

자유롭게 이동하는 메카넘휠 AI 로봇 단비

Mecanum Wheel Robot AI DanVI



- 링크를 따라가려면 <Ctrl> 키를 누른채 클릭하세요.



- 메타플레이스를 검색하면 교육자료를 다운로드 할수 있어요.

1 크롬브라우저를 선택합니다.



검색창에 : **Metaplace.land** 입력합니다.



- [코딩교구&교재] – [교육자료] : 필요한 자료를 다운로드 받으세요.



메카넘휠 AI 로봇 "단비"

📄 조립설명서

(2) 필요한 자료를 Download 버튼을 눌러 다운로드

+	- 0	×
oding-kits-download.php	☆ ひ = ⊻ 🧃) :
ΛCΞ		^
코딩교구 교육자료 쇼핑몰		
교육자료 다운로드		
·일		
사료		
파일.zip	Download	
		Ψ.

- 다운로드할때 <계속>을 누르면 다운로드 완료됩니다.

만약, 아래와 같이 "안전하지 않은 다운로드가 3 차단됨 "이란 메시지가 나오면 ⇒ [계속] 을 클릭합니다.

 ✓ S 춘천 코딩교육 전문기업 Al솔= × S 춘천 코딩교육 전문기업 코딩= × + 	- x
← → C ▲ 주의 요함 metaplace.land/coding-kits-download.php	경희 :
태 앱 ⑤ Internet Services ⑤ 창직교육센터 📦 네이버 스마트스토 ☞ 강원정보문화산업 🕕 01. Matplotlib 기본 🛕 단비_인공지능교육자 계속 >	!든 북마크
^료 기시되어에 바이고 지는 고유지 - 제소	,
	>
됴.zip	
안전하지 않은 다운로드가 차단됨	
ChatGPT캠프예제파일	
▶ 건곤_S_예제소스	
► 건곤_V_예제소스	
▶ 단비_교육자료	
늘 단비_인공지능교육자료	
단비_인공지능교육자료.zip Download	
▶ 메플파 교유자료	





- 항상 압축을 풀고 사용하셔야 합니다.

1 탐색기가 열리면



파이썬 설치를 해보아요. 파이썬은 3.8.10 버전을 사용합니다.



- python_pycharm 폴더로 이동하세요.

1 다운로드 폴더에서 폴더 클릭



Step5_Auto_Drive

- arduino-ide_2.3.3_Windows_64bit
- DroidCam.Setup.6.5.2

(2)파이썬은 3.8.10 버전으로 사용합니다. - 인공지능 학습을 위해 버전을 꼭 지켜주세요.



메카넘휠 AI 로봇 "단비"

pycharm-community-2024.2.3

[주의!!] 단계별로 <mark>옵션을 잘 체크해서 천천히</mark> 따라하세요. 차근차근 체크하지 않으면 파이썬 실행시 어려움이 있어요.



[주의!!] 단계별로 옵션을 잘 체크해서 천천히 따라하세요. 차근차근 체크하지 않으면 파이썬 실행시 어려움이 있어요.



_		×
) ings, or cho	ose	
twPython38	(2	
	Cano	el

3 설치 진행을 기다립니다.

Python 3.8.10 (64-bit) Setup		<u> </u>		×	b Python 3.8.10 (64-bit) Setup	
	Setup Progress					Set
						Spe
	Installing:					free still
	Python 3.8.10 Executables (64-bit)			141		Ne
						doo
						See
						🌩 D
						Cl by
python					python	
windows			Cano	el	windows	



설치가 완료되면, 클릭합니다.

파이썬을 실행해 볼께요.

ว	키보드에서 바로	입력여	합니다.
2	Q cmd		
	← 모두 앱 문서 웹	설정 폴더	사진 🕨 75 땅 🕖 … 🍫
	가장 정확		
	명령 프롬프트 시스템		
	웹 검색		명령 프롬프트
	*	>	N= 8
	Q cmd	>	[] 열기
	Q cmd 관리자 권한	>	🕞 관리자 권한으로 실행
	Q cmd ipconfig	>	파일위치 열기
	Q cmd 명령어	>	✓ 시작 와린에 고장 ✓ 작업 표시줄에 고정
	Q cmd 정품인증	>	
	Q 층간소음	>	
	Q 층층	>	
	Q cmd prompt	>	
	〇, 층간소음 안내문 hwp	>	
		카카오	톡
		e 🦿 🦻	> 🚘 💋 🔹
	G G	위도우? 입력하	창을 클릭후, "cmd"라고 면 됩니다.

python (엔터) 입력해봅니다. **(3)** 🔜 명령 프롬프트 - python \times + ~ Microsoft Windows [Version 10.0.22631.4249] (c) Microsoft Corporation. All rights reserved. C:\Users\unikh>python Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> 파이썬 3.8.10 설치되었습니다. 🐻 명령 프롬프트 - python × + ~ Microsoft Windows [Version 10.0.22631.4249] (c) Microsoft Corporation. All rights reserved. C:\Users\unikh>python Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ^Z

메카넘휠 AI 로봇 "단비"



(5) Ctrl + Z 을 누르면 종료

파이참 설치를 해보아요. 파이참은 최신버전을 사용하셔도 됩니다.

PyCharm

데이터 과학 및 웹 개발을 위한 Python IDE



	•1					
PyCharm Community Edition 설기	~	PyCharm Community Edi	tion 열시	- 🗆 X	PyCharm Community Edition 설지	– 🗆 X
PC	PyCharm Community Edition 설치를 시작 합니다.		설치 위치 선택 PyCharm Community Edition(을)를 설치	잘 폴더를 선택해 주세요.	실치 옵 PyChar	'선 m Community Edition 설치 구성
	이 프로그램은 당신의 컴퓨터에 PyCharm Community Edition(을)를 설치할 것입니다. 설치를 시작하기 전 가능한 한 모든 프로그램을 종료하여 주시	PyCharm Community E 다른 폴더에 설치하고 싶 면 '다음' 버튼을 눌러 주	Edition(을)를 다음 폴더에 설치할 예정입니다. 으시면 '찾아보기' 버튼을 눌러서 다른 폴더를 선택 세요.	벽해 주세요. 계속하시려	데스크탑 바로가기 생성	- PATH 변수 업데이트(재시작 필요)
	기 바랍니다. 이는 재부팅을 하지 않고서도 시스템 파일을 수정 할 수 있게 해줍니다. 계속하시려면 '다음' 버튼을 눌러 주세요				컨텍스트 메뉴 업데이트	
		설치 폴더			□ "폴더를 프로젝트로 열기" 추가	
		Program Files₩JetB	irains₩PyCharm Community Edition 2024,2,3	찾아보기	연결 생성 	
		필요한 디스크 공간: 1.7 남은 디스크 공간: 352,6	GB GB			
	다음 > 취소		< 뒤로 다음	음> 취소		< 뒤로 다음 > 취소
PyCharm Community Edition 설기	치 – 🗆 X	😰 PyCharm Community Edi	tion 설치	- 🗆 X	👺 PyCharm Community Edition 설치	- 🗆 X
	시작 메뉴 폴더 선택 프로그램의 바로 가기 아이콘이 생성될 시작 메뉴 폴더 선택.		설치 중 PyCharm Community Edition(을)를 설치 세요,	하는 동안 잠시 기다려 주	Рс	Charm Community Edition 설치 완료
프로그램의 바로 가기 아이콘이 폴더 이름을 입력하세요.	생성될 시작 메뉴 폴더를 선택하세요. 새로운 폴더를 생성하려면	압축 해제 : net,dll			Py(Charm Community Edition의 설치가 완료되었습니다. 네프로그래요 미치려며 '미치' 버트운 누리 조세요
JetBrains		자세히 보기			2^	이 프로그램을 마시더한 마음 미든을 걸니 구세요.
Accessibility Accessories Administrative Tools AhnLab Arduino Cloud Agent BODA						^D yCharm Community Edition 실행하기(R)
Boris FX Continuum CYBERL DroidCam EdrawSoft EntryLabs FileZilla FTP Client Fuji Xerox	JNK					
	< 뒤로 설치 최소		< 뒤로 []	응 지수 지수		< 뒤로 ■ 마침 최소





메카넘휠 AI 로봇 "단비"

- 한국어를 선택한 상태에서 설명합니다!



라이선싱, JetBrains Marketplace 및 다른 특정 지역에 적용되는 기 능과 링크가 올바르게 동작하도록 만들기 위해 지역을 선택하세 요. 자세한 정보는 문서를 참조하세요

Korean 한국어 ~

지역이 지정되지 않았습니다 ~

다음

- 데이터 공유는 "보내지않음 "

🎦 데이터 공유

데이터 공유

Help JetBrains improve its products by sending anonymous data about features and plugins used, hardware and software configuration, statistics on types of files, number of files per project, etc. Please note that this will not include personal data or any sensitive information, such as source code, file names, etc. The data sent complies with the JetBrains Privacy Policy.

데이터 공유 환경 설정은 설치되어 있는 모든 JetBrains 제품에 적용됩니다.

이 동작은 "설정 | 모양 및 동작 | 시스템 설정 | 데이터 공유"에서 언제든지 변경할 수 있 습니다.



- 계약약관을 체크하고, <계속>합니다.

PyCharm 사용자 계약

×

JETBRAINS COMMUNITY EDITION TERMS

IMPORTANT! READ CAREFULLY:

THESE TERMS APPLY TO THE JETBRAINS INTEGRATED DEVELOPMENT ENVIRONMENT TOOLS CALLED 'INTELLIJ IDEA COMMUNITY EDITION' AND 'PYCHARM COMMUNITY EDITION' (SUCH TOOLS, "COMMUNITY EDITION" PRODUCTS) WHICH CONSIST OF 1) OPEN SOURCE SOFTWARE SUBJECT TO THE APACHE 2.0 LICENSE (AVAILABLE HERE: https://www.apache.org/licenses/LICENSE-2.0), AND 2) JETBRAINS PROPRIETARY SOFTWARE PLUGINS PROVIDED IN FREE-OF-CHARGE VERSIONS WHICH ARE SUBJECT TO TERMS DETAILED HERE: https://www.jetbrains.com/legal/community-bundled-

"JetBrains" or "we" means JetBrains s.r.o., with its principal place of business at Na Hrebenech II 1718/10, Prague, 14000, Czech Republic, registered in the Commercial Register maintained by the Municipal Court of Prague, Section C, File 86211, ID No.:

메카넘휠 AI 로봇 "단비"

3 본인은 본 사용자 계약 약관을 읽고 동의했음을 확인합니다. 종료



- 설치완료 후 〈새 프로젝트〉를 생성합니다.



메카넘휠 AI 로봇 "단비"

18

- 새로운 프로젝트를 생성합니다.

- 프로젝트 이름과 위치를 정확히 선택합니다.

I 개 프로젝트 C 2 · C · Python Python C 2 · C · Python C 2 · Django	_ 이름: Danvi_racer 위치: C:#Users#unikh#PycharmProjects		▷ 개 프로젝트 ● 순수 Python Python ○ Django	이름: 위치:	Danvi_racer C:\Users\unikh\PycharmProjects		
이름: D	anvi_racer				프로젝트가 다음의 위치에 생성됩니다. C: #Users#unikh#PycharmProjects#Danvi_racer Git 저장소 생성 회환영 인사 스크립트 생성	2	}릭
	Python 가상 환경이 프로젝트 루트에 생성됩니다. C:#Users#unikh#PycharmProjects#E	위치: 	C:\Users\unikh\Pycharr 프로젝트가 다음의 위치에 생	nProject 성됩니다	s . C:		
	생성	취소				생성	취소

- 새로운 프로젝트를 생성합니다.

- 프로젝트 위치는 <바탕화면>에 생성합니다.

🙄 경로 선택	×
$\widehat{\ } \Box \Box \times \mathcal{O} \ \widehat{\ }$	경로 숨기기
C:\Users\unikh\Desktop	~
✓ □ Users	
> 🗀 Public	
2 unikh	
> 🗅 .arduino-create	
> 🗈 .arduinolDE	
> 🖿 .jssc	
> 🗈 .mcode	
> 🗀 .pylint.d	
> 🗀 .swt	
> 🗀 ansel	
> 2 Application Data	
> 🗅 Contacts	
3 🗸 🗈 Desktop	
위 공간에 파일을 드래그앤드롭하면 배치할 수 있습니다	
্? (4) শ্বণ	취소

users



바탕화면에 폴더 (5) 생성확인!

메카넘휠 AI 로봇 "단비"

unikhee : 사용자이름은 다릅니다.

Desktop

Danvi_racer	×	+				—		×
· → ↑ C			Danvi_racer	>		Dan	vi_race	r 검색
)새로 만들기 🗸 🐰	(D (A) ¢	Û			미리 보	7
▲ 불			이름		^			수정한
🔁 갤러리			idea 🔁					2024-
			= .venv					2024-
🛓 다운로드		*						
🛄 바탕 화면		*						
릴 문서		*						

6 Danvi_racer폴더에 두개의 폴더확인

- 프로젝트 이름: Danvi_racer/ 위치를 확인해 주세요!

	🎦 새 프로젝트		-	0	×
	 	이름: Danvi_racer 위치: C:\Users\unikh\Desktop			
	> 기타	Git 저장소 생성 │ 환영 인사 스크립트 생성			
이름: Danvi_race	ſ				
위치: C:\Users\	unikh₩Deskto	p			
프루젠트가 [나오이 위치에 산	성된니다. C:WilsersWunikhWDesktonWDanvi racer			
	- 0	름과 위치를 다시한번 확인 하	세요	21	
			생성	취:	<u>*</u>





- 단비 자율주행 프로젝트를 시작합니다!



메카넘휠 AI 로봇 "단비"

유 2 실 (Shift 두 번 같 이동 Ctrl+Shift+N 파일 Ctrl+E 테이션 박 At+Home 옵 여기에 트롭하여 열기		현재 파일 🗸	æ	2+	Q	ŝ	-	O	×
@ 접색 Shift 두 번 章 아동 Ctrl+Shift+N 파일 Ctrl+E 베이션 바 Alt+Home 출 여기에 드롭하여 열기									Ļ
검색 Shift 두 번 로 이동 Ctrl+Shift+N 파일 Ctrl+E 베이션 바 Alt+Home 을 여기에 드통하여 열기									ම
검색 Shft 두 번 로 이동 Ctrl+Shift+N 파일 Ctrl+E 비이션 바 Alt+Home 을 여기에 드롭하여 열기									
검색 Shift 두 번 로 이동 Ctrl+Shift+N 파일 Ctrl+E 베이션 바 Alt+Home 을 여기에 드롭하여 열기									
검색 Shift 두 번 로 이동 Ctrl+Shift+N 파일 Ctrl+E 테이션 바 Alt+Home 을 여기에 드롭하여 열기									
검색 Shift 두 번 같 이동 Ctrl+Shift+N 파일 Ctrl+E 페이션 바 Alt+Home 잘 여기에 드롭하여 열기									
검색 Shift 두 번 로 이동 Ctrl+Shift+N 파일 Ctrl+E 데이션 바 Alt+Home 줄 여기에 드롭하여 열기									
로 이동 Ctrl+Shift+N 파일 Ctrl+E 케이션 바 Alt+Home 을 여기에 드롭하여 열기	거새 아유드 비								
파일 Ctrl+E 케이션 바 Alt+Home 을 여기에 드롭하여 열기	리국 Shint 두 권 리이동 Ctrl+Shift+N								
II이션 바 Alt+Home 을 여기에 드롭하여 열기 	파일 Ctrl+E								
을 여기에 드롭하여 열기 Python 3.8 (Danvi raced	이션바 Alt+Home								
Python 3.8 (Danvi raren)	····· 을 여기에 드롭하여 열기								
Python 3.8 (Danyi racer)									
Python 3.8 (Danyi race)									
Python 3.8 (Danyi rarer)									
Python 3.8 (Danyi raced)									
Python 3.8 (Danyi racer)									
Python 3.8 (Danyi racer)									
Python 3.8 (Danyi racer)									
Python 3.8 (Danyi racer)									
Python 3.8 (Danyi racer)									
Python 3.8 (Danyi racer)									
						Pvt	hon 3.8 (Danvi ra	cer)

전체

파일

최근

내비기

파일

22

Teachable Machine

티처블 머신에서 학습모델 "내보내기 "

메카넘휠 AI 로봇 "단비"

[Danvi_image_py] 폴더의 모든 파일을







- Danvi_racer 폴더에 파이썬 예제 파일 복사합니다.

[바탕화면]-[Danvi_racer] 폴더로 복사!!

①모델 내보내기

미리 도기 구 모델내보내기	
입력 ● 사용 Webcam ∨	
웹캠 전환 🗸 🗸	
ቲ. ር):	
↓	
출력	
motor	
stop	
etc 100%	

프로젝트에서 모델을 사용하려면 모델을 내보내세요.	×
Tensorflow.js (i) Tensorflow (i) Tensorflow Lite (i)	
모델 변환 유형: O Keras Savedrodel 보 모델 다운로드	
모델을 keras .h5 모델로 변환합니다. 변환은 클라우드에서 이루어지지만, 학습 데이터는 업로드 탄모델만 업로드됩니다.	-되지 않으며 학습이 완료
귀델에서 사용악 코트 스티펫·	
Keras OpenCV Keras	프로젝트에서 모델을
Keras OpenCV Keras from keras.models import load_model # TensorFlow is required for Keras to work from PIL import Image, ImageOps # Install pillow instead of PIL	프로젝트에서 모델을 Tensorflow.js (j
Keras OpenCV Keras from keras.models import load_model # TensorFlow is required for Keras to work from PIL import Image, ImageOps # Install pillow instead of PIL import numpy as np # Disable scientific notation for clarity	프로젝트에서 모델을 Tensorflow.js (j) 모델 변환 유형:
Keras OpenCV Keras from keras.models import load_model # TensorFlow is required for Keras to work from PIL import Image, ImageOps # Install pillow instead of PIL import numpy as np # Disable scientific notation for clarity np.set_printoptions(suppress=True)	프로젝트에서 모델을 Tensorflow.js (i) 모델 변환 유형: () Keras () Savedm
Keras OpenCV Keras from keras.models import load_model # TensorFlow is required for Keras to work from PIL import Image, ImageOps # Install pillow instead of PIL import numpy as np # Disable scientific notation for clarity np.set_printoptions(suppress=True) # Load the model model = load_model("keras_Model.h5", compile=False)	프로젝트에서 모델을 Tensorflow.js () 모델 변환 유형: () Keras () Savedment
<pre>Keras OpenCV Keras from keras.models import load_model # TensorFlow is required for Keras to work from PIL import Image, ImageOps # Install pillow instead of PIL import numpy as np # Disable scientific notation for clarity np.set_printoptions(suppress=True) # Load the model model = load_model("keras_Model.h5", compile=False) # Load the labels class_names = open("labels.txt", "r").readlines()</pre>	프로젝트에서 모델을 Tensorflow.js () 모델 변환 유형: Keras Savedn



[주의] 앞에서 예제파일 폴더에 내가 만든 모델을 덮어쓰기 해주세요!!!!

④ 두 개의파일을 파이썬 프로젝트 폴더에 복사하기







② 압축풀기

6- 텐서플로우 : 케라스 모델 다운로드하기

		X
~	🚋 압축(Zip) 폴더 풀기	
	대상을 선택하고 압축 파일을 푸십시오.	
	압축을 풀어서 다음 폴더에 저장(F):	
	C:\Users\unikh\Downloads\converted_keras (1) 찾아보기(R)	
	✔ 완료되면 압축을 푼 파일 표시(H)	
		_
	압축 물기(E) 취소	_]

- 자율주행을 위한 준비가 되었습니다!



메카넘휠 AI 로봇의 자율주행을 위한 라이브러리 설치

메카넘휠 AI 로봇 "단비"

- 라이브러리 설치하기 : 설치할 라이브러리.txt 파일 참조

- 인공지능 단비로봇의 영상을 인식하기 위해서 다음 라이브러리를 모두 순서대로 설치합니다.

PC I	☰ DR Danvi_racer ∨ 버전 관리 ∨		현재 파일 🗸 ▷ 🔅 :
	프로젝트 ~	Ξ 설치할 라이브러리.txt ×	
80	 ➤ Danvi_racer C:\Users\unikh\Desktop\ > □ .venv library 루트 	1 # 설치할 <u>라이브러리 목록입니다</u> . 2 # <u>순서대로 설치하세요</u> .	
	 ✤ 01_블루투스_검색.py ✤ 02_블루투스_조종.py ✤ 03_드로이드캠_영상수신.py 秒 04_케라스모델_입력.py 	# [파일] - [설정] - [프로젝트 : # [+]를 눌러 검색후 <u>설치할니</u> 다	Danvi_racer(각자의 프로젝트미름
	📌 09_자율주행.py 🕏 09_자율주행_영상회전.py ☲ cfg.txt 🕄 hm10.png 🕄 keras_model.h5	# keyboard 라이브러리 설치 # opencv-python 라이브러리 설치 1 # TensorFlow 라이브러리를 설치 1 # Pillow 라이브러리를 설치 1 # bleak 라이브러리 설치	:
	≡ labels.txt	1.5 14 #	
	득 설치할 라이브러리.txt	15 # HM-10의 BLE 장치 MAC 주소	." # WM_10 ㅁ듀이 MAP 조스 이려.
\$ \$	 / ш ят чм=чч ≕ 스크래치 및 콘솔 	10 address - 34.13.13.20.A1.80 17 18 19 20	· # NM-10 포즐크 NAC 구조 입국
Ø		21 #ProidCam 스트림 URL 설정	
<u>>_</u>		22 ip_address = '192.168.0.9'	# Droid <u>Cam에서</u> 확인한 IP 주소
! প			
🗆 Da	nvi_racer > 프 설치할 라이브러리.txt		22:51 CRLF



- 설정에서 메뉴를 선택합니다.

- ■ 를 볼 = 두르면 [파일]메뉴가 나타나고, [설정] – [Python 인터프리터]메뉴에서 [+]를 눌러 패키치 설치 창을 확인합니다.

		🖺 설정	
🖻 📃 🗷 Danvi_racer 🌱 버전	관리 ~	Qr	프로젝트: Danvi_racer > Python 인터프리터
프로···트 ~ 용 > □ Jerverv library 루트 ···· > □ Jere - △3? ···· ···· ····· ···· <	<t< td=""><td> > 모양 및 동작 키맵 > 에디터 플러그인 플러그인 바전 관리 프로젝트: Danvi_racer Python 인터프리터 프로젝트 구소 실당, 실행, 배포 2 언어 및 프레임워크 도구 설정 동기화 고급 설정 </td><td>Python 인터프리터: Python 3.8 (Danvi_rac Python 패킨지 도구 창에서 새로 디자인된 + + ************************************</td></t<>	 > 모양 및 동작 키맵 > 에디터 플러그인 플러그인 바전 관리 프로젝트: Danvi_racer Python 인터프리터 프로젝트 구소 실당, 실행, 배포 2 언어 및 프레임워크 도구 설정 동기화 고급 설정 	Python 인터프리터: Python 3.8 (Danvi_rac Python 패킨지 도구 창에서 새로 디자인된 + + ************************************

			×
8			$\leftarrow \rightarrow$
er) C:/Users/unikh/Desktop/Danvi_racer/.venv/So		∨ 인터프리터 추	፤7⊦ ∽
패키지 지원을 사용해 보세요.		도구 창으로 이동	×
버전 321	최신 버전 ↑ 24 2		
8.2.0	1 15.1.0		
.41.2	1.44.0		
	확인	취소 적용	÷(A)

- 이용가능한 패키지 목록확인하기

메카넘휠 AI 로봇 "단비"

- ①번과 같이 나타나면 잠시 기다립니다. ②번과 같이 목록이 나타나면 검색하여 패키지 설치를 시작합니다.

▶ 이용 가능한 패키지	×	▶ 이용 가능한 패키지	×
♀ │		<u>ि</u> न्	설명
		0	
		0-core-client (2)	
(1) 작시기다리기		0-orchestrator	
		0.0.1	
		0.618	
패키지 정보 다우로도 중		0.67.0	
		00-viet-nam-on-top-00	
		000	
		00000	
		0000000	
		0000000	
		000000000000000000000000000000000000000	
		00101s	
		00SMALINUX	
		00print_lol	
		01-distributions	
		0101	
비행 비전 지정	· · · · · · · · · · · · · · · · · · ·	01changer	
음년 음년		01d61094 d200 1100 06d1 7c5cf04ffo90	
		01001084-0298-1189-9001-705018411888	

- Keyboard를 검색창에 입력하고 패키지를 설치한다.

- 라이브러리 이름과 같은지 확인한 후, [패키지 설치]버튼을 눌러 성공적으로 설치되었다는 메시지를 확인합니다.

≌ 이용 가능한 패키지		×	🖺 이용 가능한 패키지			
Q- keyboard X	설명		Q- keyboard	×	설명	
Стккеуроаго	Hook and simulate keyboard events on	Windows and Linux	CTkKeyboard		Hook and simulate keyboard events on Windows and Lin	nux
DancePad-Keyboard	Author		DancePad-Keyboard		Author	
KeyboardPaster	BoppreH		KeyboardPaster		ВорргеН	
NoomSmartKeyboard			NoomSmartKeyboard			
Py-Keyboard	mailto:boppreh@gmail.com		Py-Keyboard		mailto:boppreh@gmail.com	
Telegram Bot Keyboard Manager	https://github.com/boppren/keyboard		TelegramBotKeyboardManager		https://github.com/boppren/keyboard	
adb-unicode-keyboard			adb-unicode-keyboard			
amharic-keyboard			amharic-keyboard			
circuitpython-softkeyboard			circuitpython-softkeyboard			
cryingonmykeyboard			cryingonmykeyboard			
django-admin-keyboard-shortcuts			django-admin-keyboard-shortcuts			
django-keyboard-shortcuts			django-keyboard-shortcuts			
dora-keyboard			dora-keyboard			
em-keyboard			em-keyboard			
em-keyboard-py3			em-keyboard-py3			
eye-keyboard			eye-keyboard			
ilp-keyboard-layout-optimization (2)			ilp-keyboard-layout-optimization			
iupyterlab-slide-mode-keyboard-shortcuts			jupyterlab-slide-mode-keyboard-shortcuts		□ 버전 지정 0.13.5	
keyboard	버전 지정 0.13.5	~	keyboard			
ксуроага-арг						
keyboard-extended	□ 옵션	<u> </u>	패키지 'keyboard'이(가) 성공적으로 설치되었습니다.			
		패키지 설치(l) 닫기(C)	(4)			달기(C)



- 다음순서로 패키지를 설치합니다.

[파일] - [설정] - [프로젝트 : Danvi_racer(각자의 프로젝트이름)] - [Python 인터프리터] # [+]를 눌러 검색후 설치합니다. 설치할 라이브러리를 순서대로 설치하세요. 특히 bleak를 마지막에 설치하는것이 좋습니다. **# opency-python # tensorflow # Dillow** # keyboard **# bleak**

메카넘휠 AI 로봇 "단비"



메카넘휠 AI 로봇의 자율주행을 위한 나만의 블루투스 이름 바꾸기 : 여러 블루투스가 있으면 나만의 이름으로 알 수 있는

블루투스로 이름을 변경해야 합니다.



블루투스에이름이 붙어있으면 이과정은 건너뛰기!

나만의 로봇을 제어하려면 블루투스 이름을 변경하기



- **아두이노 Micro 보드** 니다.
- Serial1.begin(9600); - 시리얼통신속도를 9600으로 설정
- Serial1.read(); - 시리얼포트에서 데이터 읽음
- SErial1.write(); - 시리얼포트로데이터를 씀

Output	Serial Monitor	×			
AT		No Li	ne Ending	•	9600 k
OK					
	ln 2	3. Col 2	Arduino M	licro d	on COM

메카넘휠 AI 로봇 "단비"

- **아두이노 Micro 보드에서의 블루투스 명령어 쓰는 방법**입





메카넘휠 AI 로봇의 자율주행을 위한 인공지능 학습을 시작할께요!

STEP 1. 영상연결하기 파일명: 01_드로이드캠_영상수신.py

[단계1] 드로이드캠과 스마트폰 영상연결하기

스토어에서 "드로이드캠"을 검색하여 설치할께요~

iOS 와 Android폰 아이콘 확인





[단계1] 드로이드캠과 스마트폰 영상연결하기

스마트폰에 드로이드캠을 설치후 실행시키면 다음과 같이 화면이 나타납니다.





입력합니다.

[스마트폰 드로이드캠]

메카넘휠 AI 로봇 "단비"

무선와이파이를 이용하기 위해서 스마트폰과

PC(노트북)가 <u>동일한</u> 무선 와이파이에 연결되어

WiFi 번호와 DroidCam Port 번호를 PC의 드로이드 캠에 똑같이

와이파이 연결상태 꼭! 확인필요

[단계1] 드로이드캠과 스마트폰 영상연결하기

- 드로이드캠의 WiFi IP 번호를 파이참 프로그램 ip_address에 입력합니다. ③ 현재파일 실행하기

SKT 3:44 🚥 💬	😰 🗟 .il 🛢	형재파악 > 가: :
DroidCam	÷ :	
중 WiFi		🕏 01_블루투스_검색.py 🛛 🕏 02_블루투스_조종.py 🛛 🅏 03_드로이드캠_영상수신.py
WiFi IP: 192.168.0.5 DroidCam Port: 4747		1 # opencv-python 라이브러리 설치
IP Cam Access: http://192.168.0.5:4747/ http://192.168.0.5:4747/vie	deo	
(스마트폰 화면]		5 ip_address = '192.168.0.5' # DroidCamMd 확인한 IP 주소 6 port = 4747 # DroidCamMd 전문에도 모두 (운영하도 4747) 7 video_url = f'http://{ip_address}:{port}/video' 8 9 9 # OpenCV를 사용하며 스트림 열기 10 cap = cv2.VideoCapture(video_url) 11 12 12 if not cap.isOpened(): 13 print("스트림을 열 수 없습니다.") 14 exit()
	FF	p <u>vanvi_racervus_드로미드점_영경구전.py</u> -, tine 28, in <module> == ord('q'):</module>
메카넘휠 AI 로봇 "단비"		[파이참 화면]

④ 드로이드캠과 영상수신





[영상 화면]

STEP 2. 케라스모델 확인

파일명: 02_케라스모델_입력.py

- 케라스모델의 motor, stop, etc 클래스를 인식하는지 모델입력을 해봅니다.



ott.png



st.png



et.png

PC	☰ DR Danvi_racer ∽ 버전 관리 ∽		
	프로젝트 ~	루투스_검색	.py 👌 02_블루투스_조종.py
	✓ □ Danvi racer C:#Users#unikh#Desktoo#	48	<pre>def load_and_predict_fra</pre>
80	> 🗀 .venv library 루트		<pre>class_name = class_n</pre>
	健康 01 블루투스 검색.pv		confidence_score = p
	健康 1 2 3 3 4 5 2 3 3 4 5 3 4 5 3 4 5 5 5 5 5 5 5 5 5 5 5		
	■ 03 드로이드캨 영상수신 pv		return class_name, c
	● 04 케라스모델 인령 pv		
	● 09 자육즈해 nv		
	📮 00_대로 T 8.py	70	
	$\equiv cfa tyt$	70	# OnenCV로 이미지를 있어들의
			frame = cv2.imread(image
	Introping		# 이미지가 정상적으로 읽어졌
			if frame is None:
ය	= labels.cxt		print("이미지를 읽을 수
	l ott.png		
♦	실행 💞 03_드로이드캠_영상수신 ×		
Þ	G ■ :		
	C:\Users\unikh\Desktop\Danvi	_racer\.	venv\Scripts\python.exe C
Ľ-J			
(!)			
$\overline{}$	- - -		
୍ୱ	>		
🗆 Dar	nvi_racer ゝ 之 04_케라스모델_입력.py		



- 케라스모델의 motor, stop, etc 클래스를 인식하는지 모델입력을 해봅니다.

PC	☰ DV danVi ∨ 버전 관리 ∨			현재 파일 🗸 🕞	÷0: :	ĉ+	Q	ŝ	-		×
	프로젝트 ~	01_드로이드캠_영상수신.py	🕏 02_케라스모델_입력.py	≡ labels.txt	≡ labels1.txt ×					:	Ļ
80	 ✓ □ danVi C:#Users#unikh#Desktop#danVi ✓ □ .venv library 루트 > □ Lib > □ Scripts Ø .gitignore ≡ pyvenv.cfg ♀ 01_드로이드캠_영상수신.py ♀ 02_케라스모델_입력.py ♀ 03_블루투스_검색.py ♀ 04 블루투스 조종.pv 	1 0 motor 2 1 stop 3 2 etc 4								*	@
	 C+_⊇++++C=,py ♂ 05_자율주행.py ♂ 06_자율주행_영상회전.py 至 cfg.txt ③ et.png ④ hm10.png 至 keras_model.h5 至 Keras_model_motorcycle_stop.h5 至 labels.txt 至 labels1.txt ④ ott.png ☑ st.png 										
	> 따 외부 라이브러리 = 스크래치 및 콘솔										

케라스모델에서 학습된 이미지가 첫번째 클래스인 "motor"이미지로 99% 학습된 것을 확인할수 있다.



메카넘휠 AI 로봇 "단비"



motor 클래스로 99% 인식

이미지의 파일명을 "st.png"으로 바꾸어 stop이미지가 확인되면 99% 인식됨을 알수있다.



메카넘휠 AI 로봇 "단비"



C:\Users\unikh\Desktop\Danvi_racer\.venv\Scripts\python.exe C:\Users\u 2024-10-25 10:46:27.089874: I tensorflow/core/platform/cpu_feature_gua To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX 1/1 [======] - 1s 592ms/step

^{종료 코드} (중) ^{완료된 프로세}흘러스로 99% 인식

이미지의 파일명을 "et.png"으로 바꾸어 기타이미지가 인식되면 70%로 인식된다. 기타이미지의 학습율을 높이려면 티처블머신에서 다시 교육시킬수 있다.





STEP 3. 블루투스 검색하기

파일명:03_블루투스_검색.py

[단계3] 블루투스를 검색하기

- 단비의 전원을 켜고, 블루투스의 이름을 확인한다.

PC	☰ 📴 Danvi_racer ∨ 버전 관리 ∨		현재 파일 🗸 🕞 🕻
	프로젝트 ~	🕏 01_블루투스_검색.py 🗡 🙎	
80 	 ➤ Danvi_racer C:#Users#unikh#Desktop# ➤ O.venv library 루트 > O Lib > O Scripts O .gitignore 	1 # bleak <u>라이브러린</u> 설치 2 > import 4 5 # 주변 BLE 장치 스캔 (비동기 호출 6 def scan_ble_devices_sync(): 7 loop = asyncio.new_event	물 <u>동기식으로</u> 처리) 1개의 사용 위치 _ loop() # 새로운 이벤!
(1)	┍ ┍ 01_블루투스_검색.py	<pre>8 asyncio.set_event_loop(lo 9 devices = loop.run until</pre>	pop) # 현재 이벤트 complete(BleakScanne
4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	 ♥ 02_플루루스_소종.py ♥ 03_드로이드캠_영상수신.py ♥ 04_케라스모델_입력.py ♥ 09_자율주행.py ♥ 09_자율주행_영상회전.py ■ cfg.txt ♥ hm10.png ⑨ keras_model.h5 프 labels.txt ♥ ott.png > 印 외부 라이브러리 ➡ 스크래치 및 콘솔 	10 loop.close() # 이벤트 루프 11 # 스캔된 장치 출력 13 for device in devices: 14 print(device) 15 16 # 동기식 함수 호출 17 scan_ble_devices_sync()	프 종료
🗆 Da	anvi_racer 💈 👌 01_블루투스_검색.py		17:

메카넘휠 AI 로봇 "단비"



48

[단계3] 블루투스를 검색하기

- 자신의 단비 <u>블루투스 이름</u>을 확인하고, 맥 어드레스를 복사한다.

		🔁 09 자율주행.pv	0		h	evices = loon_ru	n until comm		
			10	베르 르프 <u>조</u> 근					
			11						
		≕ cfg.txt				_			
		🕙 hm10.png				🕙 hm10.png			
		keras_model.h5				keras_model.h5			
_		≡ labels txt	14			≡ labels.txt			
						🕙 ott.png			
			16	ල		설치할 라이브러리.tx	t j		
ŝ		⇒ 설치할 라이브러리.txt		÷	심행	췓 01 북루투스 검색	×		
	실행	🥏 01 블루투스 검색 🛛 ×		\otimes					
\otimes					GI	• :			
-	G			\bigcirc		C:\Users\unikh\De	sktop\Danvi_ra		
\bigcirc		C:\Users\unikh\Deskton\Danvi	nace	Ē		57:08:7C:EF:46:DD	: None		
			1 acc	Ľ]	× _	05:75:F5:92:5B:3A	: None		
<u>>_</u>		57.00.70.EF.40.00. None (1)		Ē	₽	18:93:D7:33:21:74	a euxa		
	=	05:75:E5:92:58:3A: None		\odot	<u>=</u> ≁	71:60:26:EE:E4:E4	(<u>U)</u> 독사(<u>C</u>)		
(\cdot)	-++ 	18:93:D7:33:21:74: danvi6850		ഋ	、 、	10:04:5E:88:47:C	🗊 클립보드와 비		
	<u>-</u>	71:60:26:EE:F4:E4: None		•	1	/D.ZO.01.Z1.0/.Z	Google 검색(S)		
ಳ	>	10:04:5E:88:47:C5: None		🗆 Da	invi_race	er 👌 😴 01_블루투스_검식			
		/D.ZO.01.Z1.0/.ZC. Nono							
🗆 Da	nvi_rac	er 冫 祲 01_블루투스_검색.py							

메카넘휠 AI 로봇 "단비"



lete(BleakScan # 스캔된 장치 출력 for device in devices: print(device) # 동기식 함수 호출



STEP 4. 블루투스로 조종하기

파일명:04_블루투스_조종.py

메카넘휠 AI 로봇 "단비"



50

[단계4] 블루투스로 조종하기

AutoDrive 프로그램 소스를 업로드 시킨다. _





[단계4]블루투스로 조종하기

- 라인센서 값을 확인하여 라인트레이서 작동을 확인한다.

🕺 01_03_Ы	luetooth_line_sense	sor_drive_PID	Arduino IDE 2.3.4				— C) X		
File Edit S	Sketch Tools He	elp V	Arduino Micro		•		\checkmark	۰Q۰۰	•	PWM_max, F
P	01_03_bl	luetooth	_line_sensor_driv	ve_PID.ino	mot	or_pwm_setting.h		•••		(step)a a
	43	SMC	(6850) 바퀴 원	란조립시 rp	om(목	촉)			•	이 값들은 또
	44	PWM	(255) = 80rpr	n						
也	45	PWM	(224) = 70rpr	n				_		с
	46	PWM	(192) = 60rpr	n						
nh	47	PWM	(160) = 50rpr	n						int PWM_r
ШИ	48	*/								int PWM m
	49	//위으	PWM 속노를 .	사유롭게 소	≥성하	여 입력해 보세요.				int DWM
\oslash	50	// Ve	r 2.0(PWM 244	4 권상) /	Ver	3.0(PWM 192 권상)				ITC FWP_
	51	int P	WM_max = 192							
\bigcirc	52	int P	$WM_min = 160$;						
Q	53		WM_SLEP = 32) 이세셔 인계값	,					•	line sense
	55	int 1	ine sensor l'	imit I = 6	50.					그나이 세네이
	56	int 1	ine_sensor_1	$mit_R = 6$	50:					다인 센지의
	57				,					이 갔도의 ;
	58	// PI	D 변수 선언							ST WEL
	59	float	Kp = 1.2;	// 비례 이	<u> </u>					с
	Output						=	× 6		
	avrduo	de: 12	126 bytes of	flash wri	tten					int line
										int line
	avrduo	de done	e. Thank you	۱.						ine line
8										
			Ln 60, Col 3	30 Arduino	Micro	on COM3 [not connected	l] 🗘 ;	2 🗖		

메카넘휠 AI 로봇 "단비"

PWM_min, PWM_step 변수는 PWM 신호의 최대, 최소값과 단계 정합니다.

모터의 속도를 제어하기 위한 범위와 변화 간격을 지정합니다.

max = 192; // PWM 최대라

min = 160; // PWM 최소값

step = 32; // PWM 변화 단계

or_limit_L, line_sensor_limit_R 변수는 왼쪽(L)과 오른쪽(R) 이 임계값을 설정합니다.

라인 센서가 얼마나 민감하게 라인을 감지할지를 결정합니다.

🗗 코드 복사

🗗 코드 복사

e_sensor_limit_L = 60; // 왼쪽 라인 센서 임계값 e_sensor_limit_R = 60; // 오른쪽 라인 센서 임계값

[단계4] 블루투스로 조종하기

- 복사한 맥 어드레스 주소를 9라인에 주소를 붙여넣기 한다.

PC	☰ DR Danvi_racer ∨ 버전 관리 ∨	현재 파일 🗸 🕞 🌣
	프로젝트 ~	♂ 01_블루투스_검색.py < < O2_블루투스_조종.py ×
8 0 	 ✓ □ Danvi_racer C:#Users#unikh#Desktop# > □.venv library 루트 ◇ 01_블루투스_검색.py ◇ 02_블루투스_조종.py ◇ 03_드로이드캠_영상수신.py ◇ 04_케라스모델_입력.py ◇ 09_자율주행.py ◇ 09_자율주행_영상회전.py 三 cfg.txt 	1 # bleak <u>라이브러리</u> 설치 2 import asyncio 3 from bleak import BleakClient 4 5 # keyboard <u>라이브러리</u> 설치 6 import keyboard # 숫자키 입력을 <u>감지하기</u> 위한 <u>라이브</u> 7 8 # HM-10의 BLE 장치 MAC 주소 9 address = "18:93:D7:33:21:74" # HM-10 모듈의 MA 10
ଦ ଜ	 Inm ro.png i keras_model.h5 ≡ labels.txt ott.png 	12 CHARACTERISTIC_UUID = "0000ffe1-0000-1000-8000- 13 14 # 자동차 제어 명령
⊗	실행 🥰 🏺 02_블루투스_조종 🛛	
Þ	G 🔲 :	
ج 1	+M-10 (18:93:D7:33:21:74) 연결 →	Scripts\python.exe C:\Users\unikh\Desktop\D
🗆 Da	nvi_racer ゝ Ҿ 02_블루투스_조종.py	8:23



메카넘휠 AI 로봇 "단비"

[주의] 커서를 아래쪽에 한번 "클릭 " 하고 <u>2번을 누르면 전진, 5번을 누르면 중지</u>하는지 확인합니다~



- 2번을 누르면 전진! 5번을 누르면 중지! - 블루투스연결이 되었으므로 파이썬에서 조종이 가능합니다.

[단계 4] 블루투스로 조종하기



STEP 5. 자율주행로봇의 영상확인

파일명:05_자율주행_영상회전.py

- 스마트폰과 거치대 연결하기

1 거치대와 스마트폰을 준비



메카넘휠 AI 로봇 "단비"

결합하기

2



- 스마트폰과 거치대 연결하기

① 라인트레이서 프로그램을 업로드한 단비는 움직이죠? 옆으로 눞게하여 놓아주세요.





메카넘휠 AI 로봇 "단비"



스마트폰의 "드로이드캠 " 을 켜주세요.



- PC(노트북) 파이썬 프로그램과 단비 연결하기





- PC(노트북) 파이썬 프로그램과 단비 연결하기

1 거치대와 스마트폰을 준비







메카넘휠 AI 로봇 "단비"

3 보드에 닿지않게 단비에 끼워주기



4 보드에 닿지않도록 조심하면 좋아요!

[단계5] 자율주행로봇 단비를 전용매트에 놓아 자율주행하도록 합니다.





메카넘휠 AI 로봇 "단비"

** 인공지능카드는 AI 프레임에 붙여주세요~

METAP

- PC(노트북) 파이썬 프로그램과 단비 연결하기

PC		관리 ~				현재 파일 🗸 🕞	æ 🔲 :	ĉ+
	프로젝트 ~	🔁 01_블루투스_검색.py	🝦 02_블루투스_조종.py	🗬 03_드로이드캠_영상수신.py	🗬 04_케라스모델_입력.py	📌 09_자율주행.py	📌 09_자율	주행_영상회전.
80	 ◇ □ Danvi_racer C.#Users#un > venv library 루트 ¹ 01_블루투스_검색.py ² 01_블루투스_조종.py ² 03_드로이드캠_영상수수 ² 04_케라스모델_입력.py ² 04_케라스모델_입력.py ² 09_자율주형_영상회전. ² cfg.txt ³ hm10.png ³ keras_model.h5 ³ Keras_model_motorcyce ⁴ labels1.txt ⁴ ott.png ⁴ 설치할 라이브러리.txt > □ 외부 라이브러리 ⁴ 월 수 의 브러리.txt 	128 def control. 186 187 187 188 189 190 190 191 192 193 194 195 196 197 198 199 200 201 201 202 203 204 205 206	car_sync(): 1개의 사용 # # 10 프레임미나 예측 if frame_counter class_name, c label = f"{cl cv2.putText(f print(class_nam print('st send_comm else: send_comm frame_counter += # 자원 해제 cap.release() cv2.destroyAllWindows ally:	위치 % 10 == 0: onfidence_score = load_and_ ass_name}: {confidence_scor rame, label, org (10, 30), ame[2:], confidence_score) e[2:] in ['Class 1','motor op') and_sync(client, key: '5') and_sync(client, key: '5') 1 ()	.predict_frame(model, fr 'e:.2f}" cv2.FONT_HERSHEY_SIMPLE ,'stop']) and (confiden	ame) :X, fontScale: 1, cold ce_score >= 0.97):	or: (0, 255,	0), thickn
	실행 🌍 09_자율주행_영상회	²⁰⁷ 전 X	·# 주의 가 하는 여겨 원대					
	ଓ 🗖 🛛 :							
2 0 0 0 2 0 2	↑ '2' 명령 전송 완료 1/1 [===================================] -] -] -	Os 21ms/step Os 20ms/step Os 21ms/step Os 24ms/step	(1) 영상 인식율	상에서 ! 을 확인	보이는 하봅	2 L C	





[단계5] 자율주행로봇 단비를 전용매트에 놓아 자율주행하도록 합니다.

① 적당한 거리에서 멈추었나요?







메카넘휠 AI 로봇 "단비"

① "2"번 키보드를 눌러 앞으로 직진하도록 하세요!

PC .	置 Danvi_racer ∽ 버전	관리 ~				현재 파일 🗸 🔀	å 🔲 i	24 Q
	프로젝트 ~	🕏 01_블루투스_검색.py	🝦 02_블루투스_조종.py	🗬 03_드로이드캠_영상수신.py	04_케라스모델_입력.py	붙 09_자율주행.py	🕏 09_자율주	행_영상회전.py
	 > □ Danvi_racer C.*Users#un > ○ .venv library 루트 ◇ 01_블루투스_검색.py ◇ 02_블루투스_조종.py ◇ 03_드로이드캡_영상수(◇ 04_케라스모델_입력.py ◇ 09_자율주형_9y ◇ 09_자율주형_영상회전. ≕ cfg.txt ③ hm10.png ② keras_model_motorcyce ≕ labels.txt ≕ labels.txt ঊ labels.txt ঊ labels.txt ঊ labels.txt ঊ labels.txt ≅ dx1할 라이브러리.txt > ⑪ 외부 라이브러리 록 스크래치 및 론솔 	128 def control_ 186 187 187 188 189 190 190 191 192 193 194 195 196 197 198 199 208 201 203 203 204 205 206 fina	car_sync(): 1개의 사용 위 # 10 프레임미나 예측 if frame_counter class_name, c label = f"{cl cv2.putText(f print(class_n if (class_nam print('st send_comm else: send_comm frame_counter += # 자원 해제 cap.release() cv2.destroyAllWindows lly:	科치 % 10 == 0: onfidence_score = load_and_ ass_name}: {confidence_score rame, label, org: (10, 30), ame[2:], confidence_score) e[2:] in ['Class 1','motor' op') and_sync(client, key: '5') and_sync(client, key: '2') 1 ()	predict_frame(model, fra e:.2f}" cv2.FONT_HERSHEY_SIMPLE; ,'stop']) and (confidenc	me) (, fontScale: 1, colo :e_score >= 0.97):	r. (0, 255, 0), thickness:
	실행 🛛 🇳 09_자율주행_영상회? 	<u>e</u> ×						
	G 🔲 :							
	↑ '2' 명령 전송 완료 ↓ 1/1 [==================================] - 0	s 20ms/step					
∆ © ©] - 0	s 20ms/step					
Þ	1/1 [===================================] - 0	s 20ms/step					
र । ।	1/1 [===================================] - 0	s 20ms/step					
D Da	nvi_racer > 📌 09_자율주행_영상회	· · · · · · · · · · · · · · · · · · ·				15	9:46 CRLF UT	F-8 4개공백



PC -	☰ DR Danvi_racer ∨ 버전	관리 ~				현재 파일 🗸 🕞	æ 🗖 :	°¢†
	프로젝트 ~	🕏 01_블루투스_검색.py	👌 02_블루투스_조종.py	🗬 03_드로이드캠_영상수신.py	🝦 04_케라스모델_입력.py	🝦 09_자율주행.py	랃 09_자율주	행_영상회전
8	 > □ Danvi_racer C:#Users#un > ○ .venv library 루트 ² 01_블루투스_검색.py ² 02_블루투스_조종.py ² 03_드로이드캠_영상수전 ² 04_케라스모델_입력.py ² 09_자율주행_영상회전. ² cfg.txt ³ hm10.png ² keras_model.h5 ³ Keras_model_motorcyct ³ labels.txt ⁴ labels ⁴ labels	128 def control, 186 187 187 188 187 188 189 190 190 191 192 193 193 194 195 196 197 198 199 200 201 202 203 204 205 fin.	car_sync(): 1개의 사용 # 10 프레임미나다 예측 if frame_counter class_name, co label = f"{cl cv2.puText(f print(class_nam print('st send_comm else: send_comm frame_counter += # 자원 해제 cap.release() cv2.destroyAllWindows	위치 % % 10 == 0: confidence_score = load_and_ .ass_name}: {confidence_score rame, label, org: (10, 30), name[2:], confidence_score) ne[2:] in ['Class 1','motor' cop') nand_sync(client, key: '5') nand_sync(client, key: '5') 1	_predict_frame(model, fra re:.2f}" cv2.FONT_HERSHEY_SIMPLE: ','stop']) and (confidenc	ame) X, fontScale: 1, colo ce_score >= 0.97):	r. (0, 255, 0), thickr
	실행 🗳 09_자율주행_영상회:	전 X						
	G 🗖 :							
තී ය ම ල ගී ගී තී	 ↑ '5' 명령 전송 완료 ↓ 1/1 [==================================	=====] - =====] - =====] -	Os 22ms/step Os 20ms/step Os 24ms/step			15	9:46 CRLF UT	ΓF-8 47∦÷



PC -	☰ <mark>DR</mark> Danvi_racer ∽ 버전	관리 ~				현재 파일 🗸 🕞	æ 🗖 :	24 Q
	프로젝트 ~	🔁 01_블루투스_검색.py	🕏 02_블루투스_조종.py	👶 03_드로이드캠_영상수신.py	🕏 04_케라스모델_입력.py	<mark>幸</mark> 09_자율주행.py	🝦 09_자율주행_	영상회전.py
8	 ▷ Danvi_racer C:#Users#un ▷ .venv library 루트 ✿ 01_블루투스_검색.py ✿ 02_블루투스_조종.py ♥ 03_드로이드캠_영상수값 ♥ 04_케라스모델_입력.py ♥ 09_자율주행_영상회전. 章 cfg.txt ♥ hm10.png ♡ keras_model.h5 ⑨ teras_model.h5 ※ data and an an	128 def control_ 186 187 187 188 187 189 188 189 190 191 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 206 fina	car_sync(): 1개의 사용 4 # 10 프레임미나 예측 if frame_counter class_name, c label = f"{cl cv2.putText(f print(class_nam print('st send_comm else: send_comm frame_counter += # 자원 해제 cap.release() cv2.destroyAllWindows lly:	위치 % 10 == 0: onfidence_score = load_and_ ass_name}: {confidence_scor rame, label, org: (10, 30), ame[2:], confidence_score) e[2:] in ['Class 1','motor' op') and_sync(client, key: '5') and_sync(client, key: '2') 1 ()	_predict_frame(model, fr re:.2f}" cv2.FONT_HERSHEY_SIMPLE	ame) EX, fontScale: 1, col ce_score >= 0.97):	or. (θ, 255, θ),	thickness:
	실행 🗳 09_자율주행_영상회간	907 <u>M</u> X	ווד⊍: גָּוָס סום וו ⊂י⊼ #					
	¢ □ :							
© ⊕ ⊘	 ↑ '5' 명령 전송 완료 ↓ 1/1 [==================================	=====] - 0 ======] - 0	us 25ms/step us 22ms/step					
\diamond	stop '5' 명령 전송 완료							
) ()	1/1 [===================================] - 0	s 21ms/step					
۲ ₀Da	invi_racer > 📌 09_자뮬주행_영상회	I전.py					59:46 CRLF UTF-8	4개 공백



메카넘휠AI로봇"단비" Mecanum Wheel Robot Al Dan VI

THANK YOU

궁금한 부분이 있으시면 아래의 메일로 연락주세요.

1ETAPLACE

E-mail : metaplace@naver.com Tel : 033-252-4787

