especie *HKE* sólo se tienen muestras para dos caras.

- Un fichero *.xml* con los patrones que caracterizan cada una de las caras de la especie. Este fichero existe si se generó el patrón de la especie correspondiente desde el módulo *Catálogo*.

Siguiendo con el ejemplo anterior, podemos tener la siguiente situación, tener generados los patrones para las especies *BRF*, y *GUN*, pero no para la especie *HKE*:



Cada imagen de muestra consta de dos imágenes concatenadas, la imagen capturada y la máscara que determina la localización de la muestra dentro de la primera imagen, de esta forma se puede compartir el catálogo de especies entre instalaciones diferentes.

En una instalación se pueden generar los patrones de muchas especies, que pueden ser utilizados en otras instalaciones simplemente con copiar los ficheros .xml correspondientes.

Si en otra instalación interesa poder modificar o ampliar las muestras de una determinada especie, también será necesario copiar la carpeta de muestras correspondiente, de esa forma, en la otra instalación se puede volver a generar el patrón para que incluya las modificaciones realizadas.

<u>Subdirectorio</u> **campaigns** – contiene un fichero *.xml* por cada una de las campañas creadas en el módulo *Campañas*.

Los ficheros contienen la enumeración de especies a tener en cuenta en el proceso de reconocimiento del módulo *Lances* si esa campaña está activa.

Importante - Si antes de eliminar una especie, se generó una campaña que la incluía, esta especie sigue perteneciendo a la campaña, lo único que ocurre es que no estará en la lista de especies a reconocer en el módulo *Lances* por no estar disponible su patrón.

Modos de funcionamiento

Modo testeo

La aplicación puede funcionar sin cámara conectada para poder testear determinadas funciones del procesado de imágenes.

Para poder funcionar de este modo, se debe dar el valor 0 al parámetro **camera** dentro del fichero **test.xml**.



El funcionamiento de los diferentes módulos del sistema trabajando en modo testeo es el siguiente:

- **Configuración de la cámara** este módulo no funciona en modo testeo.
- Calibración del sistema

Permite calibrar el sistema con imágenes copiadas en la carpeta.

Para simular la situación de una determinada instalación, se pueden copiar, en la carpeta anterior, las imágenes generadas en la ejecución de su última calibración, y se ejecutan las diferentes calibraciones de la forma habitual.

Al principio de la ejecución de cada calibración, el sistema muestra los mismos mensajes que con el modo con cámara, aunque en este caso, en lugar de comprobar las condiciones en el cuadro de la imagen de salida, se comprueban en las imágenes copiadas.

El botón *Prueba extraer fondo* funciona de la misma forma que en la ejecución con cámara.

- **Catálogo** la opción *Capturar muestra* no funciona en modo testeo, y el resto de las opciones funcionan del mismo modo que con la ejecución con cámara, ya que no capturan imágenes en sus procesos.
- **Configuración campaña** este módulo no utiliza la cámara, de modo que funciona de la misma forma que en la ejecución normal de la aplicación.
- Lances

En modo testeo sólo funciona el botón **Empezar lance**, los demás botones y campos no influyen en la ejecución del lance.

Con este botón, *Empezar lance*, se ejecuta el procesado de clasificación de las imágenes situadas en la carpeta <u>C:/iSEAS_test/samples</u>, y los resultados de esta ejecución se vuelcan en la carpeta <u>C:/iSEAS_test/results</u>.

El resultado de la ejecución del lance en modo testeo consta del fichero de resultados, al igual que en la ejecución en modo normal, y de una *imagen resultado* por cada imagen procesada.

Esta *imagen resultado*, consta de cuatro imágenes concatenadas, la imagen original, la imagen original sin el fondo, la representación de la segmentación, y la representación de la clasificación.

En la representación de la clasificación se utilizan colores aleatorios para dibujar las diferentes especies reconocidas, de forma que peces de la misma especie estarán representados con el mismo color. Esta imagen también visualiza las líneas utilizadas para el cálculo del tamaño de cada individuo reconocido.

• **Enviar resultados** – este módulo no utiliza la cámara, de modo que funciona de la misma forma que en la ejecución normal de la aplicación.

Parámetros que modifican el modo de funcionamiento

En el fichero **test.xml**, también se guardan los valores de otros parámetros que influyen en el modo de funcionamiento, tanto de la clasificación de especies, como de posibles puntos intermedios que sirven para el testeo del programa.

```
×
                                test.xml: Bloc de notas
Archivo Edición Formato Ver Ayuda
<?xml version="1.0"?>
                                                                                  ~
<opencv storage>
<!-- trabajar con o sin cámara:
        camera>0</camera>
                               NO CAMERA
                                YES CAMERA
        <camera>1</camera>
-->
<camera>0</camera>
<minpixelsobject>30000</minpixelsobject>
<!-- trabajar con objetos solapados o separados:
        <overlapping>1</overlapping> SOLAPADOS
        <overlapping>0</overlapping>
                                        NO SOLAPADOS
-->
<overlapping>0</overlapping>
<!-- guardar muestras de las imágenes capturadas en un lance:
        <savesampleimages>-1</savesampleimages> GUARDA TODAS LAS IMAGENES
        <savesampleimages>0</savesampleimages> NO GUARDA NINGUNA IMAGEN
        <savesampleimages>10</savesampleimages> GUARDA UNA DE CADA 10 IMAGENES
        <savesampleimages>45</savesampleimages> GUARDA UNA DE CADA 45 IMAGENES
-->
<savesampleimages>0</savesampleimages>
</opencv_storage>
```

Estos parámetros son los siguientes:

- **minpixelsobject** indica el área mínima, en píxeles, a tener en cuenta en los objetos detectados en la imagen. Los ejemplares que tienen un área más pequeña no se tienen en cuenta.
- **overlapping** indica si el procesado de clasificación debe suponer que los objetos pueden llegar o no solapados.
- savesampleimages indica si se deben guardar muestras de las capturas de imágenes durante el procesado de un lance. Se guardará una imagen de cada grupo de imágenes consecutivas indicadas en este parámetro. Si se indica un valor negativo, se guardarán todas las imágenes procesadas.

Instalación

El sistema costa de una caja metálica que contiene todo el hardware necesario para la captura y el procesado de las imágenes, y de dos módulos externos para sujetar la iluminación.

Esta estructura debe posicionarse, lo más perpendicular y centrada posible, sobre el punto de la cinta transportadora en el que se van a capturar las imágenes.



Hardware

Todo el hardware está dentro de una caja de acero inoxidable, sellada con un nivel de protección IP65, y un lado de policarbonato transparente a través del que la cámara captura las imágenes.

Los componentes hardware que contiene son:

- Una cámara digital, que captura las imágenes de los ejemplares a clasificar a su paso por la cinta transportadora.
- Barras de luz para la iluminación del área de captura de imágenes.
- Un PC, en el que se reciben y se procesan las imágenes capturadas por la cámara digital.
- Fuentes de alimentación necesarias para la conexión de las luces y la cámara.
- Cableado necesario para la conexión entre los diferentes componentes y el exterior, puerto USB, puerto Ethernet, cable de alimentación del conjunto y cable de conexión con un disparador que detecta el avance de la cinta transportadora.

Este hardware debe cumplir los siguientes requisitos mínimos:

Cámara digital

Espectro visible Matricial - Color Interface USB 3.0 o Gigabit Ethernet Resolución 5Mp (2560px x 2048px) Imágenes por segundo: 22fps a máxima resolución

PC

Sistema operativo y procesador de 64 bits Tarjeta Gigabit Ethernet Puerto USB 3.0 Sistema operativo Windows vista / 7 / 8 / 10

La iluminación y la óptica dependerán de la ubicación de la instalación, distancia a la cinta transportadora, tamaño de los objetos a reconocer...

El sistema fue testeado con éxito con el siguiente hardware:

<u>Cámaras</u>: Jai Ethernet – GO-5000C-PGE Jai USB 3.0 - GO-5000C-USB Basler Ethernet – acA2040-25gc

<u>PCs</u>

Procesadores: Intel® Atom™ CPU E3845 @ 1,91GHz Intel® Celeron ® CPU J1900 @ 1,99GHz Intel® Core™ i7-4510U CPU @ 2,00GHz 2,60GHz

Memoria RAM: 8GB

Sistemas operativos:

Windows 7 Professional - Service Pack 1 Windows 8.1 Windows 10 Professional

Software

El software consta de una aplicación Windows de escritorio, y el entregable contiene los siguientes ficheros y directorios:

iSEAS
 iSEAS_resultado
 iSEAS_test
 JaiControlTool
 iSEAS_Manual_.pdf
 JAI SDK Version 3.0.1.4_x64 .exe
 vc_redist.x64.exe
 vc_redist.x86.exe

Para la instalación del sistema se deben seguir los siguientes pasos:

1. Copiar las siguientes carpetas en .



- 2. Ejecutar el archivo **vc_redist.x64.exe** para la instalación de los componentes de Windows necesarios durante la ejecución de la aplicación.
- 3. Ejecutar el archivo **vc_redist.x86.exe** para la instalación de los componentes de Windows necesarios durante la ejecución de la aplicación.
- 4. Ejecutar el archivo **JAI SDK Version 3.0.1.4_x64** para la instalación de las librerías de conexión con la cámara, y para la instalación de la aplicación *JAI Control Tool,* aplicación para el manejo de la cámara fuera del sistema iSEAS.

Por si se quiere manejar la cámara para otros fines, en la carpeta *JaiControlTool* del entregable, también se incluye el manual para la instalación y el manejo de la aplicación *Jai Control Tool*. Desde esta aplicación se puede manejar cualquier cámara Gigabit Ethernet, y cámaras Jai USB 3.0.

En la instalación indicar que se instala también el JAI GigE Vision Filter Driver, y JAI USB3 Vision Driver, como indican las imágenes siguientes.

JAI SDK Version 3.0.0.0 - InstallShield W	fizard		×
Options to be installed			Æ
Please select the desired install option be	low		
		See the j	possibilities
The JAI Filter Driver is required for efficient cameras.	streaming of GigE	Vision image packe	ts from JAI
Install JAI GigE Vision Filter Driver			
The JAI SDK has some built-in debug featur environment variable JAI_SDK_DEBUG. The breakpoints while a camera is connected an Heartbeat Timeout interval if necessary.	es that can be en se debug feature d streaming image	abled/disabled using s makes it easier to is. This is done by e	g the use extending the
Enable built-in debug features			
istallShield			
	< Back	Next >	Cancel
			-
JAI SDK Version 3.0.0.0 - InstaliShield W	lizard		
Options to be installed			F A
Please select the desired install option be	low	020000	
		See the	possibilities
The JAI USB3 Vision Driver is required for st cameras.	treaming of USB3	Vision images from	USB3 Vision
V Install JAI USB3 Vision Driver			
- Mar - 12			
talishield			4
	< Back	Next >	Cancel

Si se va a utilizar el sistema con una cámara Gigabit Ethernet, hay que optimizar el funcionamiento de la tarjeta de red, según especifica el apartado 4.-Optimizing your computer for using the JAI SDK & Contro Tool del manual de JAI Control Tool, fichero JAI SDK Version 3.0.1.4_Getting-Started-Guide.pdf, que incluye el entregable.

Para iniciar el programa, ejecutar el fichero.

Para poder iniciarlo desde el escritorio, se debe crear un acceso directo a este ejecutable.

Para manejar el programa, seguir las indicaciones del apartado *Manual de usuario* de este documento.

Los resultados que genera el programa, incluyen una etiqueta temporal que se genera con la hora del sistema operativo, para seleccionar un huso horario diferente, debe modificarse en el ajuste de fecha y hora del equipo.

Para modificar el huso horario se deben seguir los siguientes pasos:

1. Pulsar el botón derecho del ratón sobre la hora del equipo, por defecto en la esquina inferior derecha de la pantalla, y seleccionar la opción **Ajustar fecha y hora**.

	Barras de herramientas	
	Ajustar fecha y hora	1
	Personalizar iconos de notificación	
	Ventanas en cascada	
	Mostrar ventanas apiladas	
	Mostrar ventanas en paralelo	
	Mostrar el escritorio	
1	Administrador de tareas	
-	Bloquear la barra de tareas	
	Propiedades	
	▲ ¶III ♥III ESP 17/05/2	2017

2. En la ventana de ajuste, seleccionar la opción Cambiar zona horaria.

ř.		Fecha y hora		×
Fecha y hora	Relojes adicionales	Hora de Internet		
Zona hora	Fe do Ha 8:5	cha: omingo, 12 de marz ora: 58:40	to de 2017 Cambiar fecha	y hora
(UTC+01:0	0) Bruselas, Copenł	nague, Madrid, Par	a Sambiar zona ho	oraria
El horario reloj está c	de verano se inicia e onfigurado para ad	el domingo, 26 de i lelantarse 1 hora a	marzo de 2017 a esa hora.	las 2:00. El
✓ Notifica	arme cuando camb	ie el reloj		
		Aceptar	Cancelar	Aplicar

3. En la ventana de configuración, en el desplegable **Zona horaria**, establecer el huso horario requerido.

af in the second	Configuración de zona horaria	×
Establezca la zona	a horaria:	
Zona horaria:		
(UTC+01:00) Bru	iselas, Copenhague, Madrid, París	~
✓ Ajustar el reloj	j automáticamente al horario de verano	
Fecha y hora actu	uales: domingo, 12 de marzo de 2017, 9:06	
	Aceptar Cancelar	

(UTC-03:30) Terranova	^
Vé (UTC-03:00) Araguaína	
(UTC-03:00) Brasilia	
(UTC-03:00) Cayena, Fortaleza	n -
(UTC-03:00) Ciudad de Buenos Aires	ľ
UTC-03:00) Groenlandia	
UTC-03:00) Montevideo	•
UTC-03:00) Punta Arenas	1
(UTC-03:00) Salvador	12
(UTC-03:00) San Pedro y Miquelón	
o (UTC-02:00) Hora universal coordinada-02	
(UTC-01:00) Azores	
ter (UTC-01:00) Cabo Verde	ia, i
(UTC) Hora universal coordinada	
(UTC+00:00) Casablanca	
(UTC+00:00) Dublín, Edimburgo, Lisboa, Londres	
(UTC+00:00) Monrovia, Reikiavik	
de (UTC+01:00) África Central Occidental	
(UTC+01:00) Amsterdam, Berlín, Berna, Roma, Estocolmo, Viena	
(UTC+01:00) Belgrado, Bratislava, Budapest, Liubliana, Praga	
(UTC+01:00) Bruselas, Copenhague, Madrid, París	
(UTC+01:00) Sarajevo, Skopie, Varsovia, Zagreb	
(UTC+01:00) Windhoek	
(UTC+02:00) Ammán	
🔜 (UTC+02:00) Atenas, Bucarest	<
UTC+02:00) Beirut	
(UTC+02:00) Chisinau	
(UTC+02:00) Damasco	
(UTC+02:00) El Cairo	
(UTC+02:00) Gaza, Hebrón	✓
(UTC+01:00) Bruselas, Copenhague, Madrid, París	~
Ajustar el reloj automáticamente al horario de verano	
,	
Fecha y hora actuales: domingo, 12 de marzo de 2017, 9:07	
Aceptar Cance	lar

Desarrollo

En este punto se explica la estructura de ficheros que conforma el código de la aplicación.

La aplicación está desarrollada en C++, en el IDE *Microsoft Visual Studio Enterprise 2015* [MSVS2015], versión 14.0.24720.00 Update 1.

En el desarrollo se utilizan las siguientes librerías externas:

- Jai_Factory para la conexión y el manejo de la cámara [JAISDK].
- **OpenCV** para el procesado de imágenes y el manejo de los ficheros XML [OPENCV].

Estructura

En el desarrollo de la lógica de la interfaz de usuario, se mantiene la estructura propia de los diálogos de la *Microsoft Foundation Class Library* [MFC].

Los diálogos desarrollados se implementan en las siguientes clases:

- **CiSEAS_Dlg** - desarrollada en los ficheros iSEAS_Dlg.*h* e iSEAS_Dlg.*cpp*.

Recoge el funcionamiento de la ventana principal de la aplicación. Contiene las llamadas a las acciones y el mecanismo de mensajes correspondientes a la configuración de la instalación y a la llamada de los demás módulos del sistema.

- **CDlg_Camera** - desarrollada en los ficheros *Dlg_Camera.h* y *Dlg_Camera.cpp*.

Recoge el funcionamiento del modulo *Configuración cámara*. Contiene las llamadas a las acciones y el mecanismo de mensajes correspondientes al manejo de los parámetros de la cámara y de la delimitación de la región de interés en las imágenes procesadas en los demás módulos del sistema.

- **CDlg_Calibration** - desarrollada en los ficheros *Dlg_Calibration.h* y *Dlg_Calibration* .cpp.

Recoge el funcionamiento del modulo *Calibración del sistema*. Contiene las llamadas a las acciones y el mecanismo de mensajes correspondientes a los procesos de calibración del sistema.

- **CDlg_Samples** - desarrollada en los ficheros *Dlg_Samples.h* y *Dlg_Samples .cpp*.

Recoge el funcionamiento del modulo *Catálogo*. Contiene las llamadas a las acciones y el mecanismo de mensajes correspondientes a la gestión de muestras y patrones del catálogo de especies. - **CDlg_Campaign** - desarrollada en los ficheros *Dlg_Campaign.h* y *Dlg_Campaign .cpp*.

Recoge el funcionamiento del modulo *Campañas*. Contiene las llamadas a las acciones y el mecanismo de mensajes correspondientes a la gestión de las campañas de reconocimiento.

- **CDIg_Lances** - desarrollada en los ficheros *DIg_Lances.h* y *DIg_Lances .cpp*.

Recoge el funcionamiento del modulo *Lances*. Contiene las llamadas a las acciones y el mecanismo de mensajes correspondientes al lanzamiento y manejo del procesado de reconocimiento de especies.

Las acciones necesarias para el funcionamiento de los diferentes módulos, llamadas desde los diálogos anteriores, se implementan en los siguientes ficheros:

- **DigApp**, **.h** y **.cpp** – incluyen la estructura de la clase **CDigApp**.

Esta clase engloba aquellas funciones necesarias para implementar la lógica de la aplicación en cuanto al acceso y manejo de la estructura de directorios y ficheros de la aplicación.

- **DigCamera**, **.h** y **.cpp** – incluyen la estructura de la clase **CDigCamera**.

Esta clase engloba aquellas funciones necesarias para la comunicación con la cámara digital, para la configuración de parámetros y captura de imágenes.

- **DigProcess**, **.h** y **.cpp** – incluyen la estructura de la clase **CDigProcess**.

Esta clase engloba aquellas funciones necesarias para implementar el procesado de imágenes, tanto los procesos de calibración del módulo *Calibración*, como el proceso de generación de patrones del módulo *Catálogo* y el reconocimiento de especies del módulo *Lances*.

- DlgFiles, .h y .cpp

En estos ficheros se implementan funciones para el manejo de ficheros de texto.

Principales funciones

En este apartado se desglosa lo más significativo de estas cuatro últimas clases.

CdlgApp

Variables y funciones para inicializar las rutas a los ficheros y directorios necesarios.

La función *InitPaths* lee del fichero de configuración *settingsapp.xml* las rutas a estos ficheros y directorios.

//directorios y ficheros del sistema private: string fileLogApp; //ruta del fichero log de la aplicación string dirResultsApp; //ruta del directorio de los resultados del procesado de imágenes //ruta del directorio que guarda las imágenes y los string dirSpeciesApp; patrones de las especies string dirCampaignsApp; //ruta del directorio que guarda los ficheros de configuración de cada campaña bool InitPaths(); //inicializa las rutas a los directorios y ficheros del sistema public:

void WriteLog(string message); //escribe un mensaje en el fichero log de la aplicación

Variables y funciones para configurar el nombre y el número de la instalación.

//parámetros de la instalación
private:
string fileInstallation; //ruta al fichero de configuración de la instalación
strInstallation paramInstallation; //parámetros de la instalación

bool LoadInstallation(); //carga los parámetros de la instalación desde el fichero fileInstallation

bool SaveInstallation(string name, int num);//guarda los parámetros de la instalación en el fichero fileInstallation

public:

```
void GetInstallation(string &name, int &num); //consulta los parámetros de la
instalación
bool SetInstallation(string name, int num); //modifica los parámetros de la
```

```
instalación
```

Funciones para la gestión del catálogo de especies y las campañas.

//especies y campañas

public:

bool GetSpeciesFolders(vector<string> &folders);//obtiene la lista de las rutas a las carpetas de las diferentes especies creadas, carpetas del directorio dirSpeciesApp.

void GetSpeciesFiles(vector<string> &files);//obtiene la lista de las rutas a los ficheros de los patrones generados, ficheros .xml del directorio dirSpeciesApp.

bool CreateSpecieFolder(string path);//crea una carpeta en el directorio dirSpeciesApp, según la ruta especificada en 'path'.

bool IsSpeciePattern(string specie); //comprueba si ya existe el patrón de la especie indicada, si existe un fichero .xml con el nombre indicado en 'specie' en el directorio dirSpeciesApp.

bool GetSamplesSide(string specie, vector<String> &pathsamples, string side);
//carga las rutas a las imágenes de muestra de la especie y cara indicadas,
ficheros .bmp que contiene la carpeta 'specie'/'side' del directorio
dirSpeciesApp.

bool RemovePattern(string specie);//elimina el patrón indicado, tanto imágenes como fichero de caracterización, el directorio y el fichero .xml con nombre 'specie' del directorio dirSpeciesApp.

void GetCampaingsFiles(vector<string> &files);//obtiene la lista de las rutas a los ficheros de las campañas configuradas, ficheros .xml del directorio dirCampaignsApp.

void SetCampaignFile(string camp, vector<string> species); //genera el fichero
que guarda la configuración de la campaña especificada 'camp' con la lista de
especies a reconocer 'species', ficheri 'camp'.xml en el directorio
dirSpeciesApp.

bool RemoveCampaign(string camp);//elimina el fichero de la campaña 'camp', fichero 'camp'.xml del directorio dirCampaignsApp.

bool GetCampaignSpecies(string camp, vector<string> &species);//carga en la lista
'species' los nombres de las especies a reconocer en la campaña 'camp', contenido
del fichero 'camp'.xml del directorio dirCampaignsApp.

Configuración del directorio y fichero de resultados.

//resultados de la clasificación

public:

void DirFileLance(int lance, string &dirlance, string &filelance, string &timeinitlance);//obtiene la configuración del directorio 'dirlance', el fichero 'filelance' de resultados del lance que se inicia, y el momento de inicio del lance 'timeinitlance' . Estos nombres se configuran con el número de lance 'lance', con los parámetros de la instalación y con la fecha y hora del sistema en el momento en que se llama a esta función.

Envío de resultados al sistema RedBox.

//comunicación con el sistema RedBox

private:

string fileRedbox;//ruta del fichero de configuración de la conexión con RedBox

public:

bool SendRedBox();//envía los resultados al directorio especificado en el fichero
de configuración fileRedbox y con las credenciales indicadas en el mismo fichero.

CDlgCamera

Variables y funciones para inicializar las rutas a los ficheros y directorios del sistema necesarios.

La función *InitPaths* lee del fichero de configuración *settingsapp.xml* las rutas a estos ficheros y directorios.

//directorios y ficheros

```
private:
```

string fileLog; //ruta del fichero de salida

string fileSettings; //ruta del script de parámetros de la cámara

string fileSettingsInit; //ruta del script de inicio de parámetros de la cámara - parámetros por defecto para el inicio de la instalación

const char* scriptfile; //fichero con todos los parámetros de configuración
de la cámara

void InitPaths(); //inicializa las rutas de los directorios y ficheros utilizados

//fichero de salida

void WriteLog(string message); //escribe un mensaje en el fichero de salida
string WriteJaiError(J_STATUS_TYPE error); //devuelve el mensaje de error
correspondiente al código de error 'error' de la librería Jai Factory.

Variables y funciones para interactuar con la cámara.

//manejo de la comunicación con la cámara

private:

//hilos de comunicación

FACTORY_HANDLE hFactory; //Factory handle

CAM_HANDLE hCamera; //Camera handle

THRD_HANDLE hThread; //Thread handle of the image stream channel

//parámetros de la imagen capturada

int32_t xSize, ySize; int64_t pixelFormat; int type;

TYPE_IMAGE_FORMAT imgFormat;

J_STATUS_TYPE retval; //código de resultado de las funciones de la librería
Jai Factory

BOOL OpenCaptureSystem(); //abre la comunicación con la cámara

void StreamCBFuncDlgProcessing(J_tIMAGE_INFO * pAqImageInfo); //función que se ejecuta cada vez que la cámara captura una imagen. En ella se copia la información de la imagen en formato Mat para manejarla con la librería OpenCV y se llama a la función 'Processing' para el inicio del procesado de la imagen capturada.

public:

bool InitCapture(APP_MODULE appmd, CStatic &boximage); //abre la comunicación con la cámara y, según el módulo desde el que se llama, también inicia la captura de imágenes. El módulo desde el que se llama se pasa en la variable 'appmd', y la dirección del marco en el que visualizará la imagen, se pasa en la variable '&boximage'.

bool StartCaptureLances(double fs, vector<string> species, string dirlance, string filelance, string timeinitlance);//prepara el modo de captura del lance, con o sin trigger, según si se especifica o no el número de imágenes a capturar por segundo 'fs', inicia la captura de imágenes, y llama a la función 'InitializeLance', de la clase CDlgProcess, que inicializa el procesado de imágenes del lance con los demás parámetros pasados a la función.

void StopCaptureLances();//finaliza la captura de imágenes y llama a la función 'FinalizeLance', de la clase CDlgProcess, que finaliza el procesado de imágenes del lance.

bool StartAcquisition();//inicia la captura de imágenes.

bool StopAcquisition();//para la captura de imágenes

void CloseCaptureSystem();//cierra la comunicación con la cámara

Variables y funciones para llar al las funciones del procesado de las imágenes capturadas y su visualización en pantalla.

//manejo de la imagen capturada

private:

Mat srcImage; //imagen en la que se vuelca la captura de la cámara.

CStatic *box_image; //dirección del marco de la ventana de diálogo en el que se muestra la imagen de salida.

APP_MODULE app_module;//identifica qué módulo del sistema llama a determinadas funciones.

void Processing(Mat in); //llama a la función de la clase CDlgProcess que corresponde según el módulo desde el que se captura la imagen, ya que según el módulo que lo llama, se realiza un procesado diferente.

void ShowImageOut(Mat im);//muestra la imagen 'im' en el marco de la ventana del sistema que visualiza las imágenes de salida.

public:

CDlgProcess myDlgProcess; //instancia de la clase CDlgProcess, que contiene las funciones de procesado que se pueden realizar sobre las imágenes capturadas. Punto de conexión entre la cámara y el procesado de imágenes.

void GetSizeImage(int32_t &x, int32_t &y);//Obtiene el tamaño de las imágenes que
está capturando la cámara.

Variables y funciones para el acceso a determinados parámetros de la cámara, necesarios para el ajuste del sistema.

//parámetros de la cámara

private:

double maxGain;//valor máximo para el parámetro "ganancia" de la cámaradouble maxTime;//valor máximo para el parámetro "tiempo de exposición" dela cámara

public:

double GetGain(); //devuelve el valor del parámetro "ganancia" que tiene configurado la cámara

double GetMaxGain();//devuelve el valor máximo del parámetro "ganancia" de la cámara

bool SetGain(double gain);//modifica el parámetro "ganancia" de la cámara

double GetTimeExposure();//devuelve el valor del parámetro "tiempo de exposición"
que tiene configurado la cámara

double GetMaxTime();//devuelve el valor máximo del parámetro "tiempo de exposición" de la cámara

bool SetTimeExposure(double exposure);//modifica el parámetro "tiempo de exposición" de la cámara

bool SaveSettingsCamera(); //guarda el script de los parámetros de la cámara con la configuración actual

bool LoadInitSettingsCamera();//carga del script de inicio de parámetros de la cámara - parámetros por defecto para el inicio de la instalación

void ChangeRoi(int up, int down, int left, int right);//llama a la función ChangeRoi de la clase CDlgProcess para variar la ROI en función de las indicaciones recibidas desde la ventana del módulo 'Dlg_Camera'

bool SaveRoi();//llama a la función SaveRoi de la clase CDlgProcess para guardar, en el fichero correspondiente, la última ROI configurada.

En la variable 'app_module', se mantiene el tipo de módulo que está activo en cada momento, para diferenciar el tipo de procesado necesario en cada situación.

typedef enum {

//identificación de los diferentes módulos del sistema

//cada módulo conlleva un comportamiento diferente en cuanto al manejo de la captura y procesado de las imágenes.

MD_CAMERA = 10, //Módulo de configuración de la cámara

MD_CALIBRATION = 20, //Módulo de calibración

MD_SAMPLES = 30, //Módulo catálogo

MD_LANCES = 40, //Módulo Lances

}APP_MODULE;

En cada módulo se crea una instancia de la clase CDlgCamera, y se inicia la comunicación con la cámara a través de la función 'InitCapture', a la que se le pasa el módulo desde el que se llama.

Una vez iniciada la captura de imágenes, cada vez que la cámara recibe una imagen, salta a la función 'StreamCBFuncDlgProcessing', en la que se vuelca la información de la imagen capturada en una variable tipo 'Mat', de OpenCV, y se llama a la función 'Processing' enviando esta imagen 'Mat'.

Es en esta última función en la que, según el módulo desde el que se capturan las imágenes, se llama a una u otra función de la clase CDlgProcess en la que se ejecuta el procesado de imagen necesario en cada situación:

- Para el módulo *Calibración del sistema* se llama a la función 'ProcessingCalibration.
- Para el módulo Catálogo se llama a la función 'ProcessingSamples'.
- Para el módulo Lances se llama a la función 'ProcessingLances'.

Éste es el punto de conexión entre las clases CDlgCamera y CDlgProcess, entre las imágenes capturadas y el procesado de imagen necesario en cada momento.

CDlgProcess

Se desarrollaron dos métodos para la caracterización de los objetos, tanto en la generación de los patrones como en el estudio de los objetos en el proceso de clasificación.

El primer método tiene en cuenta la caracterización del color de los objeto, y el segundo método tiene en cuenta tanto el color como la textura de los objetos.

El método que queda activo en la entrega del proyecto es el método 2.

Estructuras de datos utilizados para agrupar la caractericación del fondo, los valores de calibración, la caracterización de los patrones y los valores necesarios en la clasificación de cada objeto.

```
struct str_cparams
{
//Parámetros de calibración y caracterización del fondo
```

//color

```
string datecolor = "";//fecha de la última calibración de color
int negro = -1; int blanco = -1; //se guardan valores [0, 255]
```

//distance

```
string datedistance = "";//fecha de la última calibración de las distancias
double relation = 1; //relación pixels/mm
```

//background

string datebg = "";//fecha de la última calibración del fondo

int npixels = 0; //número de píxeles estudiados para la caracterización del fondo
Scalar sumargb = Scalar::all(0); //cogiendo valores CV_64F [0, 1] para asegurar
que se puede almacenar en este tipo de datos double - suma de los valores de cada
uno de los canales RGB de todos los píxeles estudiados para la caracterización
del fondo

Mat sumacrgb = Mat(3, 3, CV_64F, Scalar::all(0)); //cogiendo valores CV_64F [0, 1] - suma de las matrices de covarianza de los canales RGB de todos los píxeles estudiados para la caracterización del fondo

Scalar medrgb = Scalar::all(0); //cogiendo valores CV_64F [0, 1] - media de los
valores de cada uno de los canales RGB de todos los píxeles estudiados para la
caracterización del fondo

Mat varrgb = Mat::zeros(3, 1, CV_64FC1); //cogiendo valores CV_64F [0, 1] varianza de los valores de cada uno de los canales RGB de todos los píxeles
estudiados para la caracterización del fondo

double sdrgb = 0; //cogiendo valores CV_64F [0, 1] - desviación escalar de los valores de cada uno de los canales RGB de todos los píxeles estudiados para la caracterización del fondo

Scalar sumah = 0; //cogiendo valores CV_64F [0, 1] - suma de los valores del canal H de todos los píxeles estudiados para la caracterización del fondo

Scalar sumach = 0; //cogiendo valores CV_64F [0, 1] - suma de la covarianza de los valores del canal H de todos los píxeles estudiados para la caracterización del fondo

double medh = 0; //cogiendo valores CV_64F [0, 1] - media de los valores del canal H de todos los píxeles estudiados para la caracterización del fondo

double varh = 0; //cogiendo valores CV_64F [0, 1] - varianza de los valores del canal H de todos los píxeles estudiados para la caracterización del fondo

double desvh = 0; //cogiendo valores CV_64F [0, 1] - desviación escalar de los valores del canal H de todos los píxeles estudiados para la caracterización del fondo

};

```
struct str_features{
```

//conjunto de variables que caracterizan las muestras y los objetos que se deben clasificar

//en el caso de las muestras, serán valores acumulativos de la caracterización de todas las muestras utilizadas para generar un patrón determinado

string name = "";

string side = "";

double lenght = 0;//parámetro a del cálculo de la estimación del peso
(peso = a*(talla^b))

double weight = 0;//parámetro b del cálculo de la estimación del peso
(peso = a*(talla^b))

Mat histraw = Mat();//suma de los histogramas, en el espacio CIE-Lab, de los objetos estudiados

uint nspecimens = 0;//número de objetos estudiados

uint totalpixels = 0;//suma de los píxeles de los objetos estudiados

```
Mat histnorm = Mat();//histraw normalizado
```

#ifdef METHOD_2

Mat HistM = Mat(); //concatenación de histogramas, en el espacio CIE-Lab, de los objetos estudiados

Mat HistKM = Mat();//resultado de K-means aplicado a HistM

Mat HistVar = Mat();//varianza de color de los objetos estudiados

Mat TextM = Mat();//concatenación de los vectores de caracterización de la textura de los objetos estudiados - histogramas sobre el filtro de Sobel del canal V en las direcciones x e y

Mat TextKM = Mat();//resultado de K-means aplicado a TextM

Mat TextVar = Mat();//varianza de textura de los objetos estudiados

#endif // METHOD_2

};

Variables y funciones necesarias para el procesado de imágenes desde los diferentes módulos del sistema, propias de la clase CDlgProcess.

//principales directorios y ficheros de la aplicación

private:

string fileLog; //ruta al fichero de salida.

string fileRoi; //ruta al fichero con la ROI configurada

string fileCalibration; //ruta al fichero con los parámetros de calibración.

string dirCalibration; //ruta en la que se guardan las imágenes utilizadas en la última calibración del sistema.

string fileParametersBG;//guarda coeficientes necesarios para la separación del fondo.

string dirSpecies; //ruta en la que se guardan las imágenes y los patrones de las especies.

string fileactivecampaign;//ruta al fichero que contiene información de la campaña que está activa.

string dirResults; //directorio en el que se guardan los resultados de los procesados de imagen del módulo 'Lances'.

string dirResultlance; //directorio en el que se guardan las imágenes y el fichero de resultados del lance en curso.

string fileResultLance; //ruta al fichero de resultados del proceso de clasificación de especies del lance en curso.

string filetest; //ruta al fichero que recoge valores necesarios para
ejecutar la aplicación en modo testeo.

void InitPaths(); //inicializa la ruta a los directorios y ficheros
anteriores.

//fichero de salida

private:

void WriteLog(string message); //escribe un mensaje en el fichero log de salida

//variables y funciones para los diferentes procesados de imágenes

private:

//para diferenciar el comportamiento de pruebas, sin la cámara conectada, modo
testeo

//en este modo testeo, se leen las imágenes de cada módulo en las carpetas correspondientes del directorio iSEAS_test ('calibration' imágenes para calibración, 'samples' imágenes de muestra para la clasificación en el módulo Lances, 'results' resultados de la clasificación aplicada a las imágenes de la carpeta 'samples'

int isCamera; //indica si se está ejecutando la aplicación en modo testeo, sin cámara

int minpixelsobject; //mínimo número de píxeles que ocupa un objeto para tenerlo en cuenta en la clasificación

int overlapping; //indica si se ejecuta la clasificación suponiendo que los
objetos pueden estar o no solapados

int savesampleimages; //indica el número de imágnees de muestra que se quieren guardar durante la ejecución de un lance. Se guardará una imagen de cada 'savesampleimages'.

public:

bool GetIsCamera();//devuelve el valor de isCamera como booleano

private:

void AnalizarHist(Mat ImIN, Mat& ImOUT, int &negro, int &blanco); //calcula los niveles de negro y blanco descartando el 1 % del histograma de la imagen de entrada en los extremos inferior y superior, ImOUT es la imagen de entrada con el histograma expandido según los niveles de negro y blanco calculados.

void ColorEnhance(Mat in, Mat &out);//mejora el color de la imagen de entrada según los niveles del negro y blanco de los parámetros 'params', la imagen mejorada se devuelve en out.

void ColorEnhance2(Mat ImIN, Mat& ImOUT, int negro, int blanco); //mejora el color de la imagen de entrada según los niveles del negro y blanco indicados, la imagen mejorada se devuelve en ImOUT.

private:

Rect RoiRect; //rectángulo de la ROI configurada, que se aplica a las imágenes recibidas.

bool LoadRoi();//carga la ROI, RoiRect, desde el fichero correspondiente.

public:

bool SaveRoi();//guarda la ROI en el fichero correspondiente

void ChangeRoi(int up, int down, int left, int right, int32_t x, int32_t
y);//varía el tamaño de la ROI

private:

str_cparams params; //parámetros de la última calibración ejecutada.

vector<Mat> CalibrationMats;//vector de las imágenes capturadas durante los procesos de calibración.

TYPE_CALIBRATION typeCalibration; //tipo de calibración en curso durante los procesos de calibración.

bool calibrated; //una vez lanzada alguna calibración, indica si ésta fue
ejecutada con éxito

vector<int> parametersBG; //coeficientes para la separación del fondo.

bool LoadParametersBG(vector<int> ¶meters);//obtiene, del fichero
correspondiente, los coeficientes para la separación del fondo.

bool SaveParametersBG(vector<int> parameters);//guarda, en el fichero
correspondiente, los coeficiente para la separación del fondo.

void ClearCalibrationBG();//inicializa los parámetros de 'params' que corresponden a la calibración del fondo

void CalibrateBackground(Mat in); //calcula los parámetros de caracterización de una imagen del fondo y los acumula en los parámetros de calibración del fondo de 'params'.

public:

bool LoadCalibration();//carga los parámetros de calibración desde el fichero de calibración.

bool SaveCalibration();//guarda los parámetros de calibración en el fichero de calibración.

void GetDateCalibration(string &datecolor, string &datedistance, string &datebg);//obtiene las últimas fechas de las diferentes calibraciones.

bool CalibrateColor(); //modifica las variables necesarias para saltar a la calibración de color en la función ProcessingCalibration.

bool CalibrateDist(double height, double width); //realiza la calibración de las distancias.

bool CalibrateBG(); //modifica las variables necesarias para saltar a la calibración del fondo en la función ProcessingCalibration.

private:

string samplePath; //indica la ruta en la que guardar la imagen capturada en el proceso de captura de muestras de especies.

bool capture; //indica que se guarda una imagen capturada en la ejecución de la función ProcessingSamples, durante la visualización de imágenes del módulo 'Catálogo'.

public:

bool CaptureImage(string path);//modifica las variables necesarias para guardar una imagen durante la ejecución de la función ProcessingSamples.

private:

void SegmentarFondoPeixes(Mat ImIn, Mat& ImOUT, Mat& mascara, str_cparams cparams, double &fo);//separa el fondo de la imagen según los parámetros de fondo de la variable cparams, ImOUT devuelve la imagen con el fondo en negro, mascara devuelve la máscara que contiene los objetos extraídos del fondo, y f0 devuelve el factor indicativo de si la imagen está vacía, es todo fondo, basada en la media de los valores de la imagen sin fondo.

void AddFishSample(str_features &features, Mat In, Mat mask);//calcula los
parámetros que caracterizan la porción de imagen indicada para la clasificación,
la porción de imagen se indica con la imagen de entrada 'In' y la máscara 'mask'.
Las características calculadas se acumulan a las pasada en 'features'.

public:

void GeneratePatterns(vector<Mat> images, str_features &features);//calcula las variables de caracterización del objeto de las imágenes de entrada, va acumulando los resultados en el parámetro 'features'.

bool GeneratePatterns(string specie, vector<String> pathsamplesV, vector<String>
pathsamplesL, vector<String> pathsamplesD, int lenght, int weight);//genera el
fichero correspondiente al patrón de la especie indicada, calculando las
características de las imágenes de las caras Ventral, Lateral y Dorsal por
separado, y guardando el fichero en el directorio correspondiente.

private:

str_camp activecamp; //campaña activa

public:

bool LoadActiveCamp();//carga la campaña activa desde el fichero correspondiente.

bool SaveActiveCamp();//guarda la campaña activa en el fichero correspondiente.

bool SaveActiveCamp(str_camp camp);//guarda la campaña activa en el fichero correspondiente y cambia los valores de la variable campaña activa.

void GetActiveCamp(str_camp &camp);//devuelve la información de la campaña
activa.

void SetActiveCamp(str_camp camp); //cambia los valores de la campaña activa.

private:

bool SaveSamplesLances; //indica si se guarda un muestreo de las imágenes capturadas durante el curso de un lance.

int numimages; //va contando el número de imágenes guardadas en la ejecución del lance en curso, para marcar aquellas que han de ser mantenidas como muestra.

bool grabbingLance;//indica si el procesado de un lance está activo.

string timeStopLance;//mantiene la fecha y hora en que se finaliza el lance, por si en el fichero de resultados del lance en curso se quiere registrar este dato y no el de finalización de procesado de las imágenes de ese lance.

vector<str_features> species; //lista de los patrones de las especies a reconocer en el proceso de clasificación del lance en curso.

int numImage;//lleva el número de las imágenes que se van procesando en la clasificación del lance en curso.

int lineResults;//lleva el número de la líneas de detalle del fichero de resultados de la clasificación del lance en curso.

void LoadSpecies(vector<string> sp);//carga la lista de patrones a reconocer en el lance en curso a partir de la lista con los nombres de estas especies pasada en 'sp'.

void ResetNumImages();//pone a cero el número de imágenes procesadas en el lance en curso.

void InitLines(); //pone a cero el número de líneas de detalle del fichero de resultados del lance en curso.

void processingiSEAS_lances(Mat in);//procesa la imagen de entrada para la clasificación de los objetos detectados según los patrones de las especies a reconocer en el lance en curso.

void SetFinalProcess();//registra el momento de finalización del procesado de las imágenes del lance en curso o del último lance finalizado.

void ThreadProcessingImage();//función que se ejecuta en un hilo diferente y va procesando las imágenes del directorio de resultados del lance en curso para la clasificación de los objetos detectados en cada una de ellas, elimina cada imagen procesada a menos que esté activado el guardar un muestreo de las mismas.

public:

bool InitializeLance(vector<string> species, string dirlance, string filelance, string timeinitlance);//carga los patrones de las especies a reconocer según la lista 'species', crea el directorio y el fichero de resultados del lance en curso con los parámetros 'dirlance', 'filelance' y 'timeinitlance', registra la cabecera de este fichero, e inicializa las variables necesarias para luego lanzar el hilo que ejecuta el procesado de clasificación del lance que se inicia.

void FinalizeLance();//modifica el estado de la variable correspondiente para indicar que ya no hay ningún lance activo, se finalizó la captura de imágenes.

bool IsGrabbingLance();//devuelve el estado de la variable que indica si hay un lance activo, si aún se están capturando imágenes.

public:

//funciones a las que se difiere la ejecución cada vez que la cámara captura una imagen.

//cada una de las funciones corresponde al procesado necesario en cada uno de los módulos de la aplicación desde los que se captura imágenes.

void ProcessingCamera(Mat in, Mat &out);//vuelca en la imagen de salida, la imagen de entrada con el color mejorado.

void ProcessingCalibration(Mat in, Mat &out);//vuelca en la imagen de salida, la imagen de entrada con el color mejorado, y ejecuta el procesado de los diferentes tipos de calibración según el tipo de calibración en curso.

void ProcessingSamples(Mat in, Mat &out);//vuelca en la imagen de salida, la imagen de entrada con el color mejorado, y guarda la ROI de la imagen en el directorio correspondiente, si está indicado que se capture.

void ProcessingLances(Mat in, Mat &out);//vuelca en la imagen de salida, la imagen de entrada con el color mejorado, y guarda las imágenes en el directorio de resultados del lance en curso, marcando aquellas que se deben mantener como muestra una vez procesadas cuando así se indica.

Hilo de ejecución del procesado de imágenes

Breve explicación del hilo de ejecución del procesado de las imágenes recibidas.

Como se explica en el punto anterior, 'CDlgCamera', el funcionamiento para el procesado de una imagen es el siguiente:

- Se configura en el código la función a la que salta la ejecución en el momento en que la cámara envía una imagen.
- Una vez abierta la comunicación con la cámara, siempre se va a ejecutar esa función por cada imagen capturada.

De esta forma, si durante el hilo de comunicación que se mantiene abierto con la cámara es necesario hacer diferentes procesados de estas imágenes, ésto debe manejarse a través de variables globales de la clase CdlgProcess, que hagan que se ejecute una parte del código u otra dentro de la función mencionada, porque no se puede ejecutar una función diferente en cada momento, para ello habría que cerrar la comunicación y volver a abrirla configurando otra función a la que saltar al recibir las imágenes.

Funcionamiento:

Al abrirse un módulo, se abre la comunicación con la cámara a través de la función 'CDlgCamera::InitCapture(APP_MODULE appmd, CStatic &boximage)', en la que se indica el módulo desde el que se inicia la comunicación, que marcará qué función se debe ejecutar para procesar las imágenes recibidas.

En *InitCapture*, se crea el hilo de comunicación configurando como función de salto la función 'CDIgCamera::StreamCBFuncDIgProcessing(J_tIMAGE_INFO * pAqImageInfo)', donde una vez volcada la información de la imagen en el formato requerido, salta a la función 'CDIgCamera::Processing(Mat in)'.

Y es en esta última función en la que se desvía la ejecución a una u otra función de la instancia de CDIgProcess en CDIgCamera según el módulo especificado en la llamada a *InitCapture*.

```
switch (app_module) {
case MD_CAMERA:
    myDlgProcess.ProcessingCamera(in, out);
    break;
case MD_CALIBRATION:
    myDlgProcess.ProcessingCalibration(in, out);
    break;
case MD_SAMPLES:
```

```
myDlgProcess.ProcessingSamples(in, out);
    break;
case MD_LANCES:
    myDlgProcess.ProcessingLances(in, out);
    break;
}
```

Si una vez abierta la comunicación con la cámara desde un módulo determinado se quiere variar el procesado de las imágenes capturadas, ésto debe indicarse a través de la modificación de variables de estado de la instancia de CDlgProcess creada desde CDlgCamera.

Por ejemplo:

• En el módulo *Catálogo*:

Una vez abierta la comunicación con la cámara desde este módulo, la función final que procesa las imágenes recibidas es 'CDlgProcess::ProcessingSamples(Mat in, Mat &out)'.

Esta función simplemente mejora el color de la imagen para ser mostrada en la ventana de este módulo.

La variación que hay es si se pulsa el botón para capturar la muestra de una especie, en cuyo caso la imagen también debe ser procesada para calcular su máscara y guardarla, junto con la imagen original mejorada, en el directorio correspondiente.

Para ello, al pulsar ese botón, se llama a la función 'CDlgProcess::CaptureImage(string path)', de la instancia de CDlgProcess creada desde CDlgCamera, en la que se modifican las variables necesarias para que en el procesado de la próxima imagen capturada, se guarde ésta en el directorio especificado.

```
bool CDlgProcess::CaptureImage(string path) {
    samplePath = path;
    capture = true;
    while (capture);
    return true;
}
```

En la misma función, se espera a que desde el hilo de capturas de imágenes se indique que ya fue guardada la imagen de muestra requerida, *capture = false* en la función *ProcessingSamples*.

• En el módulo *Calibración del sistema*:

El funcionamiento es el mismo, simplemente se visualiza en la ventana la imagen mejorada y, cuando se pulsa el botón para iniciar alguno de los tipos de calibración, se llama a la función correspondiente que modifica las variables necesarias para iniciar el proceso de la calibración indicada, ejemplo:

```
bool CDlgProcess::CalibrateColor() {
    CalibrationMats.clear();
    calibrated = false;
    typeCalibration = CL_COLOR;
    while (typeCalibration != CL_NO);
    return calibrated;
}
```

El módulo *Lances* es algo diferente, ya que al iniciar un lance ejecuta la función 'CDIgProcess::InitializeLance(vector<string> species, string dirlance, string filelance, string timeinitlance)', que además de preparar el directorio y fichero de resultados de la clasificación, indica la función que maneja las imágenes capturadas, que en este caso simplemente se mejoran en color, se visualizan en la ventana de salida y se guarda su ROI en el directorio de resultados.

La función encargada del procesado de estas imágenes, que incluye el procesado de clasificación, es una función que se ejecuta en un hilo diferente y en paralelo al hilo de captura de imágenes.

Esta función es 'CdlgProcess::ThreadProcessingImage()' y el hilo se lanza desde InitializeLance:

```
numimages = 0;
grabbingLance = true;
thread second(&CDlgProcess::ThreadProcessingImage, this);
second.detach();
```

numimages se pone a cero para inicializar el contador de imágenes guardadas. *grabbingLance* se cambia de estado para que el hilo que procesa las imágenes sepa que el lance está en curso.

Referencias

[JAISDK] – Librería de funciones para el manejo de cámaras Ethernet y Jai USB. http://www.jai.com/en/support/jai_sdk_and_control_tool

[MFC] – Microsoft Foundation Class, librería de clases para desarrollo de entornos Microsoft. <u>https://msdn.microsoft.com/es-es/library/aa261798(v=vs.60).aspx</u>

[MSVS2015] – IDE Microsoft Visual Studio 2015. https://www.visualstudio.com

[OPENCV] – Librería de código abierto para procesado de imagen. http://opencv.org/
Manual iSEAS

Anexo 1

Plantilla de calibración de 14x18 cm.

Anexo II

Formato del fichero de resultados generado en el procesado de reconocimiento de especies.

El formato de este fichero, atiende a las especificación de comunicación con otro sistema externo, el sistema RedBox, que procesa la información generada por los lances y que queda fuera del alcance de este proyecto y, por tanto, de este manual.

Durante el procesado de reconocimiento de especies se van registrando los resultados del lance en un fichero de texto con **formato .csv**.

El fichero se guarda en el directorio de resultados de la aplicación, en una carpeta cuyo nombre se compone por los siguientes campos: <YYYMMDD>_<HHMMSS>_lance_<Nº_DE_LANCE>.

El *número de lance* es el número especificado en la ventana *Lances* cuando se inicia el lance.

<YYYMMDD>_<HHMMSS> - campos de la fecha y hora del inicio del lance, momento en que se pulsa el botón *Empezar lance* de la ventana *Lances*.

El nombre del fichero se forma con los siguientes campos: BEOS_<NOMBRE_DEL_BARCO>_<Nº_SISTEMA_BEOS>_<Nº_DE_LANCE>_<YYYYMMDDHHM MSS>.csv

- Nombre del barco y N.º del sistema BEOS, son el nombre y el número de la instalación especificados en la cabecera de las ventanas del sistema, modificables desde la *Ventana principal*.
- **N.º de lance** es el número especificado en la ventana *Lances* cuando se inicia el lance.
- **<YYYYMMDDHHMMSS>** campo con información de la fecha y hora del inicio del lance, momento en que se pulsa el botón *Empezar lance* de la ventana *Lances*.

El fichero está formado por una cabecera y por unas líneas del detalle del procesado de la imágenes.

La **cabecera** está formada por cuatro líneas:

- 1. Nombre del fichero sin la extensión.
- **2. <YYYYMMDDHHMMSS>** campo con información de la fecha y hora del inicio del lance, momento en que se pulsa el botón *Empezar lance* de la ventana *Lances*.
- **3. <YYYYMMDDHHMMSS>** campo con información de la fecha y hora del inicio del procesado de las imágenes guardadas durante el lance.

Manual iSEAS

4. <YYYYMMDDHHMMSS> - campo con información de la fecha y hora de la finalización del procesado de las imágenes guardadas durante el lance.

Las **líneas de detalle** del fichero reflejan el resultado del reconocimiento de cada uno de los objetos detectados en cada una de las imágenes procesadas.

Estas líneas están formadas por los siguientes campos separados por punto y coma:

- Número de línea enumera las líneas del detalle del fichero empezando por el número 1.
- **<YYYYMMDDHHMMSS>** campo con información de la fecha y hora del inicio del procesado de la imagen en la que se detecta el objeto reflejado en la línea.
- **Especie** nombre de la especie reconocida en el objeto reflejado en la línea.
- **Talla** número entero que indica el tamaño en milímetros del objeto reflejado en la línea.
- **Peso** número entero que indica el peso estimado del objeto reflejado en la línea.
- **Fiabilidad** nivel de fiabilidad en el reconocimiento de la especie a la que pertenece el objeto reflejado en la línea. Valores posibles: BAJO, MEDIO y ALTO.

Ejemplo del contenido de un fichero:

BEOS_BARCO_4_4_20170306120839 20170306120839 20170306120839 20170306120950 1;20170306120839;BACALAO;114;114;ALTO 2;20170306120839;JUREL;47;47;ALTO 3;20170306120839;BACALAO;60;60;ALTO 4;20170306120839;BACALAO;44;44;ALTO 5;20170306120839;BACALAO;28;28;ALTO 6;20170306120843;BACALAO;123;123;ALTO 7;20170306120843;JUREL;54;54;ALTO 8;20170306120843;JUREL;57;57;ALTO 9;20170306120843;JUREL;60;60;ALTO

Manual iSEAS

10;20170306120843;PLATIJA;67;67;ALTO

11;20170306120847;BACALAO;131;131;ALTO

12;20170306120847;BACALAO;116;116;ALTO

13;20170306120847;BACALAO;25;25;ALTO