



**UNIVERSIDAD
DE GRANADA**

Práctica 2

CoppeliaSim

Ismael Díaz Díaz

16 DE DICIEMBRE DE 2023

Cuestiones

1. Indica cuántos apartados de la práctica has resuelto.

He resuelto todos los apartados de la práctica.

2. ¿Qué parámetros has usado?

Parámetro	Valor
Iteraciones	50
Cerca	0.5
Media	1.5
Lejos	2.5
MinPuntos	2
MaxPuntos	30
UmbralDistancia	0.055

En cuanto a los parámetros para capturar las posiciones, he usado los parámetros por defecto, pues he visto el número de iteraciones adecuadas y las posiciones de cerca, media y lejos coinciden con las distancias de la escena final a usar “escenaTest.ttt”.

A la hora de elegir los parámetros para el resto de funcionalidades, que son el mínimo, el máximo y el umbral de distancia entre puntos de un clúster concreto, he realizado varias pruebas donde he visto reflejadas diferencias en la Cross Validation del clasificador con el kernel ‘rbf’, que es el que mejor resultados me ha dado. La mejor agrupación de parámetros que he encontrado es la descrita en la tabla.

Las capturas se han seguido realizando un movimiento uniformemente en diagonal dos diagonales, de forma que se pudiesen abarcar la mayor cantidad posible de espacio. A mayor sea el número de iteraciones, mayor será el espacio abarcado.

De igual manera, al clasificar el SVM, he tenido en cuenta los siguientes parámetros además de los kernels:

- **degree:** Este parámetro se utiliza cuando el kernel es 'poly'. Define el grado del polinomio utilizado en la función del kernel.
- **C:** Este parámetro es la penalización por error en la clasificación. Un valor más alto de C crea un margen más pequeño y puede llevar a un sobreajuste del modelo. Un

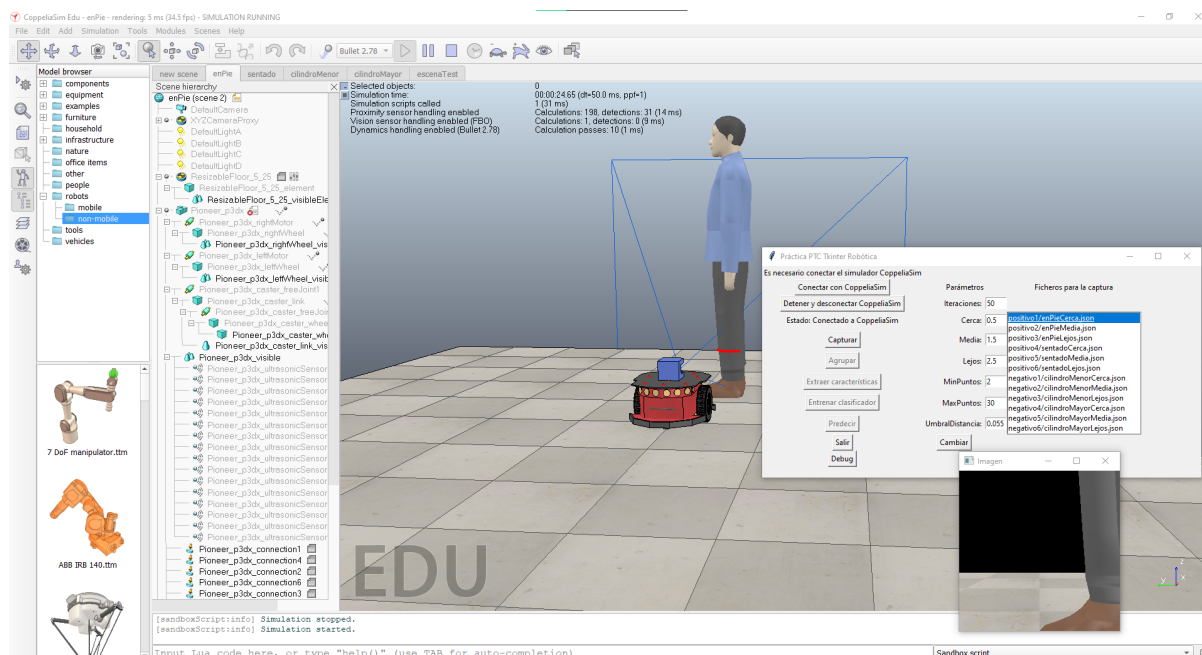
valor más bajo de C crea un margen más grande, lo que puede llevar a un subajuste.

- **gamma:** Este parámetro se utiliza cuando el kernel es 'rbf' o 'poly'. Define cuánta influencia tiene un solo ejemplo de entrenamiento. Un valor más bajo significa 'lejos' y un valor más alto significa 'cerca'.
- **test_size:** Este parámetro en `train_test_split` define la proporción del conjunto de datos que se utilizará como conjunto de prueba. Un valor más alto significa que se utilizará más datos para las pruebas, lo que puede llevar a un entrenamiento insuficiente. Un valor más bajo significa que se utilizará menos datos para las pruebas, lo que puede llevar a una prueba insuficiente.
- **random_state:** Este parámetro en `train_test_split` controla la aleatoriedad de la división de los datos en conjuntos de entrenamiento y prueba. Cambiar este valor puede llevar a diferentes divisiones y, por lo tanto, a diferentes resultados.
- **cv:** Este parámetro en `cross_val_score` y `GridSearchCV` define el número de pliegues en la validación cruzada. Un valor más alto significa más pliegues, lo que puede llevar a una mejor estimación del rendimiento del modelo, pero también a un tiempo de ejecución más largo.
- **param_grid:** Este parámetro en `GridSearchCV` define el espacio de búsqueda para los parámetros del modelo. Puedes agregar o eliminar parámetros y valores para optimizar tus resultados.

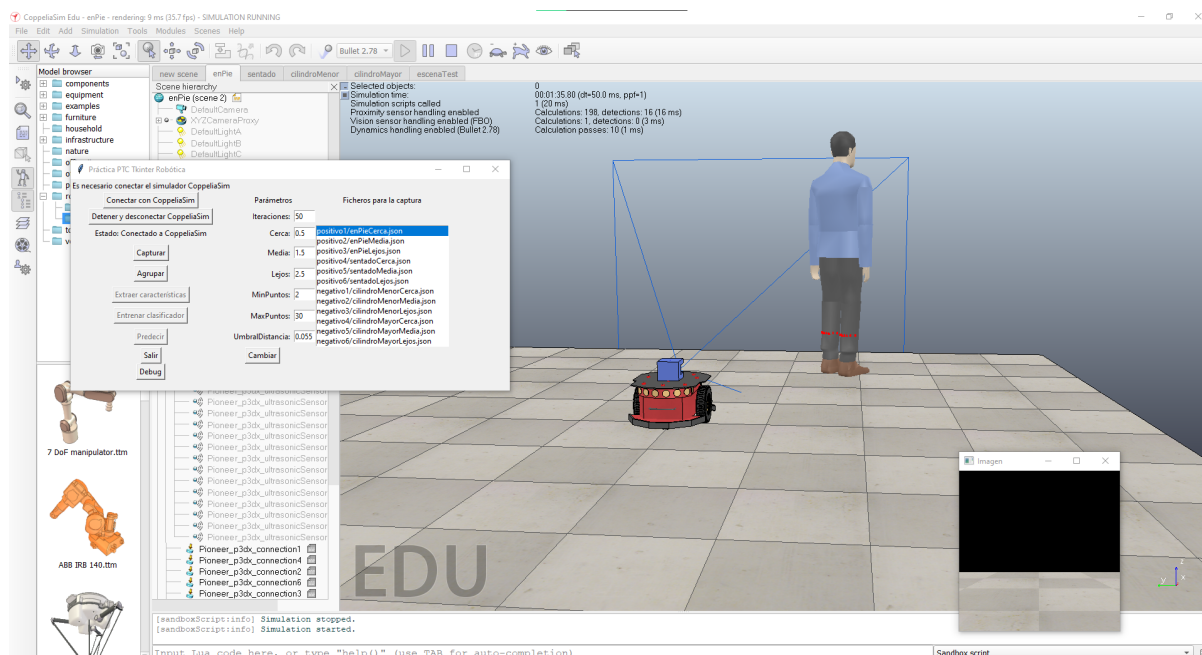
Imágenes

Sistema de captura

Iteración 3 (enPie, Cerca)



Iteración 50 (enPie, Cerca)



SVM

Fase de entrenamiento

```
El mejor estimador es: SVC(C=1000, gamma=100)
Matriz de confusión: [[ 98  7]
 [ 12 106]]
Accuracy: 0.9147982062780269
Precision= TP / (TP + FP), Recall= TP / (TP + FN)
f1-score es la media entre precisión y recall
```

		precision	recall	f1-score	support
	0	0.89	0.93	0.91	105
	1	0.94	0.90	0.92	118
	accuracy			0.91	223
	macro avg	0.91	0.92	0.91	223
	weighted avg	0.92	0.91	0.91	223

```
Cross validation:
Scores 5-cross validation: 0.8543 (+/- 0.1683)
```

Post-predicción

```
Predicción:
[1 0 1 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 1 1 0]
```

Para cada cluster encontrado, se imprime si es una pierna o no lo es, quedando como resultado la siguiente imagen ante la escenaTest.ttt:

