Sharky ESP32CAM AI Masters edukacja AI w liceum

Instrukcja zaprogramowania robota Arduino Sharky omówienie :

1. Każdy podpunkt uczy nowej umiejętności, która ostatecznie przyda się przy stworzeniu w pełni działającego robota na bazie uczenia maszynowego.
2. Treść jest bardzo prosta w wykonaniu, a w razie problemów zostanie umieszczony wypełniony kod w odpowiednich miejscach.
3. Uczeń po wykonaniu zadania powinien umieć obsługiwać się Teachable Machine od Google.

Robot Sharky – „Shurby”

Na zajęciach będziesz programował robota sharky z wbudowaną kamerą ESP32-CAM, Silniczkami kierującymi opony i OLED. Celem zajęć jest stworzenie robota zachowującego się jak robot FURBY. W tym celu uczeń nauczy się obsługi robota sharky tzn. działania silniczków, kamery ESP, wyświetlaczy OLED, zaimplementowanie animacji na kamery OLED, programowanie klas z użyciem narzędzia Teachable Machine od Google. Dzięki tej instrukcji uczeń dowie się jak działają klasy nauczania maszynowego, nauczy się obsługi wielu PIN-ów i narzędzi takich jak kamera, silniczki, ekrany etc.

Wszystkie wykorzystywane kody, aplikację i linki do obsługi robota i Teachable Machine zostaną umieszczone w folderze o nazwie : Shurby_folder.

Wykorzystanie robota Sharky ESP32-CAM w nauce AI

1. Nauka podstaw widzenia komputerowego (Computer Vision)

Dzięki wbudowanej kamerze robot umożliwia:

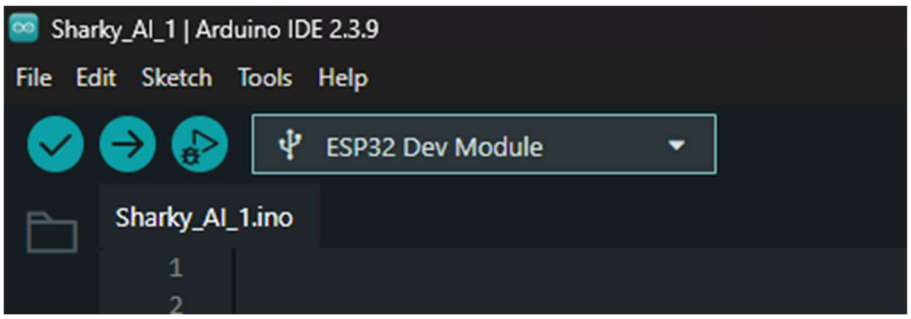
- rozpoznawanie obiektów (np. przeszkód, twarzy, kolorów),
- śledzenie ruchu,
- analizę obrazu w czasie rzeczywistym.

Uczniowie mogą:

- trenować proste modele klasyfikacji obrazów,
- korzystać z gotowych bibliotek (np. OpenCV, TensorFlow Lite),
- obserwować, jak AI „widzi” świat.

1. Podstawy obsługi robota

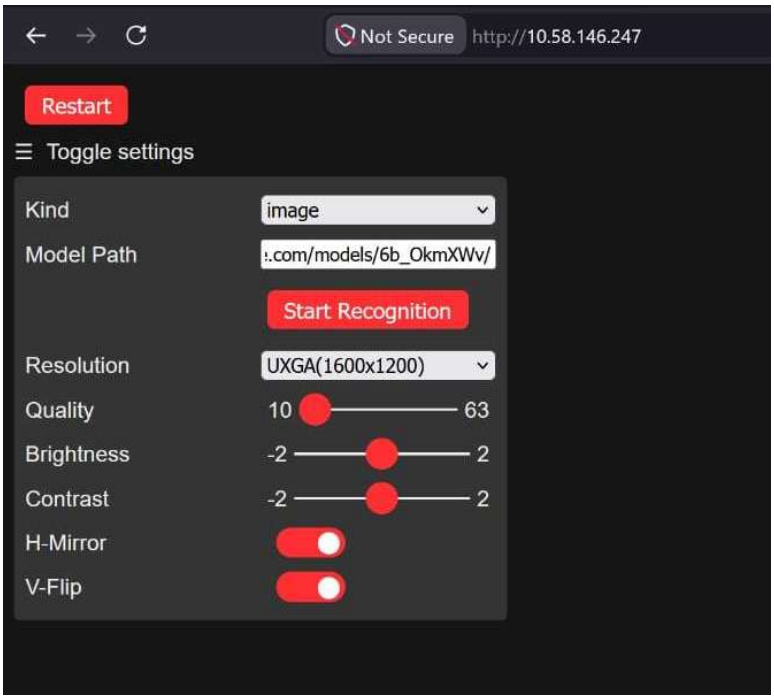
Zanim zaczniemy pracę z robotę upewnij się że masz pobrany program Arduino IDE. Jeśli tak to zaczniemy od otworzenia i wgrania pliku Shurby_AI.ino.



Na początku upewnij się, że masz wybrany moel płytki ESP32 Dev Module oraz poprawny port. W następnym kroku musimy podłączyć naszego do robota do wifi, w tym celu zamień linijki w kodzie poniżej na dane od nauczyciela oraz wgraj kod klikając strzałkę.

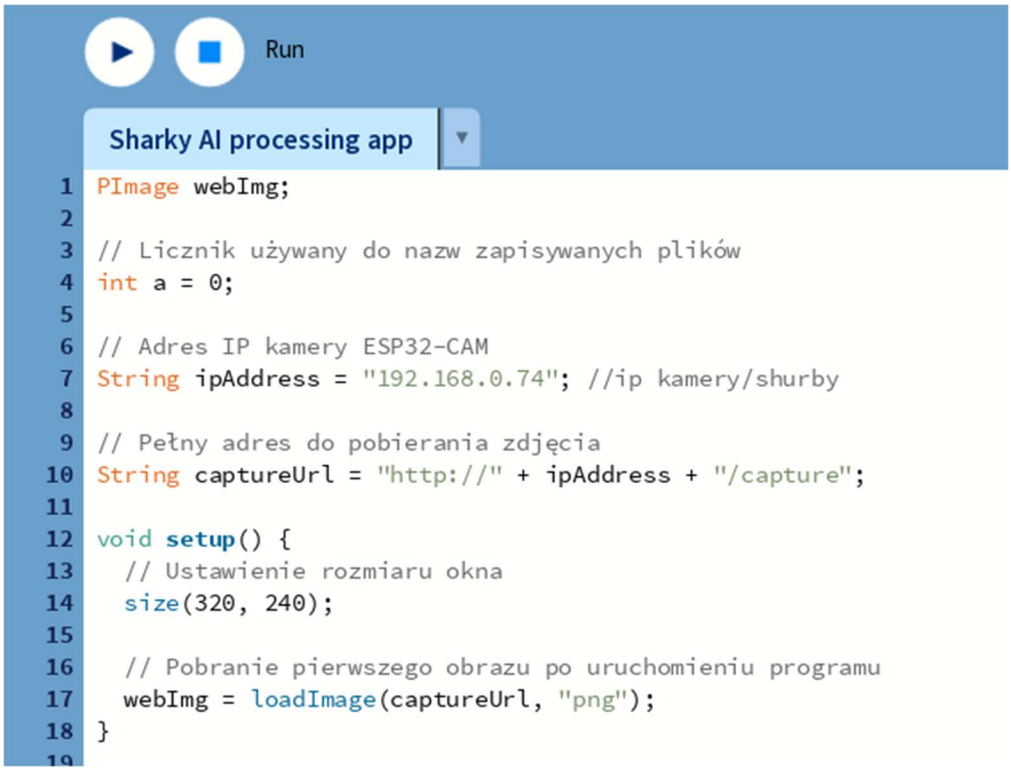
```
16  // ============================================================
17  //  USTAWIENIA SIECI
18  // ============================================================
19  const char* ssid       = "Wifi z telefonu";
20  const char* password   = "haslo123";
```

Po wgraniu kodu na ekanie powinien wyświetlić się adres ip. Połącz urządzenie na którym pracujesz (komputer, laptop, telefon lub tablet) d otej samej sieci wifi do której podłączony został robot, a następnie w przeglądarce wpisz to co wyświetla ekran. Powinieneś zobaczyć coś podobnego do tego.



Wykonywanie zdjęć przy użyciu kamery ESP-32

W celu wykonania zdjęć z widoku kamery, musisz pobrać ze strony processing.org instalator dedykowany twojej wersji komputera, w celu wgrania kodu dostępnego w folderze Shurby_folder pod tytułem „Sharky_AI_processing”. Po pobraniu programu należy otworzyć ten plik za pomocą programu „processing” który przed chwilą pobraliśmy. Po otwarciu powinniśmy zobaczyć to co poniżej.



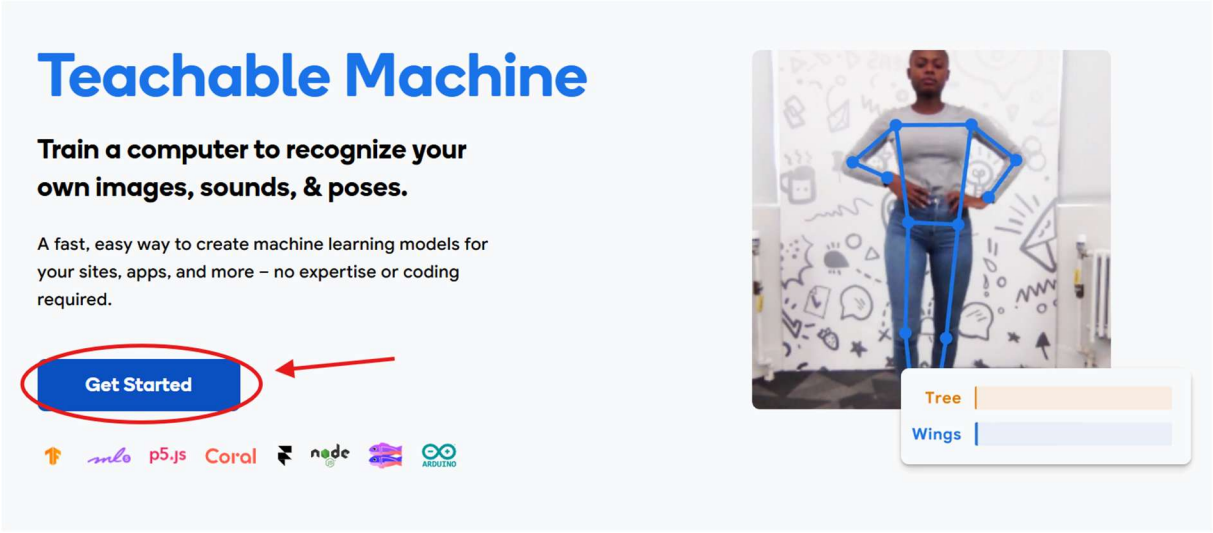
Wiersz 7 to jedyna linijka, która będzie miała dla nas znaczenie. W tym celu wstaw tam numer IP twojego urządzenia, który z góry zostanie wyświetlony na ekranie po podłączeniu do wifi. Kod „Sharky_AI_processing", będzie wyrzucał ci zdjęcia ponumerowane od 0 do momentu, aż zakończysz wykonywanie programu. Podglądowe zdjęcia proponujemy skasować, w celu nie powodowania większych komplikacji i zamieszania.

Tworzenie „klas"

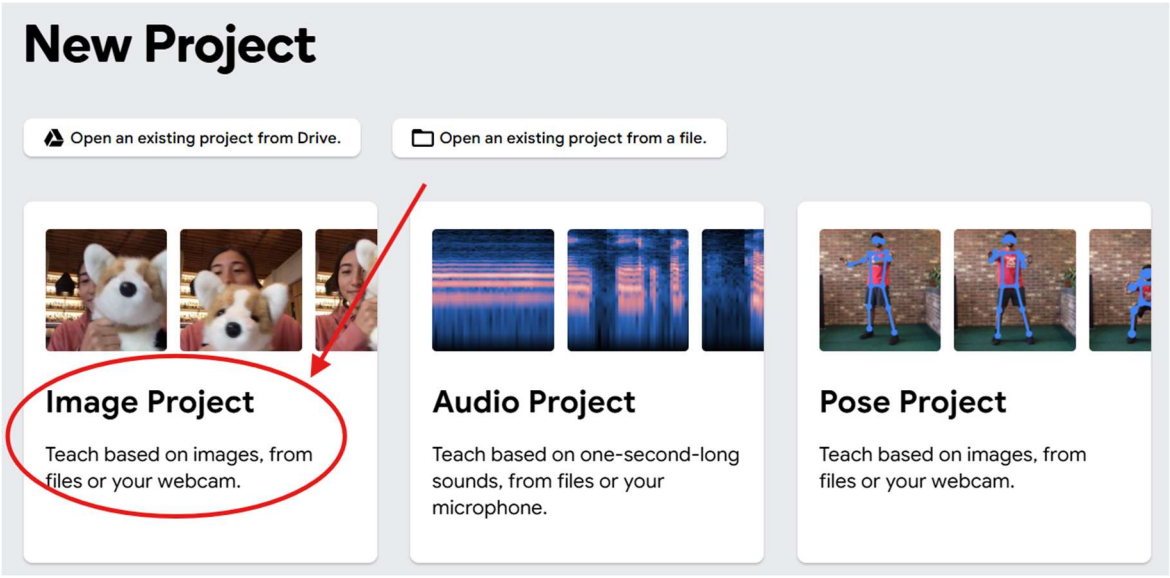
Teraz przejdziemy do definiowania na co nasz robot będzie reagował. Zrobimy to poprzez pokazanie mu 3 różnych grup zdjęć – klas. Stań przed kamerą i odpal program „Sharky_AI_processing" do meomentu aż nie zostanie zrobione ok. 30-40 zdjęć, a następnie zatrzymaj go. Te zdjęcia przenieś do folderu o nazwie „1" i teraz zrób to samo ale nie pokazując się w kamerze robota. Otrzymane zdjęcia przenieś do folderu „2". Ostatnia klasa jest dowolna – pokaż do kamery jakąś rzecz po której „zobaczeniu" chcesz żeby robot reagował i przenieś otrzymane zdjęcia do folderu „3".

Praca z Techable machine

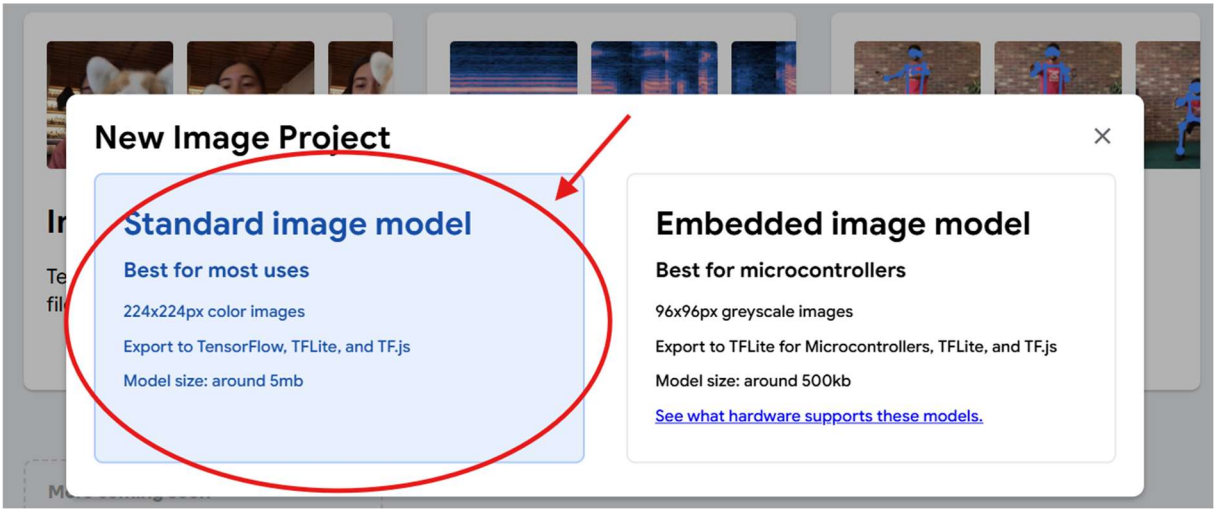
Zaczniemy pracę z narzędziem teachable machine. Aby przenieść się na stronę wklej link https://teachablemachine.withgoogle.com , kliknij get started, utwórz image project -> standard image project.



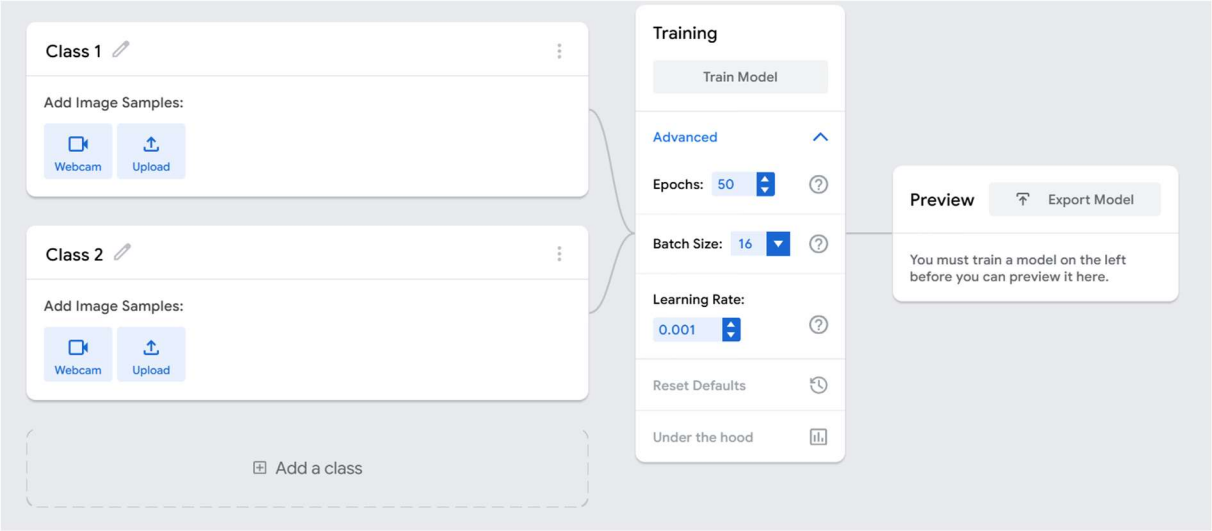
Krok 1



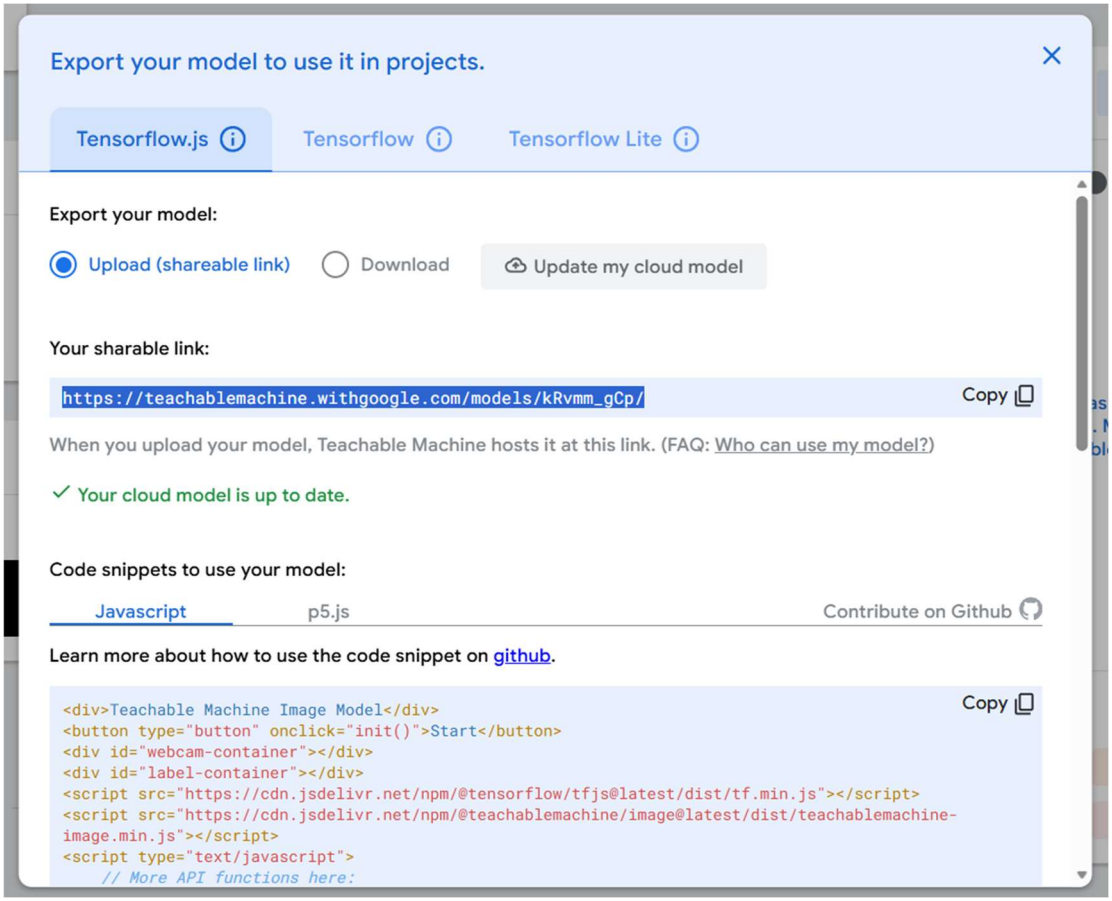
Krok 2



Krok 3

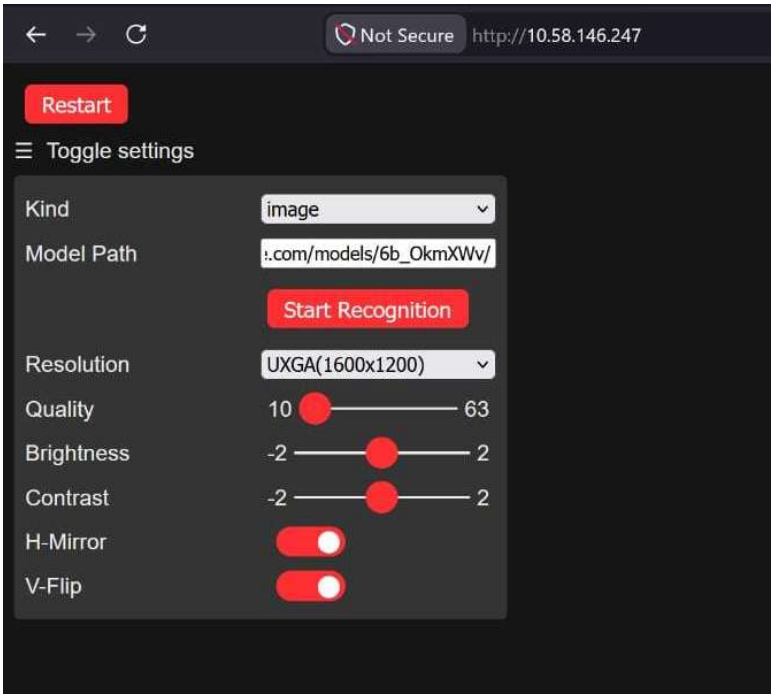


Ostatecznie powinieneś otworzyć okno widoczne na zdjęciu powyżej. Teraz będziemy tworzyć projekt dla 3 głównych klas. Zacznijmy od stworzenia wgrania naszych klas. W tym celu załadujemy nasze poszczególne foldery („1", „2" i „3") do odpowiednich miejsc na stronie Techable machine klikając „upload" i przeciągając zawartość naszych folderów. Następnie przechodzimy wytrenowania naszego modelu klikając „Train Model" oraz do wyeksportowania naszego projektu. Klikając przycisk „Export model" wyświetli nam się taka strona :



Musisz wtedy kliknąć w ikonę „Upload my model”, dzięki czemu utworzy się cały model i oraz link do niego, który będzie nam bardzo przydatny. Kolejnym krokiem jest „umieszczenie” tego linka w naszym robocie. W tym celu musisz otworzyć IP swojego robota, połączonego tą samą siecią, w wyszukiwarce.

Ostatnie zdjęcie w instrukcji pokazuje panel uruchamiany po wpisaniu adresu IP robota w przeglądarce. Jest to końcowy etap konfiguracji modelu AI. Na widocznym, ciemnym ekranie znajduje się miejsce na wpisanie lub wklejenie linku do wytrenowanego modelu, czyli pole **Model Path**, oraz przycisk **Start Recognition**. To właśnie tutaj użytkownik podłącza przygotowany wcześniej model z Teachable Machine do robota.



Efektem końcowym będzie robot, który rozpoznaje obraz z kamery ESP32-CAM za pomocą modelu stworzonego w Teachable Machine. Model będzie składał się z **3 klas** - po ich wykryciu robot wykonuje konkretną akcję, np. zmienia animację na ekranach OLED, porusza się lub zatrzymuje, robi zdjęcia itp.

Kod robota będzie działał modularnie, czyli osobne fragmenty programu będą odpowiadały za kamerę, silniki, ekrany OLED, animacje oraz reakcje na klasy AI. Dzięki temu łatwo będzie później dodać nowe zachowania albo zmienić istniejące reakcje bez przebudowy całego kodu.

Mamy już nasze klasy to teraz przejdźmy do reakcji naszego robota!

W pliku „akcje.txt” znajdują się opisane „bloki” kodu które reprezentują to co ma robić nasz Shurby w zależności od tego co widzi kamera. W naszym kodzie Arduino w linijce 135 możemy zobaczyć coś takiego:

```
135  void applyEmotionTargets(EmotionType emotion) {
136    switch (emotion) {
137      case NEUTRAL:
138      case 1:
139        targetBrowThickTop = 4;  targetBrowThickBot = 11; targetBrowPoint = 8;
140        break;
141      case 2:
142        targetBrowThickTop = 2;  targetBrowThickBot = 10; targetBrowPoint = 19;
143        break;
144      case 3:
145        targetBrowThickTop = 13; targetBrowThickBot = 20; targetBrowPoint = 3;
146        break;
147    }
148  }
```

Pod każdym z 3 „case” (które odzwierciedlają nasze klasy które utworzyliśmy wcześniej) możemy teraz wkleić „bloki” z pliku „akcje.txt”. Np. jeżeli chcemy żeby robot kręcił się kiedy nas widzi to wybieramy blok „kręcenie” i umieszczamy go w „case” 1, a jeżeli chcemy żeby pokazywał animacje smutku kiedy nas nie „widzi” to dla „case” 2 wklejamy „blok” „smutek”. Możesz śmiało testować różne kombinajce jak dokładnie Twój robot ma reagować – **miłej zabawy!**