

ПРАКТИЧНА РОБОТА 2

МАШИННЕ НАВЧАННЯ ЗАСОБАМИ СЕРВІСУ TEACHABLE MACHINE

Мета: отримати знання та практичні навички у роботі з методами штучного інтелекту ШІ без коду на прикладі сервісу Teachable Machine.

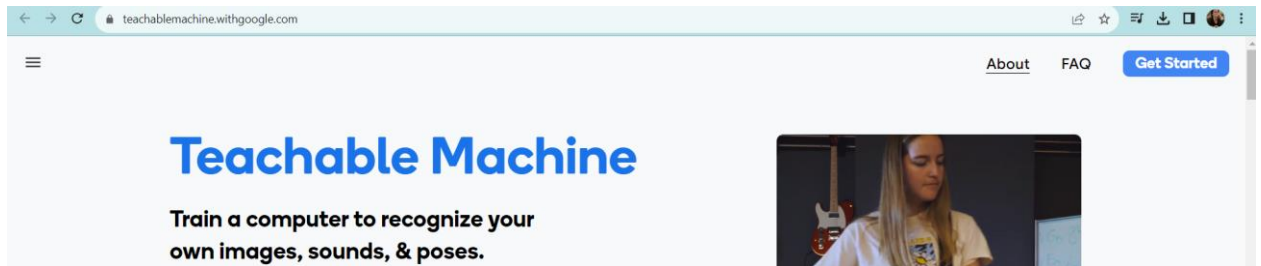
Теоретичний матеріал

Машинне навчання (machine learning) – один з методів функціонування штучного інтелекту, а саме – практичної реалізації його можливостей шляхом створення алгоритмів для виявлення закономірностей під час аналізу великих даних, та їх подальше використання для самонавчання. Останнє – головна особливість і пріоритетне завдання: не вирішити одну конкретну задачу напрямку, а навчитися в процесі застосування рішень виконувати інші подібні завдання. Для цього використовуються математичний та статистичний аналізи, оптимізація та інші техніки опрацювання даних.

Загальну мету machine learning можна охарактеризувати як автоматизацію та оптимізацію рішень складних рутинних задач у найрізноманітніших сферах, від метеорології до комунікацій (про котрі піде мова нижче). Сучасний функціонал інструментів машинного навчання справді вражає: додатки вже здатні розпізнавати мову, жести та образи, проводити медичну та технічну діагностику, біржовий і фінансовий аналізи, систематизувати документацію та виявляти спам. Тотальна діджиталізація призводить до накопичення величезних обсягів інформації у різних галузях, що, в свою чергу, розширює сферу застосування machine learning.

Компанія Google запустила сервіс Teachable Machine, який наочно пояснює принципи машинного навчання. Він дозволяє просто у браузері провести

невелике навчання штучного інтелекту і потім переглянути результат такого навчання.



На сайті Teachable Machine можна продемонструвати системі якийсь об'єкт за допомогою веб-камери комп'ютера. Після цього за допомогою натискання трьох різнокольорових клавіш проводиться процес навчання штучного інтелекту – система розпізнає об'єкти. Для кожної кнопки (кожного класу навчання) потрібно використовувати різні об'єкти. Після навчання сервіс пов'язує розпізнані об'єкти з однією з доступних варіантів представлення. Відповідність може бути представлена у вигляді GIF-анімації, звукових ефектів чи голосового повідомлення. Надалі користувач може змінювати об'єкти перед камерою пристрою, а система визначатиме ці зміни та виводитиме пов'язану з конкретним об'єктом відповідність.

Незважаючи на простоту сервісу Teachable Machine, що здається, він демонструє деякі фундаментальні аспекти машинного навчання.

По-перше, для навчання алгоритмів потрібні приклади, на підставі яких визначаються та запам'ятовуються зразки.

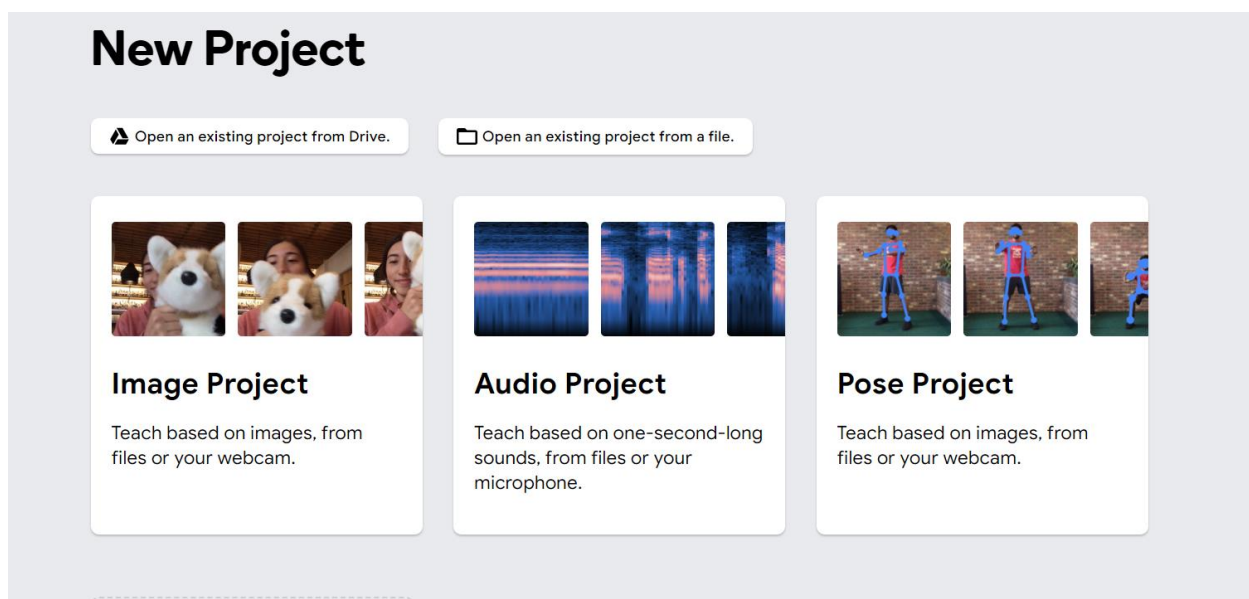
По-друге, на навчання потрібно безліч подібних прикладів.

По-третє, сервіс підвищує рівень розуміння світу алгоритмів штучного інтелекту.

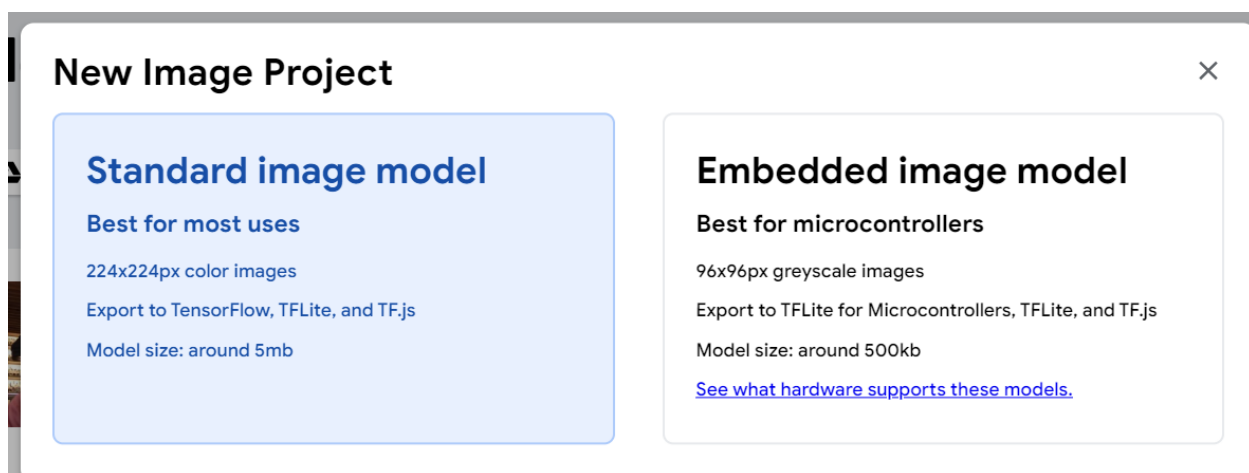
Хід роботи:

1. Переходимо за посиланням <https://teachablemachine.withgoogle.com/>
2. Натискаємо Get Started.

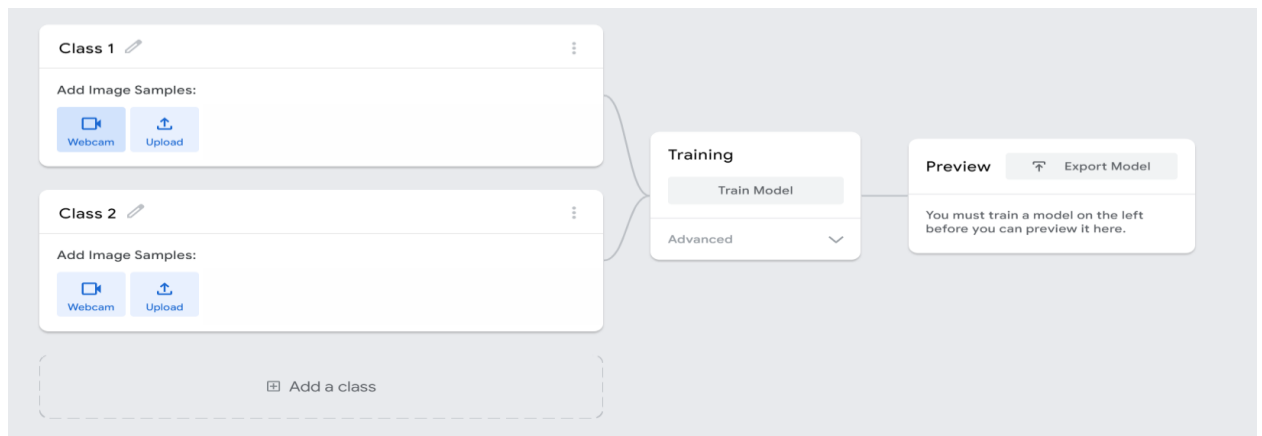
3. Вибираємо тип проєкту.



4. Вибираємо «Стандартна модель зображень».



5. Отримуємо робочу область в якій клас це об'єкт який потрібно розпізнати та відрізнити від іншого об'єкта. Зображення можна завантажувати з диску або веб камери.



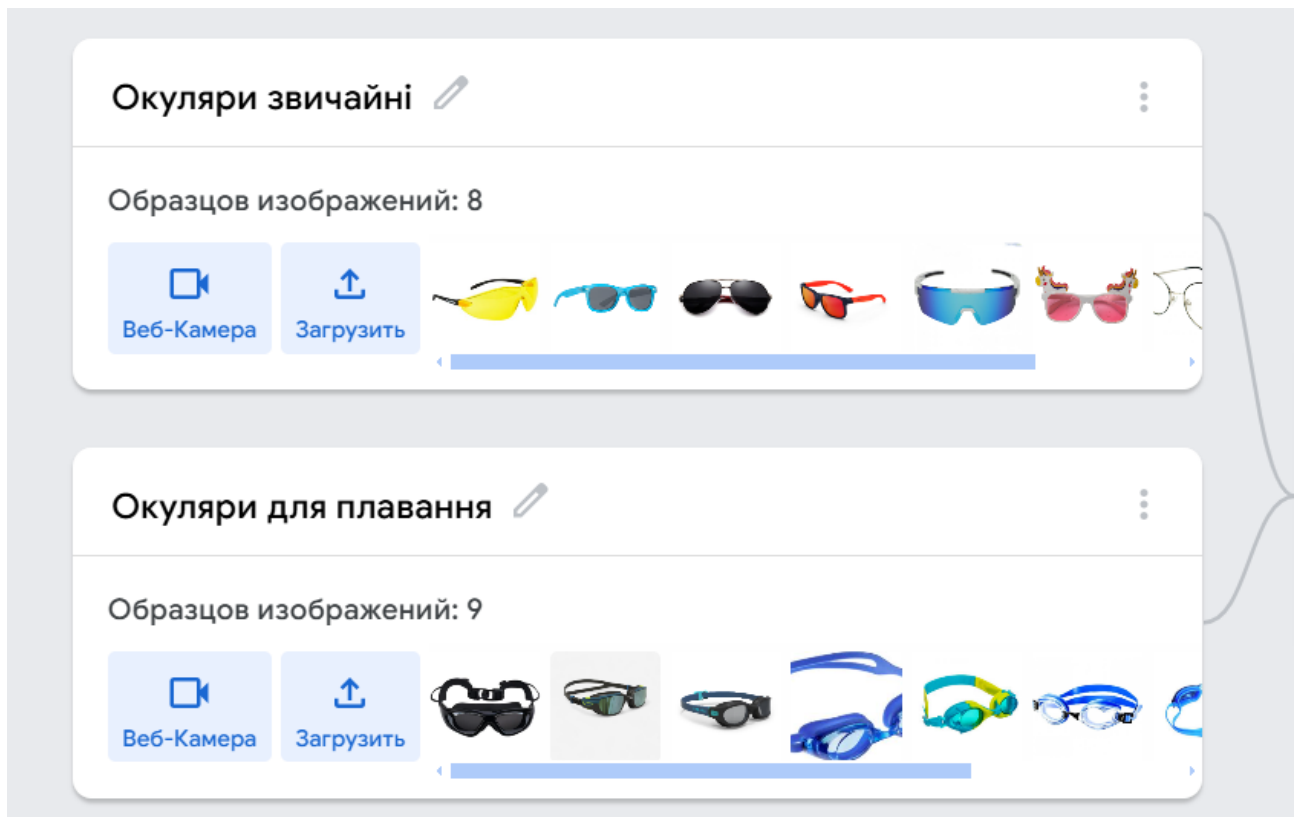
6. Задаємо назви класів.

Наприклад:

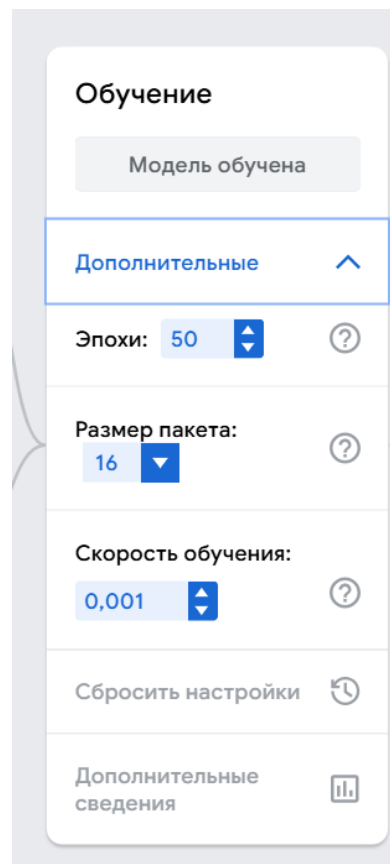
«Окуляри звичайні»

«Окуляри для плавання».

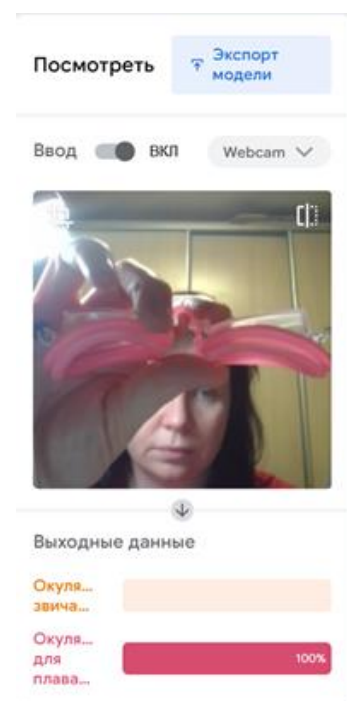
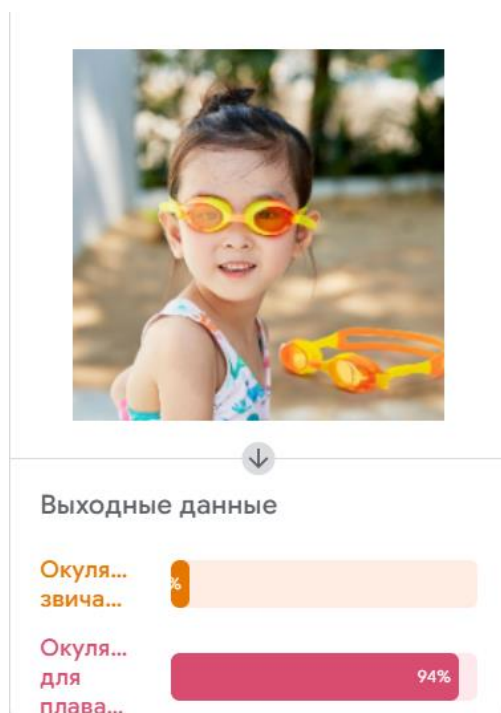
7. Під кожен клас створюємо набір вибірок даних. Налаштовуємо джерела. Кількість вибірок для кожного класу має бути приблизно однаковим. Можна використовувати дзеркальне відображення для збільшення варіантів.

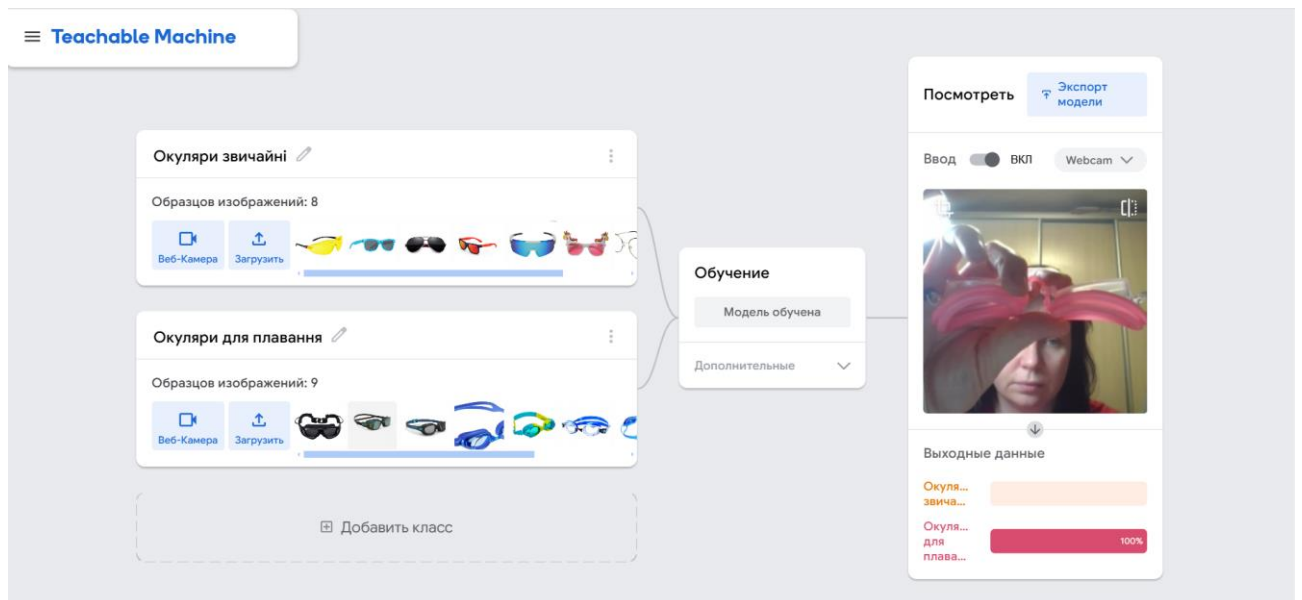


8. Далі нейронну мережу потрібно навчити і переглянути, показуючи їй відповідні об'єкти та оцінюючи результат.



9. Після навчання нейронної мережі її необхідно протестувати використовуючи розділ Preview.





5. Після навчання нейронної мережі її можна забрати використовуючи Export Model.

Export your model to use it in projects. ✕

Tensorflow.js ⓘ **Tensorflow ⓘ** Tensorflow Lite ⓘ

Model conversion type:

☒ Keras ☐ Savedmodel [Download my model](#)

Converts your model to a keras .h5 model. Note the conversion happens in the cloud, but your training data is not being uploaded, only your trained model.

Code snippets to use your model:

Keras OpenCV Keras [Contribute on Github](#) ⓘ

```
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

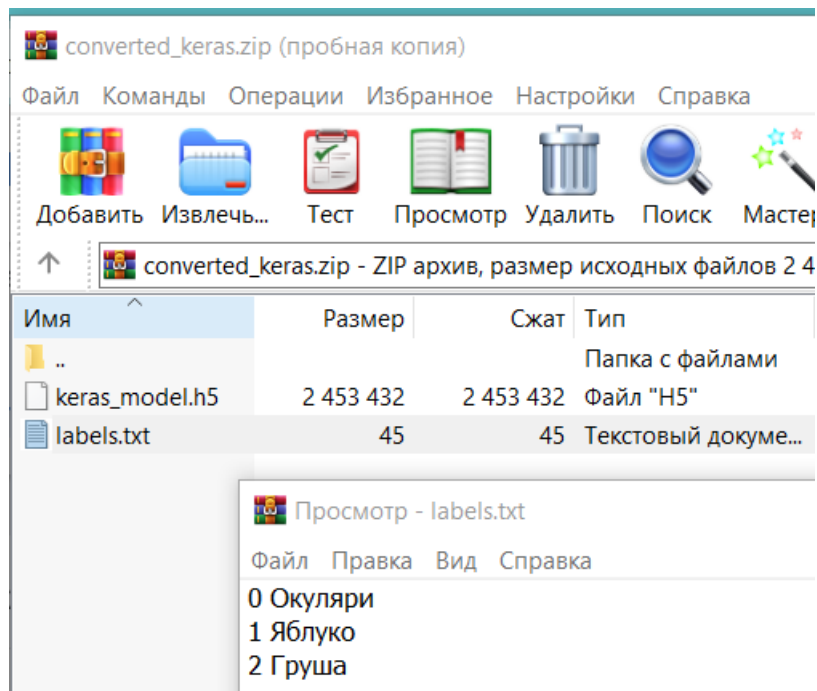
# Disable scientific notation for clarity
np.set_printoptions(suppress=True)

# Load the model
model = load_model("keras_Model.h5", compile=False)

# Load the labels
class_names = open("labels.txt", "r").readlines()

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
```

Copy ⓘ



```
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()
```

```
# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
```

```
# Replace this with the path to your image
image = Image.open("<IMAGE_PATH>").convert("RGB")
```

```
# resizing the image to be at least 224x224 and then cropping from the
center
size = (224, 224)
image = ImageOps.fit(image, size, Image.Resampling.LANCZOS)
```

```

# turn the image into a numpy array
image_array = np.asarray(image)

# Normalize the image
normalized_image_array = (image_array.astype(np.float32) / 127.5) - 1

# Load the image into the array
data[0] = normalized_image_array

# Predicts the model
prediction = model.predict(data)
index = np.argmax(prediction)
class_name = class_names[index]
confidence_score = prediction[0][index]

# Print prediction and confidence score
print("Class:", class_name[2:], end="")
print("Confidence Score:", confidence_score)

```

6. За результатами роботи оформити звіт в якому відобразити етапи роботи ілюструючи їх відповідними скрінами.

Матеріали

1. <https://www.youtube.com/watch?v=bDsR1K53Ew0> відеоінструкція по роботі з сервісом.
2. <https://teachablemachine.withgoogle.com/> посилання на Teachable Machine.