

DAIGTŲ INTERNETO PROGRAMAVIMO UŽDUOTIS.

Sukurkite sistemą, kuri analizuotų aplinką ir atpažintų penkis skirtingus objektus. Programa - atpažinusi nurodytus objektus turi valdyti apšvietimą – įjungti šviesą, kai tam tikras objektas yra aptinkamas ir išjungti šviesą, kai aptinkamas kitas objektas.

Sukurkite sistemą, atpažįstančią penkis skirtingus objektus. Ji turi:

- Analizuoti aplinką;
- Atpažinti nurodytus objektus;
- Valdyti apšvietimą - ...

Užduotį galite atlikti vadovaudamiesi apraše pateiktu sprendimo būdu. Jeigu jums pažįstamas tokio tipo eksperimentas – galite naudoti ir kitus jums žinomus įrankius.

Eksperimentui atlikti jums reikės naudoti tokias sistemas, kaip:

1. Teachable machines (URL: <https://teachablemachine.withgoogle.com>)
Teachable Machines tai Google kompanijos sukurtas įrankis. Jo paprasta aplinka, ir yra lengvai naudojamas. Įrankio pagalba galima kurti mašininio mokymosi modelius ir neturintiems programavimo žinių. Šis įrankis leidžia apmokyti modelį atpažinti vaizdus, garsus ar pozas. Vartotojai gali pateikti pavyzdžius, o mašininio mokymosi modelis išmoks klasifikuoti naujus, nematytus pavyzdžius pagal šiuos įvesties duomenis. Šiandien – jums pasiūlysiu naudoti vieną iš įrankio galimybių – apmokyti ir atpažinti vaizdus.
2. Python programavimo kalba,
3. Programavimo aplinka *Spyder** (galima ir kita),
4. Papildomos python bibliotekos (*paho-mqtt, keras, numpy, OpenCV, pillow*)

Atliktos užduoties vertinimas balais - taisyklės:

A. Pilnai atlikta Pirmoji Užduotis. Eksperimentas. Duomenų surinkimas. Surinktų duomenų apmokymas. Dirbtinio intelekto sukurtu modelio eksportavimas	1 balas
B. Pilnai atlikta Antroji užduotis - Programavimo aplinkos paruošimas, <i>Python</i> bibliotekų diegimas, ir programinio kodo užkrovimas.	1 balas
C. Pilnai atliktas Programavimas. Aptikus objektą šviesa įjungiama/išjungiama.	1 balas



A. PIRMOJI UŽDUOTIS. EKSPERIMENTAS:

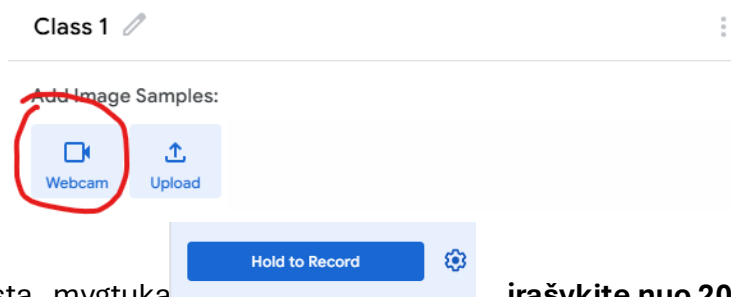
Naudodami [Teachable Machines “Image” įrankį](https://teachablemachine.withgoogle.com/train/image)

(<https://teachablemachine.withgoogle.com/train/image>) atlikite tokius veiksmus:

1. Surinkite duomenis: Jums reikia surinkti penkių skirtingų objektų (kategorijų) duomenis, kuriuos pavadinkite skirtingais pavadinimais. Pavadinimui naudokite aiškių, trumpą žodį be lietuviškų raidžių. **Filmuokite tokia eilės tvarka:**

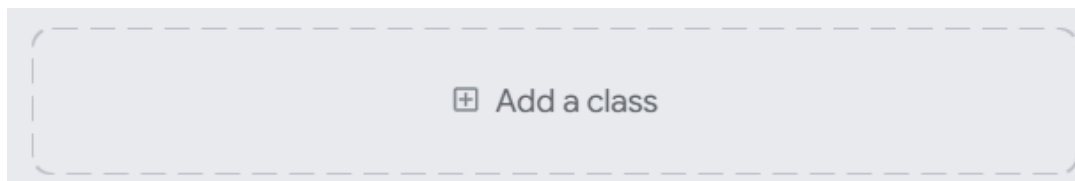
- 1) Pirmojo moksleivio veidas
- 2) Antrojo moksleivio veidas
- 3) Vieno iš moksleivių delnas (pirštai turėtų būti suspausti/suskleisti)
- 4) “V” ženklas, (atlenkti du pirštai)
- 5) Juoda uždanga (uždaryta kamera, naudokite jungtuką kameros viršuje)

Objektų filmavimo pradžia paspauskite “Webcam” mygtuką:



Ir laikydami paspaustą mygtuką **Hold to Record**, įrašykite nuo 200 iki 300 vaizdų. Filmuodami objektus juos judinkite (sukiokitės), kad kuo daugiau informacijos būtų surinkta. Šiuos veiksmus atlikite su penkiais skirtingais objektais, išvardintais aukščiau.

Pridėti naujo objekto filmavimo galimybę, pasirinkite “Add a class” komandą, kaip parodyta žemiau:

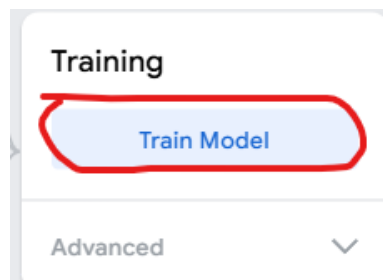


Kiekvieną klasę (nufilmuotą objektą) pavadinkite DIDŽIOSIOMIS RAIDĖMIS pagal prasmę, tuo tarpu veidus pavadinkite nufilmuoto moksleivio vardu. Pavadinime negali būti lietuviškos rašybos raidžių. Keiskite jas lotyniškoms raidėms, arba “švepluokite”.

Klasės pavadinime taip pat negali būti tarpų.

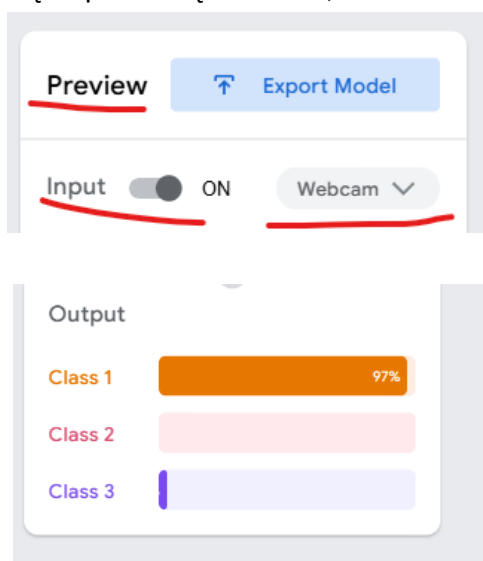
Atlikus penkių klasių pavyzdžių įrašymus (nufilmavus penkis objektus) atlikite mašininio modelio mokymus, pagal ką tik surinktus duomenis.

2. Treniravimas (apmokymas): *Teachable Machine* naudoja jūsų pateiktus pavyzdžius modelio treniravimui. Šis žingsnis vyksta automatiškai, ir galite stebėti treniravimosi eigą. Modelio apmokymui, pasirinkite **“Train model”**.



3. Testavimas Po apmokymo (treniravimo) galite išbandyti modelio tikslumą pateikdami jam naujus pavyzdžius.

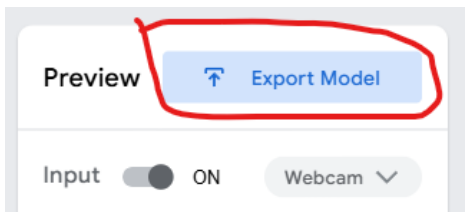
“Preview” dalyje ištestuokite modelio veikimą patikrindami, ar teisingai išskiria objektus. Kuo tiksliau atpažįstą veidą, plaštaką ar plaštaką su veidu, tuo didesniu procentu tai įvertina:



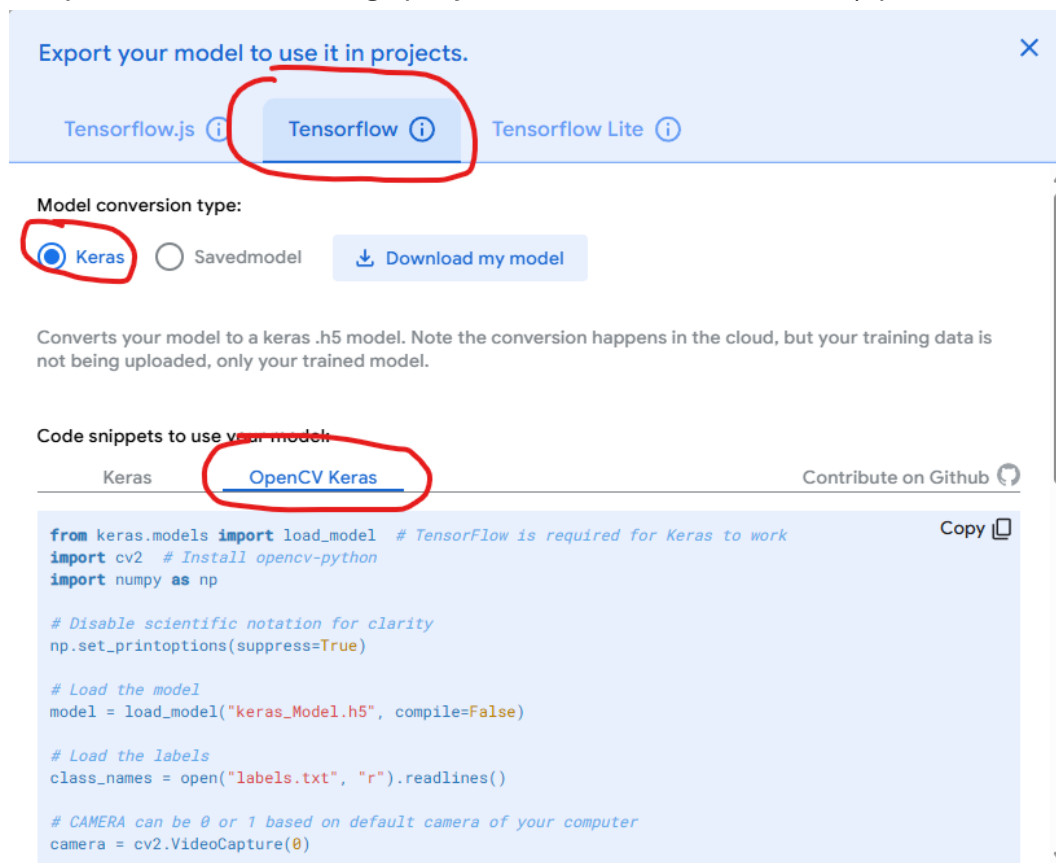
Išbandykite *“Teachable Machine”* aplinoje. Ir pažiūrėkite, ar esate patenkinti objektų aptikimo ir atpažinimo rezultatais

Kai būsite patenkinti veikimu, galėsite **eksportuoti modelį** į rinkmeną (failą) ir naudoti programėlėse, svetainėse ar kituose projektuose:

4. **Mašininio modelio sukūrimas ir eksportavimas.** Tam pasirinkite šiuos parametrus naršyklės ekrane. Paspauskite “Export Model”:



Naujai atsidariusiame lange pažymėkite svarbius elementus (apvesta raudonai)

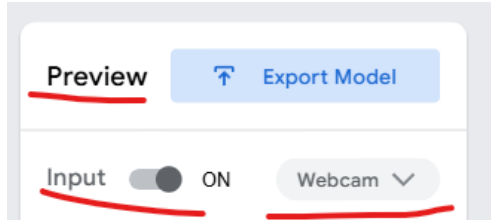


Ir pasirinkę , minutėlę luktelkite, kol atliks skaičiavimus ir bus atsiųstas jums suarchyvuota rinkmena (failas).

Sveikiname. Pirmąją užduotį jūs atlikote.

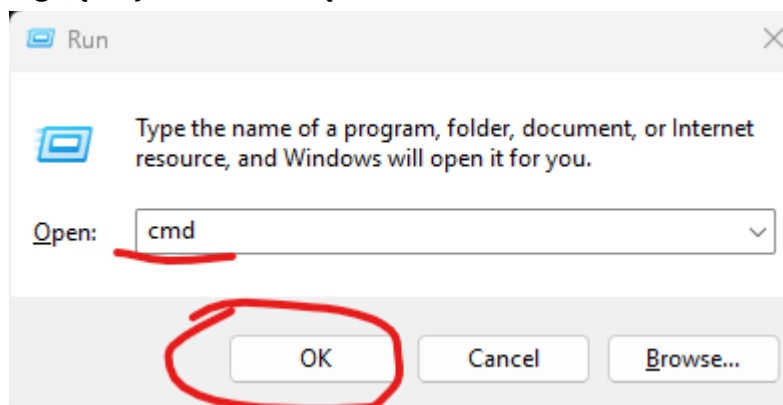
B. ANTROJI UŽDUOTIS. Programavimo aplinkos paruošimas.

“Teachable Machine” aplinkoje “preview” parinkite OFF.

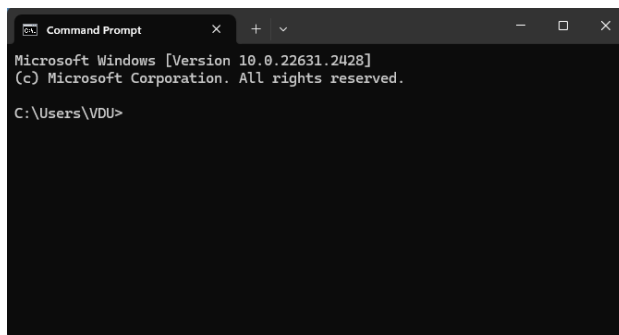


Tam, kad su programavimo kalba turėtumėte visas teises dirbti be apribojimų, jums reikia paleisti *Python* programavimo kalbą naudojant virtualią aplinką. Šiam tikslui įgyvendinti atlikite tokius veiksmus:

- 1) vienu metu paspauskite sparčiųjų klavišų kombinaciją „Windows” + R.
- 2) atsidariusiame lange, įrašykite komandą **cmd**



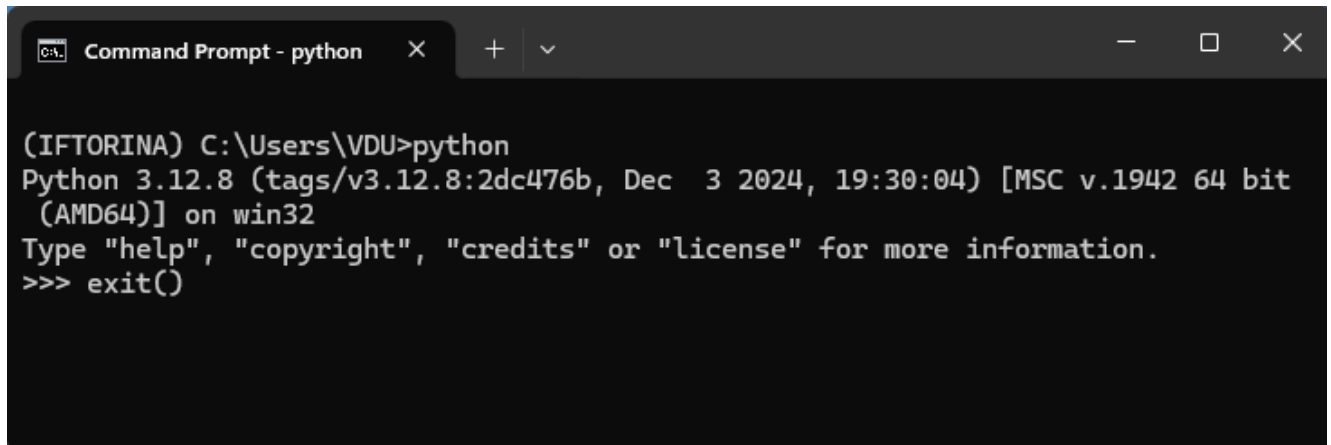
- 3) nuspaudus „OK“ mygtuką - atsidaro naujas „juodos spalvos” langas – terminalas.



Pastaba: šio lango negalima uždaryti, kol jūsų darbas bus neįvertintas.

Tekstas iki mirgsinčio kursoriaus (C:\Users\VDU>) Jums rodo darbinio katalogo vietą. Kartais jis galio būti ir kitoks.

4) patikrinkite ar kompiuteryje įdiegtas, ir veikia *python* programavimo kalba. Tam tikslui „juodajame“ lange parašykite komandą ***python*** (ir nuspauskite enter).

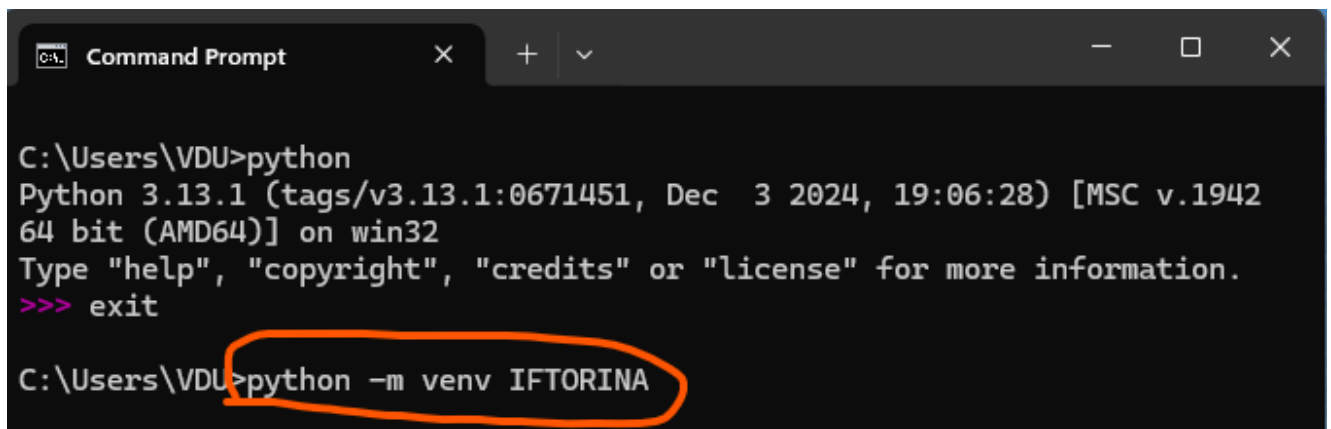


```
(IFTORINA) C:\Users\VDU>python
Python 3.12.8 (tags/v3.12.8:2dc476b, Dec 3 2024, 19:30:04) [MSC v.1942 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
```

Jeigu ekrane atspausdinamas panašus pranešimas, kaip pateikta paveiksle, aukščiau, surinkite ***exit()***, arba ***quit()***, arba CTRL-Z.

5) Sukurti python programavimo kalbai skirtą virtualią aplinką, jums reikia juodame lange

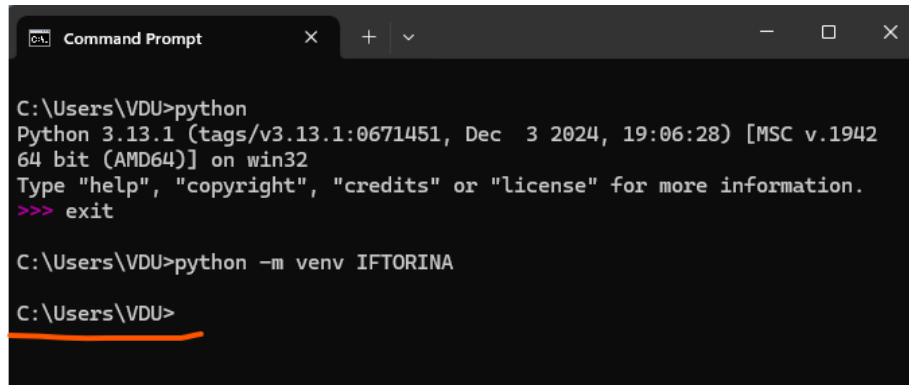
surinkti komandą: `python -m venv IFTORINA`



```
C:\Users\VDU>python
Python 3.13.1 (tags/v3.13.1:0671451, Dec 3 2024, 19:06:28) [MSC v.1942
64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit

C:\Users\VDU>python -m venv IFTORINA
```

Sėkmingai įvykdžius užduotį, ekrane nebus jokio pranešimo apie klaidą ir „juodajame lange“ atsiras naujas pranešimas:



```
Command Prompt
C:\Users\VDU>python
Python 3.13.1 (tags/v3.13.1:0671451, Dec 3 2024, 19:06:28) [MSC v.1942
64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit

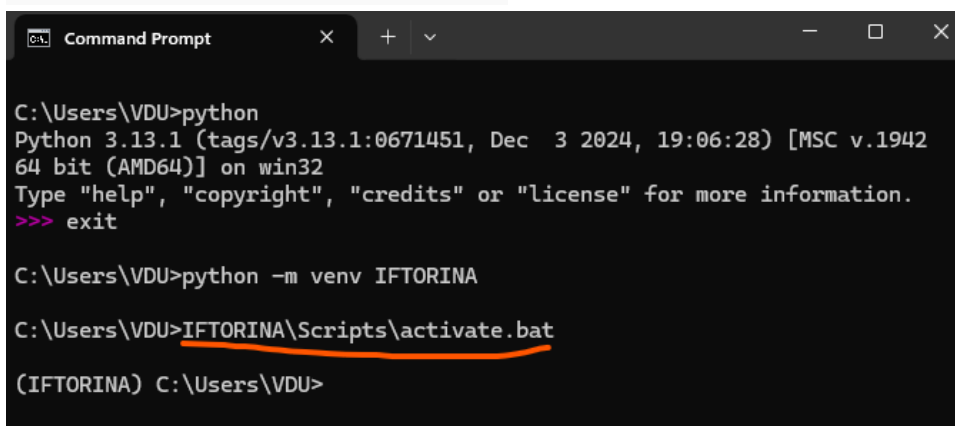
C:\Users\VDU>python -m venv IFTORINA

C:\Users\VDU>
```

Taip bus sukurta programavimui su *python* skirta, tik jums vieniems skirta virtuali aplinka IFTORINA.

6. **Aktyvuokite python aplinką** tolimesniam darbui. Tam surinkite komandą:

IFTORINA\Scripts\activate.bat



```
Command Prompt
C:\Users\VDU>python
Python 3.13.1 (tags/v3.13.1:0671451, Dec 3 2024, 19:06:28) [MSC v.1942
64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit

C:\Users\VDU>python -m venv IFTORINA

C:\Users\VDU>IFTORINA\Scripts\activate.bat

(IFTORINA) C:\Users\VDU>
```

Teisingai parašius komandą, ir ją įvykdžius „juodajame“ lange atsiras užrašas:

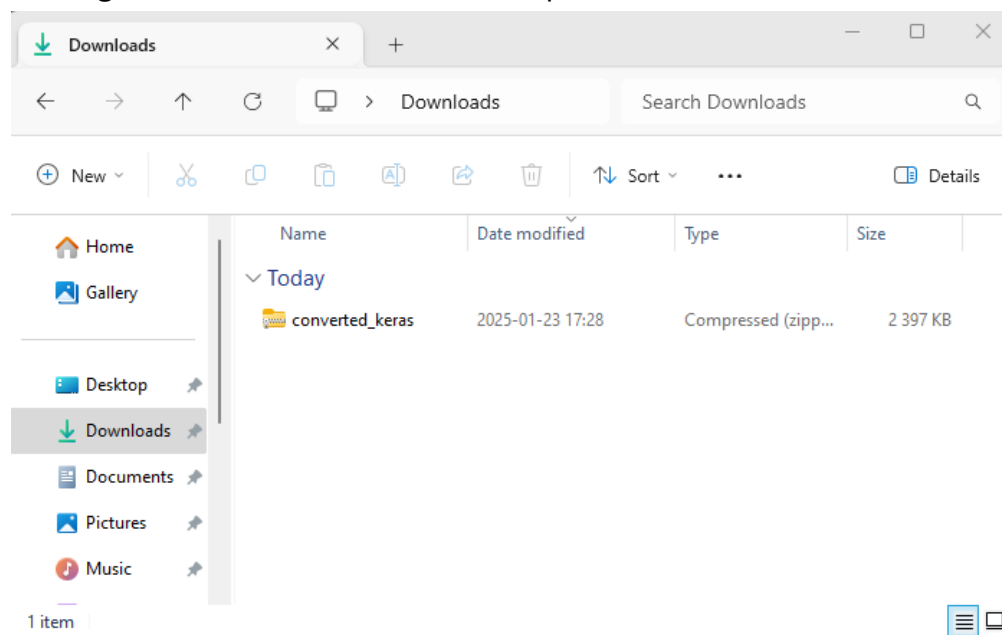
(IFTORINA) C:\Users\VDU> , kaip parodyta aukščiau paveiksle, apatinėje eilutėje.

Atlikus šį veiksma, - jūsų terminalas (juodasis langas) dirbs tik jums skirtoje aplinkoje, ir programuoti galėsite be jokių apribojimų

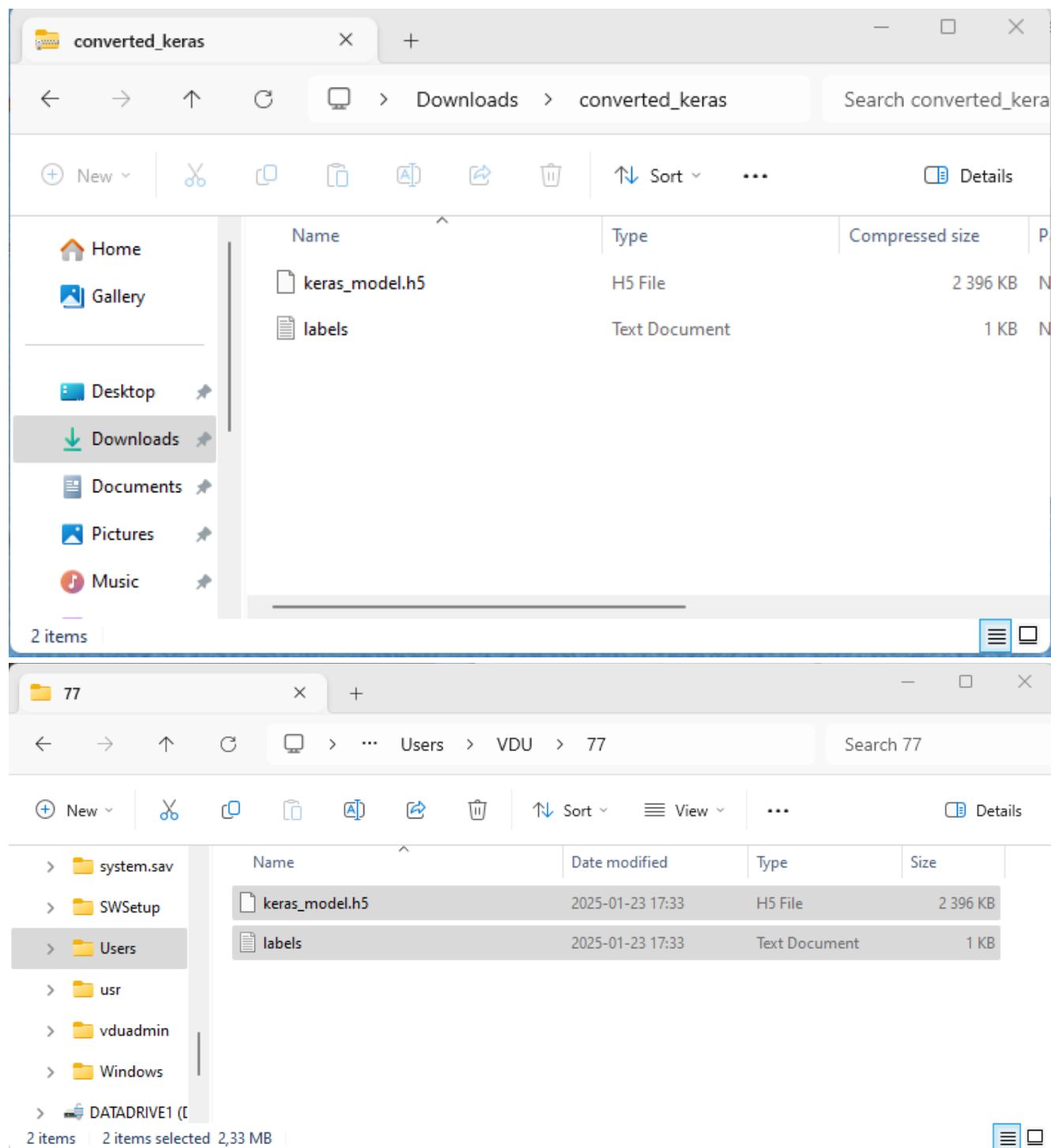
7. Kataloge **C:\Users\VDU** **sukurkite naują katalogą** - kurio pavadinimas jūsų komandos numeris (pvz, 77) . tai padaryti galite su komanda **md 77** , kurią rašykite „juodo lango eilutėje“

```
C:\Users\VDU>md 77
```

8. Į naujai sukurtą katalogą (pvz.: **C:\Users\VDU\77**) **įrašykite** apmokymų metu sukurtą dirbtinio intelekto mašininį modelį, kuriuos atsisiuntėte **A dalyje 4-tajame punkte**. Jeigu nepakeitėte failo pavadinimo, suarchyvuotas dokumentas turėtų būti įrašytas „Downloads“ kataloge ir vadintis „converted_keras.zip“.



9. Atidarykite archyvą `converted_keras.zip`, ir naudodami standartines Windows galimybes nukopijuokite abu dokumentus, `keras_model.h5` ir `labels.txt`, į savo komandos numerio katalogą (pvz., `C:\Users\VDU\77`).



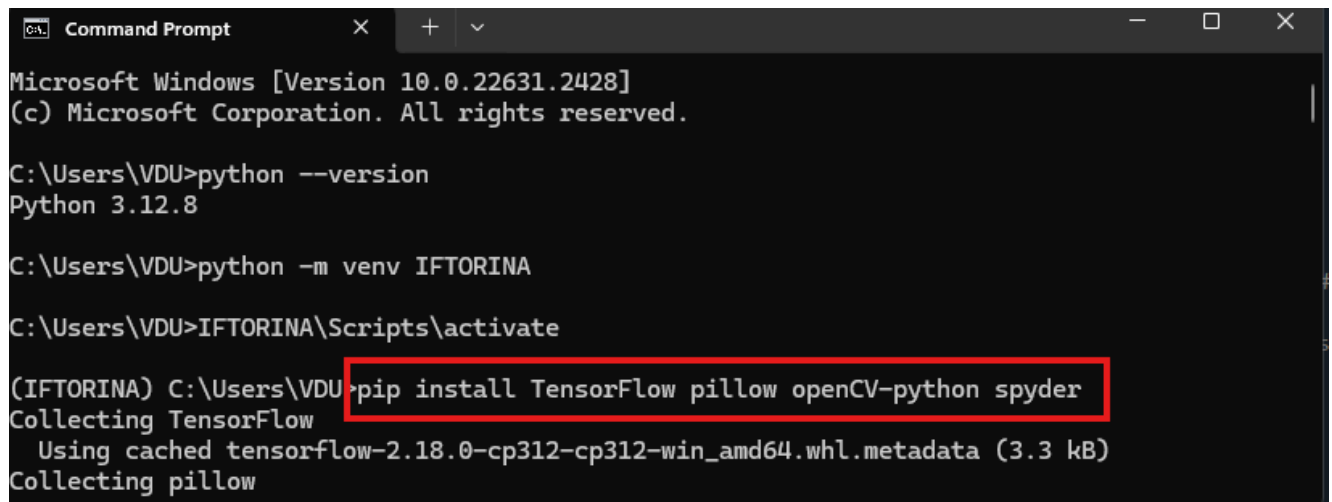
10. Įdiekite papildomas bibliotekas ir grafinę programavimo aplinką, kurios bus skirtos uždaviniui atlikti. Trūkstantomis bibliotekos, tai (*paho, keras, numpy, OpenCV, pillow*), ir programavimo aplinka *spyder*. (vietoj *spyder* galite naudoti ir kitą, jūsų mėgiamą programavimo aplinką, jeigu su tokia esate dirbę).

Trūkstančių bibliotekų ir *spyder* grafinės aplinkos diegimui, jums reikia surinkti šias komandas.

Svarbu, kad žemiau pateikta komanda būtų įvykdyta be klaidų:

```
pip install TensorFlow pillow openCV-python spyder
```

Juodajame lange vyks prašomų įrankių atsisiuntimo ir diegimo procesai.



```
Command Prompt
Microsoft Windows [Version 10.0.22631.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\VDU>python --version
Python 3.12.8

C:\Users\VDU>python -m venv IFTORINA

C:\Users\VDU>IFTORINA\Scripts\activate

(IFTORINA) C:\Users\VDU>pip install TensorFlow pillow openCV-python spyder
Collecting TensorFlow
  Using cached tensorflow-2.18.0-cp312-cp312-win_amd64.whl.metadata (3.3 kB)
Collecting pillow
```

Jeigu ekrane nebus jokio pranešimo apie klaidą, ir atsiras tik nauja tuščia eilutė:

```
(IFTORINA) C:\Users\VDU\77>|
```

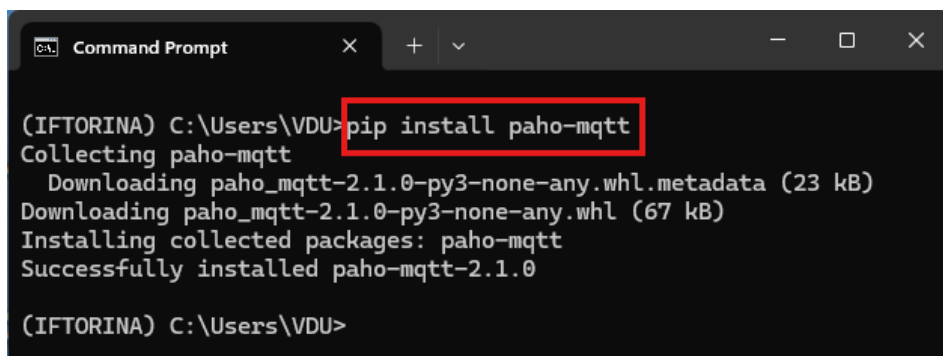
reiškia - diegimas įvykdytas sėkmingai. Kitu atveju ieškokite sintaksinių rašybos klaidų.

13) tam, kad būtų galima valdyti daiktų interneto valdiklį, ir valdyti šviesos uždegimą, reikia **įdiegti python skirtą valdyti MQTT protokolui**, kurios užtikrins valdiklio ir įrenginio valdymą.

Svarbu, kad žemiau pateikta komanda būtų įvykdyta be klaidų:

Tam „juodajame“ lange, surinkite komandą:

```
pip install paho-mqtt
```



```
(IFTORINA) C:\Users\VDU>pip install paho-mqtt
Collecting paho-mqtt
  Downloading paho_mqtt-2.1.0-py3-none-any.whl.metadata (23 kB)
  Downloading paho_mqtt-2.1.0-py3-none-any.whl (67 kB)
Installing collected packages: paho-mqtt
Successfully installed paho-mqtt-2.1.0

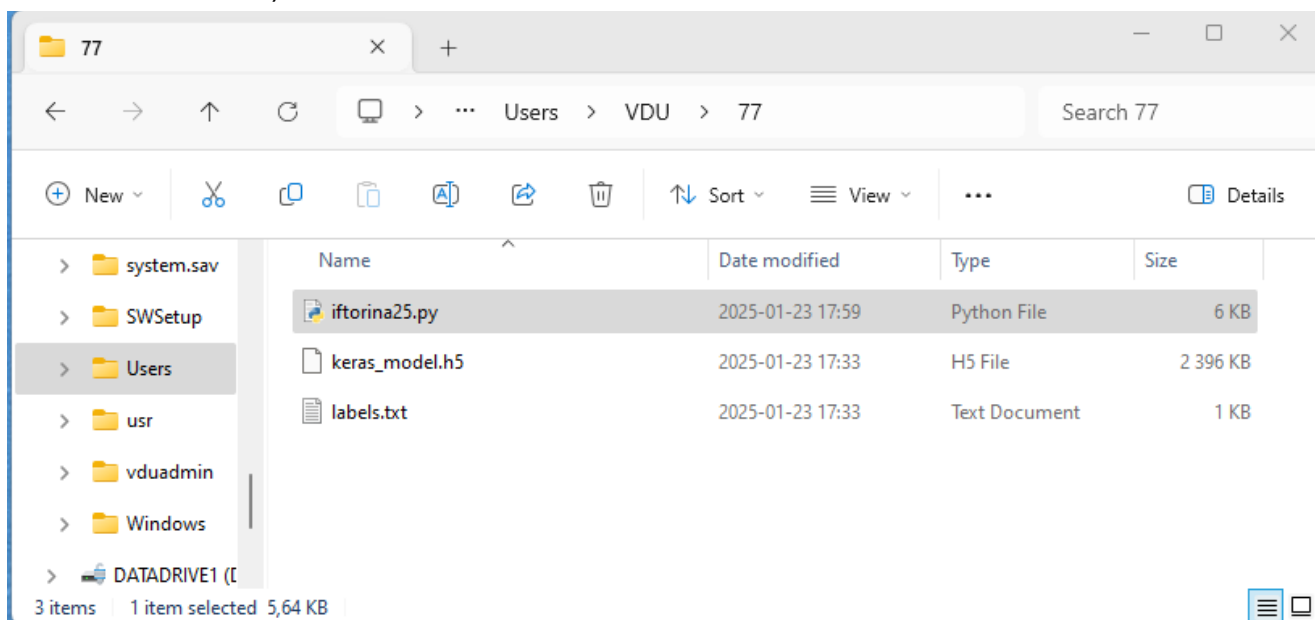
(IFTORINA) C:\Users\VDU>
```

Jeigu neatsiras ekrane jokio pranešimo apie klaidą, - bibliotekos diegimas įvykdytas sėkmingai.

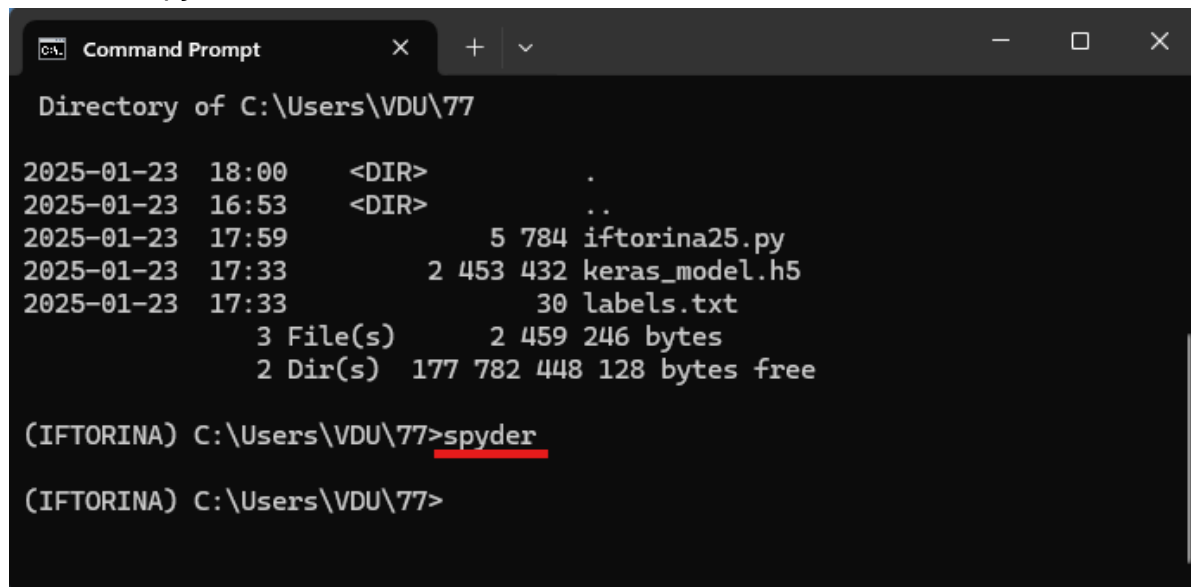
12) **atsisiųskite programos šabloną**. Programą, pavadinimu *iftorina25.py* rasite internete adresu: <https://github.com/audriuszj/IFTorina>

13) **programą išsaugokite** tame pačiame kataloge, kur yra išsaugotas jūsų apmokytas mašininės kalbos modelis, laikmenos *keras_model.h5* ir *labels.txt*. (pvz.:

C:\Users\VDU\77)



14) iškvieskite spyder aplinką, kurioje galėsite programuoti, tam „juodajame lange“ surinkite komandą `spyder`:



```
Command Prompt

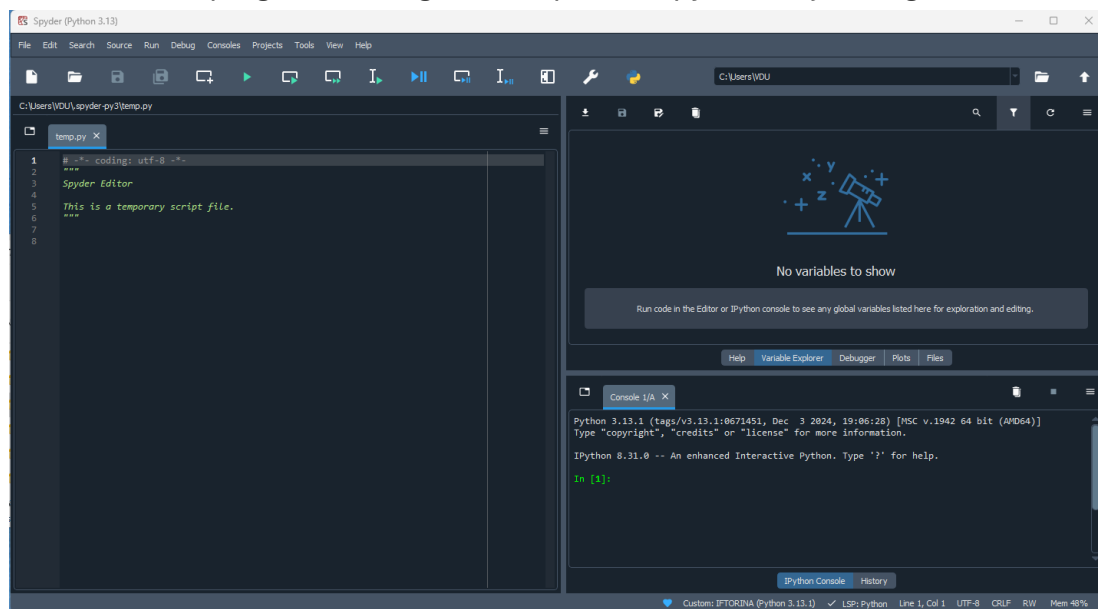
Directory of C:\Users\VDU\77

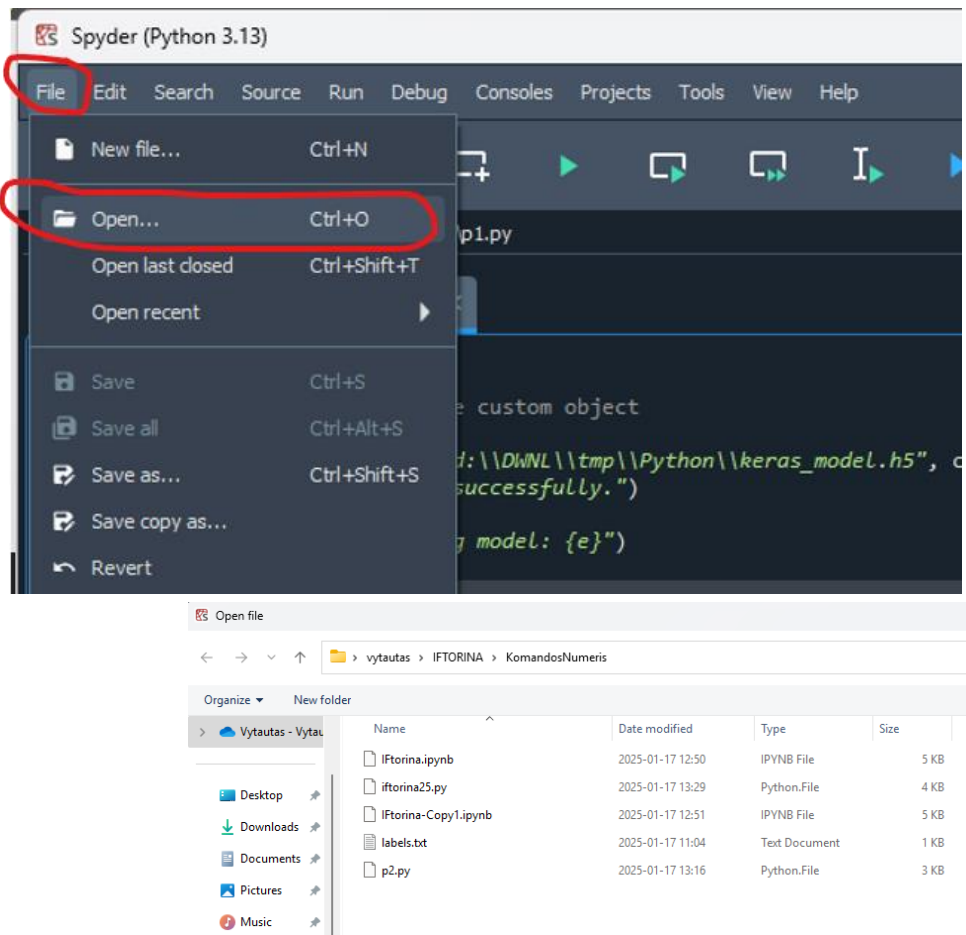
2025-01-23  18:00    <DIR>          .
2025-01-23  16:53    <DIR>          ..
2025-01-23  17:59             5 784 iftorina25.py
2025-01-23  17:33             2 453 432 keras_model.h5
2025-01-23  17:33             30 labels.txt
                3 File(s)          2 459 246 bytes
                2 Dir(s)  177 782 448 128 bytes free

(IFTORINA) C:\Users\VDU\77>spyder

(IFTORINA) C:\Users\VDU\77>
```

ekrane atsiras programavimo grafinės aplinkos `spyder` naujas langas



15. Spyder aplinkoje atsidarykite `iftorina25.py` programą.

16. Pabandykite programą.



Programos vykdymas iškviečiamas mygtuku

The screenshot shows the Spyder Python IDE interface. The left pane displays a Python script named `iftorina25.py` with the following code:

```
1 from keras.models import load_model # Ten
2 from keras.layers import DepthwiseConv2D
3 import cv2 # Install opencv-python
4 import numpy as np
5 import time
6 import random
7 import paho.mqtt.client as mqtt
8 #import ssl
9
10 ##### Naudotojo kintami
11
12 broker = "serverio_adresas" # reikalinga
13 port = 1883 # reikalinga
14 valdiklis="valdiklio_pavadinimas" # reika
15 irenginys="LIGHT"
16 topic = valdiklis+'/' +irenginys
17 client_id = f'python-mqtt-{random.randint(
18
19
20
21 model_dir = "D:/77/"
22 model_name = "keras_model.h5"
23 model_label = "Labels.txt"
24
25
26 # Atpazinto objekto etikete, kuri turi iju
27 atpazinto_objekto_etikete_on = "VYTAUTAS"
28
29 # Atpazinto objekto etikete, kuri turi išj
30 atpazinto_objekto_etikete_off = "V"
31
32 # Publish a message
```

The right pane shows the Variable Explorer with the following variables:

Name	Type	Size	Value
atpazintas_objektas	str	7	AUDRIUS
atpazinto_objekto_etikete_off	str	1	V
atpazinto_objekto_etikete_on	str	8	VYTAUTAS
atpazinto_objekto_nr	int	1	1
broker	str	16	serverio_adresas
camera	VideoCapture	1	VideoCapture object of cv2 module
class_name	str	10	1 AUDRIUS
class_names	list	5	['0 NULL', '1 AUDRIUS']
client	mqtt.client.Client	1	Client object of paho.mqtt.client module
client_id	str	14	python-mqtt-77

The bottom pane shows the IPython Console with the following output:

```
File "D:\77\Iftorina25.py", line 10, in
getaddrinfo(host, port, family, type, proto, flags)
    # We override this function since we want to translate the numeric family
    # and socket type values to enum constants.
    addrlist = []
--> 976 for res in _socket.getaddrinfo(host, port, family, type, proto, flags):
    af, socktype, proto, canonname, sa = res
    addrlist.append((_intenum_converter(af, AddressFamily),
                    _intenum_converter(socktype, SocketKind),
                    proto, canonname, sa))

gallerror: [Errno 11001] getaddrinfo failed

In [10]:
```

Programa kompiliavosi, ir nebuvo įvykdyta, nes joje yra sintaksinių klaidų.

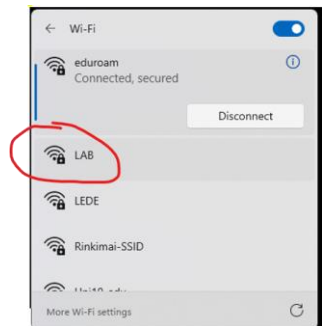
Jūs įsitikinate, kad visi ANTRAME etape diegti įrankiai veikia.

O sintaksines klaidas taisysite TREČIAJAME užduoties etape.

Sveikiname. ANTRĄJĄ užduotį jūs atlikote.

C. TREČIASIS UŽDAVINYS. PROGRAMOS REDAGAVIMAS.

1) įsitikinkite, kad jūsų kompiuteris pajungtas prie **LAB** bevielio tinklo



2. Atlikite programoje pakeitimus.

(ištaisykite sintaksines klaidas, nurodymais teisingus programoje naudojamų kintamųjų parametrus). Šiuos taisymus atlikite 14 – 30 eilutėse.

2.1 `serverio_adresas`, nurodomas tekstu: “192.168.1.244”
(pvz.: `broker = "192.168.1.244"`)

2.2 `porto_numeris` yra skaičius 1883

2.3 `valdiklio_pavadinimas` – nurodomas tekstu.

Priklausomai nuo jūsų komandos numerio, reikia naudoti lentelėje pateiktą valdiklio pavadinimą. Valdiklio pavadinimas turi būti rašomas būtinai tik didžiosiomis raidėmis.

<i>Komandos numeris</i>	<i>Valdiklio pavadinimas</i>
1, 6, 11, 16	“VDU1”
2, 7, 12, 17	“VDU2”
3, 8, 13, 18	“VDU3”
4, 9, 14, 19	“VDU4”
5, 10, 15, 20	“VDU5”

2.4 `irenginys` – nurodomas įrenginio pavadinimas tekstu.
šviesos valdymas vadinamas “LIGHT”

2.5 disko_vieta_kur_yra_modelis - nurodoma disko vieta, disko katalogas iki tos vietos, kurią naudojote ANTROSIOS užduoties vykdymo metu atlikote 7 – 9 žingsniuose. (pvz.: `model_dir = "C:/Users/VDU/77/"` - skaičius 77 - jūsų komandos numeris) atkreipkite dėmesį, į pasvirą ženklą, jis privalo būti tik toks / .

2.6 keras_modelio_pavadinimas – nurodykite laikmenos pavadinimą, su plėtiniu.

2.7 Priklausomai nuo to, kuris objektas įjungia/Išjungia apšvietimą, reikia koreguoti objektų pavadinimus, šiose eilutėse. Objektgo vardą arba pavadinimą nurodykite kurį naudojote apmoant modelį vykdydami PIRMAJĄ užduotį 1 punkte.

objekto_vardas_arba_pavadinimas_1 (šis objektas įjungia apšvietimą)

objekto_vardas_arba_pavadinimas_2 (šis objektas įjungia apšvietimą)

2.8. Patikrinkite ar teisingai atlikti pakeitimai

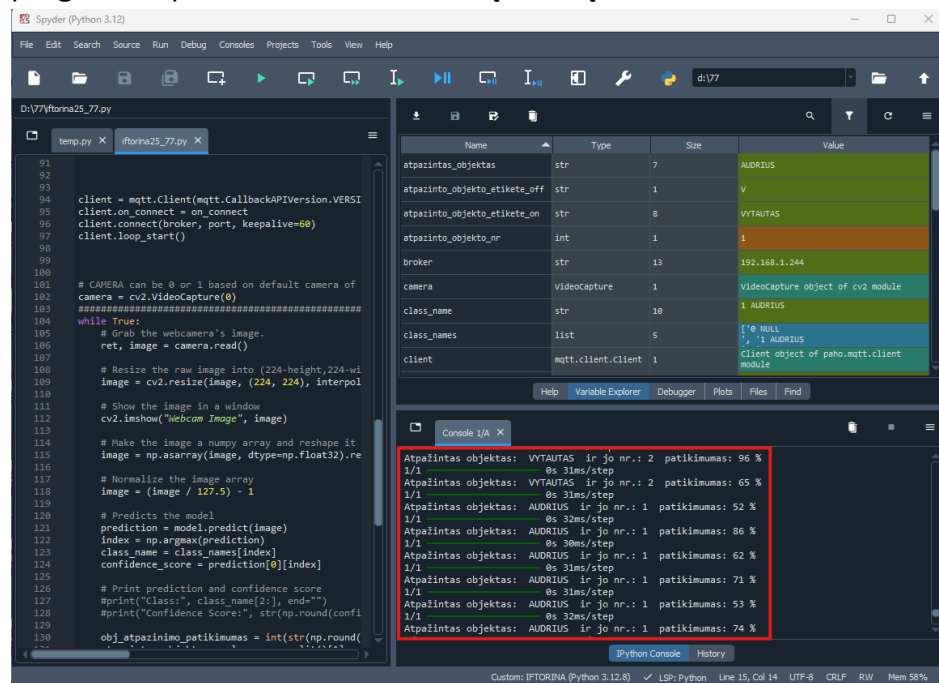
Įvykdykite programą spyder IDE aplinkoje:



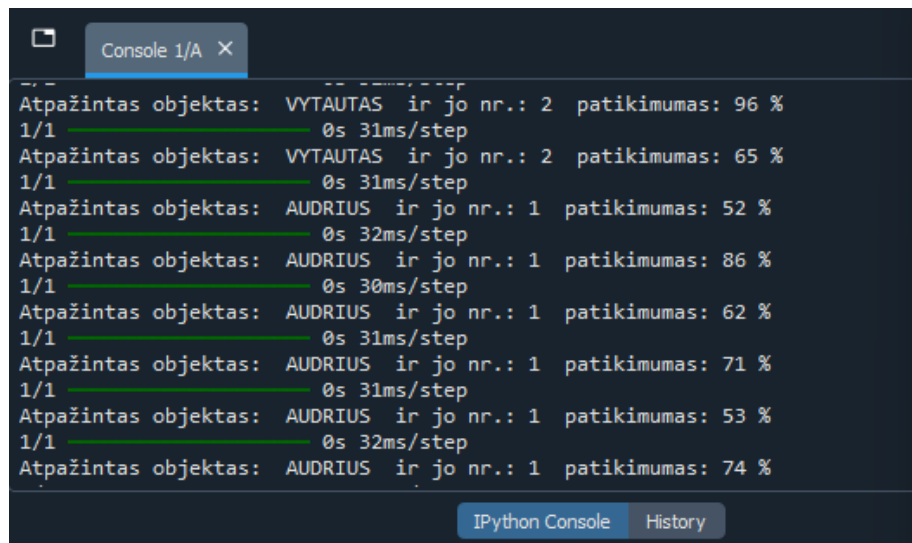
Jeigu kameros vaizdas neatsiranda atskiru langeliu, virš programavimo aplinkos - ieškokite užduočių juostoje , apačioje ekrano, naujai atsiradusios ikonėlės. Ją suaktyvinkite.



Programos veikimo metu, *spyder* aplinkos lange, dešinėje, apačioje, bus rodomas, kaip programa aptinka kameros rodomą vaizdą.



2.9 Analizuodami programos rašomus pranešimus (*spyder* aplinkos lange, dešinėje, apačioje), įvertinkite kaip kamera aptinka Objektus, kuriuos fimavote ir naudojote apmokymui.

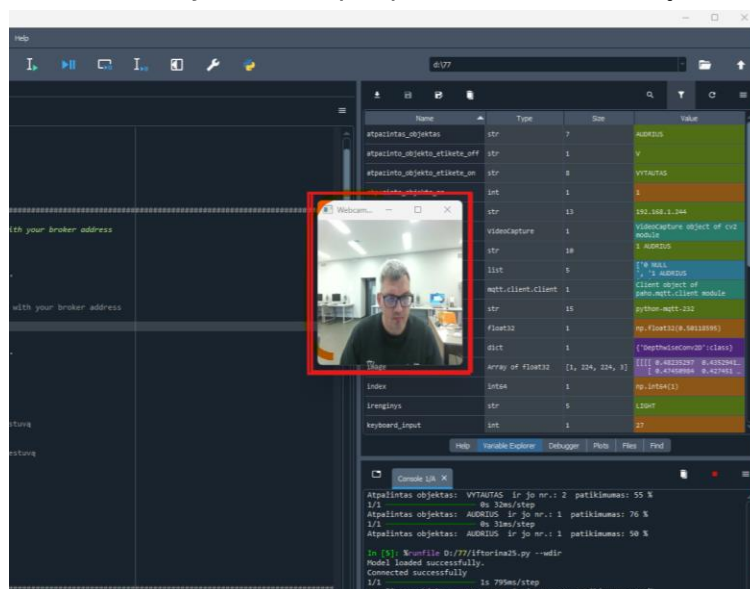


```

-- -- -- -- --
Atpažintas objektas: VYTAUTAS ir jo nr.: 2 patikimumas: 96 %
1/1 ----- 0s 31ms/step
Atpažintas objektas: VYTAUTAS ir jo nr.: 2 patikimumas: 65 %
1/1 ----- 0s 31ms/step
Atpažintas objektas: AUDRIUS ir jo nr.: 1 patikimumas: 52 %
1/1 ----- 0s 32ms/step
Atpažintas objektas: AUDRIUS ir jo nr.: 1 patikimumas: 86 %
1/1 ----- 0s 30ms/step
Atpažintas objektas: AUDRIUS ir jo nr.: 1 patikimumas: 62 %
1/1 ----- 0s 31ms/step
Atpažintas objektas: AUDRIUS ir jo nr.: 1 patikimumas: 71 %
1/1 ----- 0s 31ms/step
Atpažintas objektas: AUDRIUS ir jo nr.: 1 patikimumas: 53 %
1/1 ----- 0s 32ms/step
Atpažintas objektas: AUDRIUS ir jo nr.: 1 patikimumas: 74 %
  
```

2.10 Jeigu jūsų komandai priskirta lemputė nereaguoja į kameros aptinkamus objektus, jums reikia terisingai parinkti ir nurodyti teisingą procentinę objekto atpažinimo patikimumo procentinę vertę. Tai galima atlikti logiškai keičiant `obj_atpazinimo_patikimumas` kintamojo eikšmę (137 eilutė).

Prireikus koreguoti programą - programos sustabdymui užveskite pelytę ant kameros rodomo vaizdo, suaktyvinkite, ir paspauskite ESC klavišą



The screenshot shows the Spyder IDE interface. On the left, there's a code editor with some placeholder text. In the center, a video feed window shows a person's face. On the right, the IPython Console displays the following output:

```

Atpažintas objektas: VYTAUTAS ir jo nr.: 2 patikimumas: 55 %
1/1 ----- 0s 31ms/step
Atpažintas objektas: AUDRIUS ir jo nr.: 1 patikimumas: 76 %
1/1 ----- 0s 31ms/step
Atpažintas objektas: AUDRIUS ir jo nr.: 1 patikimumas: 50 %
  
```

Atlikus užduotį, pakelkite ranką, ir pasigirkite dėstytojui.