



Izazovi i mogućnosti novog doba
- priručnik za učenike starijeg uzrasta

ARTIE



Sufinancirano sredstvima
programa Europske unije
Erasmus+

ARTIE: Umjetna inteligencija u obrazovanju - izazovi i prilike novog doba: razvoj novog kurikuluma, vodiča za učitelje i online
tečaja za učenike
Sufinancirano sredstvima programa Europske unije Erasmus+
2020-1-HR01-KA201-077800

ARTIE

Izazovi i mogućnosti novog doba: priručnik za poučavanje učenika starijeg uzrasta

Autori

Željko Krnjajić
Janko Radigović

Izdavač

Hrvatski robotički savez, Croatia

"Artificial Intelligence in Education - challenges and opportunities of the new era: development of curriculum, guides for teachers and e-courses for students", project number 2020-1-HR01-KA201-077800 under Erasmus+ Programme.

Suradnici

Katarzyna Garbacik
Andrzej Garbacik
Bogusław Klimczuk
Ivana Ružić
Ana Pina
Christina Eirini Karvouna

Grafički dizajn i ilustracije

Christina Eirini Karvouna

Prevoditelji

Jura Cmrečak (hrvatski jezik)
Bogusława Denys (poljski jezik)
Ana Pina (portugalski jezik)
Christina Eirini Karvouna (nizozemski jezik)



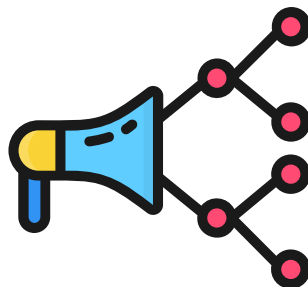
11	3	Umjetna inteligencija u robotici
		Detekcija i prepoznavanje lica u Scratchu
26	19	Programiranje detekcije lica u Scratchu
		Programiranje prepoznavanja lica u Scratchu
	41	Detekcija i klasifikacija objekata za početnike
49		Programiranje detekcije objekata u Scratchu
	57	Projekt s klasifikacijom objekata
63		Prepoznavanje i generiranje govora za početnike u Scratchu
	68	Programiranje prepoznavanja govora u Scratchu
73		Programiranje generiranja govora u Scratchu
	77	Projekt glasovnog upravljanja objektom
83		Uvod u hardware - mikrokontroler, kamera i upravljač motora
	93	Povezivanje mikrokontrolera s kamerom i računalom, početak rada
100		Sastavljanje robota
	119	Programiranje kretanja robota
128		Programiranje robota - praćenje objekata
	140	Programiranje robota - praćenje linije



Umjetna inteligencija u robotici

Može li umjetna inteligencija u robotici promijeniti budućnost?
Što je umjetna inteligencija u robotici?
Koristimo li već robote s umjetnom inteligencijom u svakodnevnom životu?
Kako? Gdje?

U nastavku ćemo objasniti što je umjetna inteligencija i promišljati o njenoj trenutnoj i budućoj primjeni u robotici.



Sufinancirano sredstvima
programa Europske unije
Erasmus+



<http://erasmus-artie.eu>



Teme za raspravu:

Što je robot?

Koristi li svaki robot umjetnu inteligenciju?

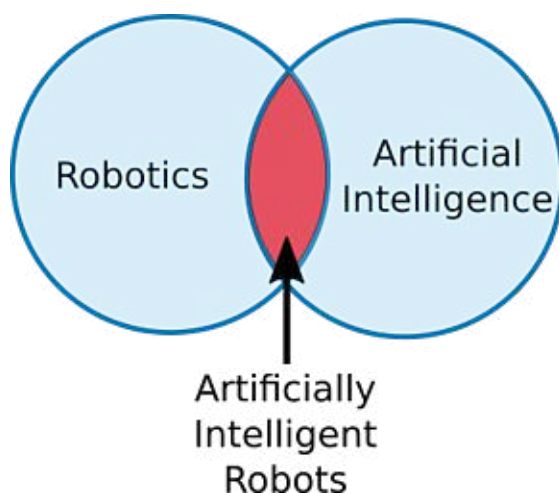
Znate li neke primjere umjetne inteligencije u robotici?

Mogu li roboti s umjetnom inteligencijom zamijeniti ljude?

Jesu li robotika i umjetna inteligencija ista stvar?

Prvo što treba pojasniti je da robotika i umjetna inteligencija uopće nisu iste stvari. Zapravo, ta dva polja su gotovo potpuno odvojena..

Venovim dijagramom prikazujemo područja umjetne inteligencije, robotike i područje umjetno inteligentnih robota:



Postoji malo područje gdje se ta dva područja preklapaju: Umjetno inteligentni roboti. Samo preklapanje tih dvaju područja ponekad djeluje zbunjujuće.

Što je umjetna inteligencija?

Umjetna inteligencija (UI) je grana informatike. Ona uključuje razvoj računalnih programa za rješavanje zadataka koji bi inače zahtijevali ljudsku inteligenciju. Algoritmi umjetne inteligencije mogu se baviti učenjem, percepcijom, rješavanjem problema, razumijevanjem jezika i/ili logičkim rasuđivanjem.

Prema ocu umjetne inteligencije, Johnu McCarthyju, također je to i "Znanost i inženjerstvo izrade inteligentnih strojeva, posebno inteligentnih računalnih programa".



Jednostavno rečeno, umjetna inteligencija je način **na koji računalno, računalno upravljani robot ili program inteligentno razmišljaju**, na sličan način na koji inteligentni ljudi misle.

Ciljevi umjetne inteligencije

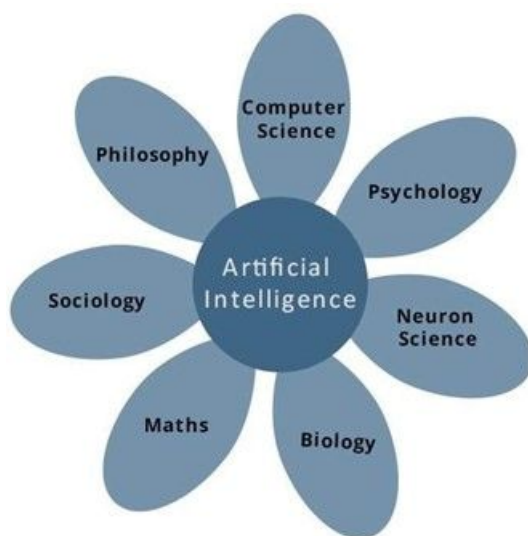
stvaranje sustava – ovi sustavi pokazuju inteligentno ponašanje, uče, pokazuju, objašnjavaju i savjetuju svoje korisnike,

primjena ljudske inteligencije u strojevima – takvi bi sustavi razumjeli, razmišljali, učili i ponašali se kao ljudi.

Što pridonosi umjetnoj inteligenciji?

Umjetna inteligencija je znanost i tehnologija koja se temelji na disciplinama kao što su informatika, biologija, psihologija, lingvistika, matematika i inženjerstvo. Glavni poticaj umjetnoj inteligenciji daje razvoj računalnih funkcija povezanih s ljudskom inteligencijom, kao što su rasuđivanje, učenje i rješavanje problema.

Grafički prikaz područja koja koristi područje umjetne inteligencije:



Što je robotika?

Robotika je grana umjetne inteligencije koja se sastoji od elektrotehnike, strojarstva i računalnih znanosti za projektiranje, konstrukciju i primjenu robota.

Što su roboti?

Roboti su umjetni agenti koji djeluju u stvarnom okruženju..

To su programabilni strojevi koji obično mogu samostalno ili poluautonomno obavljati niz radnji.



Cilj

Roboti su razvijeni tako da imaju sposobnost manipuliranja objektima opažanjem, skupljanjem, kretanjem, modificiranjem fizičkih svojstava objekta, uništavanjem ili da oslobađaju radnu snagu od obavljanja ponavljajućih zadataka bez dosade, ometanja ili iscrpljenosti.

Aspekti robotike

Roboti imaju **mehaničku konstrukciju**, oblik ili oblik dizajniran za izvršavanje određenog zadatka.

Imaju **električne komponente** koje napajaju i kontroliraju strojeve.

Sadrže neki oblik **računalnog programa** koji određuje što, kada i kako robot nešto radi.

Umjetna inteligencija u robotici

Umjetna inteligencija u robotici pomaže robotima u obavljanju ključnih zadataka s ljudskom vizijom za otkrivanje ili prepoznavanje različitih objekata. Roboti se razvijaju kroz strojno učenje i obuku, a ogroman broj skupova podataka koristi se za obuku modela računalnog vida, tako da roboti mogu prepoznati različite objekte, izvršiti radnje u skladu s tim i ostvariti željene ishode. Računalni vid je jednostavno proces opažanja slika i videozapisa dostupnih u digitalnim formatima. Umjetna inteligencija u robotici ne samo da pomaže naučiti model da obavlja određene zadatke, već i strojeve čini inteligentnijima i stoga sposobnijima djelovati u različitim scenarijima.

Evo nekoliko primjera najnaprednijih humanoidnih, industrijskih i uslužnih robota koji mijenjaju budućnost uz pomoć umjetne inteligencije.

Sophia

Najnapredniji robot iz tvrtke Hanson robotics izgleda poput čovjeka, zove se Sophia i utjelovljuje naše snove za budućnost umjetne inteligencije.

Kao jedinstvena kombinacija znanosti, inženjerstva i umjetnosti, Sophia je istovremeno znanstvenofantastični lik izrađen od strane čovjeka koji prikazuje budućnost umjetne inteligencije i robotike te platforma za naprednu robotiku i istraživanje umjetne inteligencije.

Lik Sofije zaokuplja maštu publike diljem svijeta.

Ona je prvi robot građanin na svijetu i prva robotska ambasadorica inovacija Programa Ujedinjenih naroda za razvoj (UNDP). Sophia je danas poznato ime, s nastupima u Tonight Showu i Good Morning Britain, a pored toga govori na stotinama konferencija širom svijeta.

Upoznaj Sophiu: <https://www.youtube.com/watch?v=BhU9hOo5Cuc>





Digit

Digit je zamišljen kako bi pomogao u brizi za ljude u njihovim domovima te pomogao u reakciji na katastrofe i dostavljao pakete ulaznim vratima. Sa svojim pokretnim ekstremitetima i torzom prepunim senzora, Digit se može kretati u složenim okruženjima i obavljati zadatke poput isporuke paketa. U svibnju 2019. godine tvrtke Ford Motor Company i Agility najavili su partnerstvo za razvoj logističkog rješenja u zadnjoj fazi dostave koje kombinira Fordovu tehnologiju autonomnih vozila i Agilityjevog robota Digit.



Pepper

Pepper je prvi društveni humanoidni robot na svijetu koji je u stanju prepoznati lica i osnovne ljudske emocije. Pepper je prihvatilo više od 2000 tvrtki širom svijeta. Savršen je u maloprodajnoj i financijskoj industriji, Pepper gdje ima brojne funkcionalnosti, uključujući povećanje prometa u trgovinama privlačenjem pažnje kupaca, stvaranjem nezaboravnih iskustava u trgovini, poticanjem kupnje te zadržavanjem kupaca. Pepper također može prikupiti sveobuhvatne podatke kako bi obogatio bazu podataka o kupcima i davao uvid u njihovo ponašanje.



<http://erasmus-artie.eu>

Atlas

Atlas je najdinamičniji humanoidni robot na svijetu koji je izgradio BostonDynamics, tvrtka koja je prije bila u vlasništvu Googlea, a sada SoftBanka. Atlas iz godine u godinu postaje sve sofisticiraniji zahvaljujući najsuvremenijem hardveru i algoritmima koji mu omogućuju brzo razumijevanje uputa. Sa svojih 28 hidrauličnih aktuatora, visinom od 1,5 m i težinom od oko 80 kg, robot može izvesti i impresivne i istovremeno zastrašujuće radnje, uključujući kretanje po neravnom terenu, skakanje po parkour stazi a može čak i raditi salta. Sve ove karakteristike pokazuju agilnost na ljudskoj razini i kako bi robot mogao biti savršena opcija za operacije traganja i spašavanja te obavljanje ljudskih zadataka u okruženjima u kojima ljudi ne mogu preživjeti.





Spot

Spot je robotski pas dizajniran za industrijsku uporabu kao što je prijevoz robe u skladištu te pregled udaljenog mjesta s nepovoljnim okruženjem za ljudske operatere. Može se kretati brzinom od 1,5 metara u sekundi, ima kameru s vidnim poljem od 360 stupnjeva i može raditi na temperaturama od -15 do 45 stupnjeva Celsiusa. Sa svojim API-jem i fleksibilnim sučeljem korisnog tereta, robot se može lako prilagoditi željenim zadacima. Spot također proizvodi BostonDynamics i sada se iznajmljuje prihvatljivim tvrtkama.



HRP-5P

HRP-5P je napredni humanoidni robot dizajniran za autonomni rad i obavljanje teškog rada u opasnim okruženjima. Opremljen je senzorima okoliša i prepoznavanjem objekata, planiranjem i kontrolom pokreta cijelog tijela te upravljanjem zadacima i izvršavanjem. Rad na HRP-5P obuhvaća više od 20 godina humanoidnog istraživanja u AIST-u. U tih 20 godina institut je stvorio još 4 robota koji su prethodnici HRP-5P.



<http://erasmus-artie.eu>

Surena IV

Surena IV četvrta je generacija serije humanoidnih robota Surena koju je razvilo Sveučilište u Teheranu u Iranu. S visinom od 1,7 m i težinom od 154 kg, ovaj robot može hodati brzinom od 0.7 km/h. Njegovi prilagođeni senzori sile na dnu stopala pomažu robotu da prijeđe neravne površine podešavanjem kuta i položaja svake noge.





Aquanaut

Aquanaut je napredno bespilotno podvodno vozilo koja se može transformirati iz okretne podmornice za velike udaljenosti u polu-humanoidnog robota sposobnog za obavljanje složenih zadataka podvodne manipulacije. Dizajniran od strane Houston Mechatronics Inc, Aquanaut može pregledati podmorsku naftnu i plinsku infrastrukturu, upravljati ventilima i koristiti podmorske alate sa samo nekoliko klikova mišem. Djelujući potpuno nevezano i bez pomoćnih brodova, Aquanaut može putovati više od 200 km u podmorskom načinu rada, ima maksimalnu brzinu od 7 čvorova i maksimalnu operativnu dubinu od 300 m.



f

Stuntronic robot

Robot Stuntronic je kaskader dizajniran za zabavu publike u Disneyjevim tematskim parkovima i odmaralištima. Sa svojim ugrađenim sofisticiranim senzorima može donositi vlastite odluke u stvarnom vremenu - sve to dok leti na 60 stopa gore u zraku. Zna kada skupiti koljena da bi izveo salto, kada povući i saviti ruke pa čak i kako usporiti okretanje kako bi osigurao savršen doskok.



<http://erasmus-artie.eu>

Handle

Handle je još jedan robot iz Boston Dynamicsa. Sa svojim softverom za viziju dubokog učenja, ovaj robot može identificirati i locirati kutije, istovariti kamione, paletizirati i depaletizirati teret pritiskom na gumb. Njegova mobilnost omogućuje mu rad u više radnih okruženja, krećući se kroz objekte paralelno s transportom robe. Može pokupiti i do 360 kutija na sat.





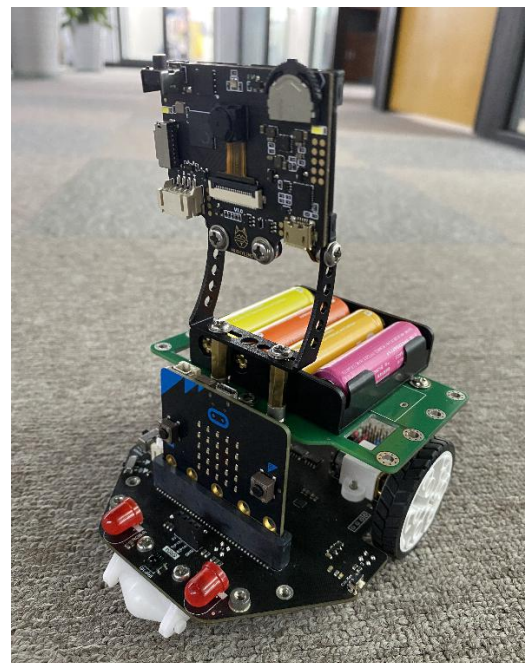
Danas koristimo umjetnu inteligenciju u robotici u zdravstvu, poljoprivredi, automobilske industriji, u skladištima, u lancu opskrbe itd.

Kasnije ćemo sastaviti vlastitog robota s umjetnom inteligencijom i osposobiti ga za otkrivanje i prepoznavanje lica, otkrivanje objekata i prepoznavanje govora.

Ovo je mobilni robot s kamerom i UI mogućnostima koje ćemo koristiti za ovaj projekt.

Pokazati ćemo vam kako ga napraviti i koristiti za:

- Detekciju lica
- Prepoznavanje lica
- Praćenje lica
- Detekciju objekata
- Praćenje objekata
- Praćenje crte



Umjetna inteligencija je konačno ovdje i većina nas je već aktivno koristi u svakodnevnom životu (čak i bez znanja). Buduće generacije trebaju razumjeti kako to koristiti AI prije svega! Tek tada ga mogu koristiti za pomoć pri učenju i rješavanje problema iz stvarnog svijeta.

Umjetna inteligencija (AI) i robotika danas su snažno povezane.

Umjetna inteligencija u robotici sve se više koristi u svakodnevnom životu i ključna je u različitim područjima kao što su industrija, vojne operacije, medicina, istraživanje, zabava.

Zapamtite, umjetna inteligencija je vjerojatno najmoćnija tehnologija koju je čovjek ikada izumio. Može se koristiti i za dobre i za loše stvari. Na kraju, na nama je kako ćemo ju koristiti.

ZAKLJUČAK

Umjetna inteligencija je način na koji računalo, računalno upravljani robot ili program inteligentno razmišljaju, na sličan način na koji inteligentni ljudi misle.

Promisli!



Što sada znam?



Što sam naučio/la?



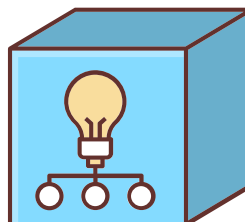
Što još želim znati?



Detekcija i prepoznavanje lica u Scratchu

Pojmovi detekcija lica i prepoznavanje lica ponekad se koriste naizmjenično, ali postoje neke ključne razlike. Da bismo lakše razlikovali to dvoje, razmotrimo prvo pojam detekcije lica i kako se on razlikuje od pojma prepoznavanje lica.

U nastavku ćemo upoznati otkrivanje i prepoznavanje lica za početnike kroz primjere različitih aplikacija.



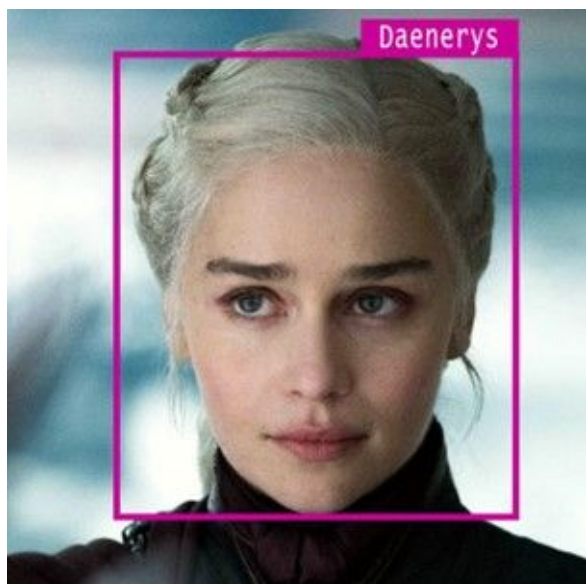
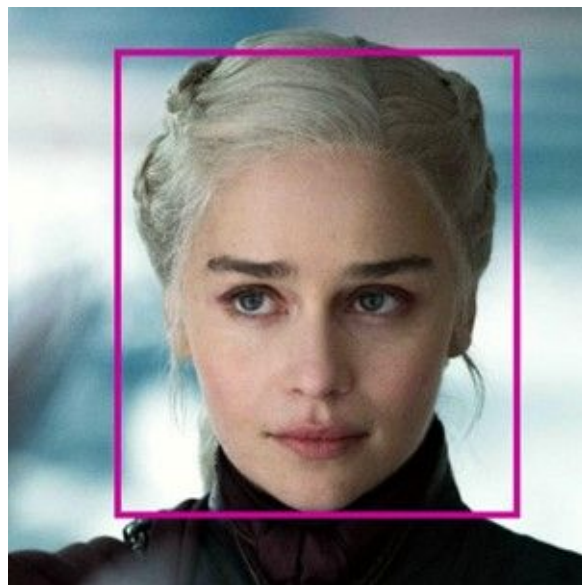
Sufinancirano sredstvima
programa Europske unije
Erasmus+



<http://erasmus-artie.eu>



Detekcija lica je proces koji određuje prisutnost lica na slici ili u videoisječku. Na primjer, ta funkcija je dostupna u većini aplikacija za kamere pametnih telefona. Modul za detekciju ne određuje čije je lice u okviru.



U postupku detekcije lica ne pamte se niti spremaju crte lica. Ako program otkrije lice neke određene osobe u okviru i kasnije pronađe isto lice na drugoj slici, neće odrediti da lice pripada istoj osobi, samo će otkriti prisutnost lica u okvirima. Program može pružiti podatke o dobi i spolu osobe na svakom okviru, ali ne više od toga. Softver za otkrivanje lica ne može prepoznati određene osobe.

Nasuprot tome, prepoznavanje lica odnosi se na identificiranje i prepoznavanje osoba. Svrha algoritma za prepoznavanja lica je izvršiti identifikaciju osoba koje se pojavljuju na slici ili videoisječku uspoređujući je s bazom podataka. Kako bi se osigurala uspješna identifikacija, odgovarajući podatci za lice prvo se moraju unijeti u bazu podataka. Program određuje jedinstvene značajke lica, sprema ih i koristi za naknadnu identifikaciju. Kasnije, tijekom postupka identifikacije, algoritam će usporediti jedinstvene značajke i identificirati lice određene osobe u slučaju da se te značajke podudaraju.



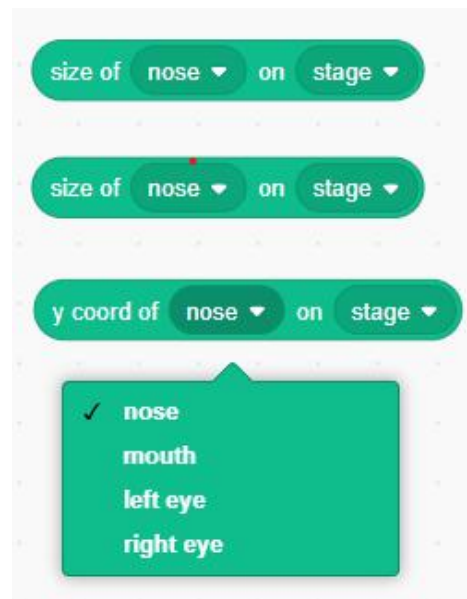


Scratch i druge aplikacije za korištenje

Scratch

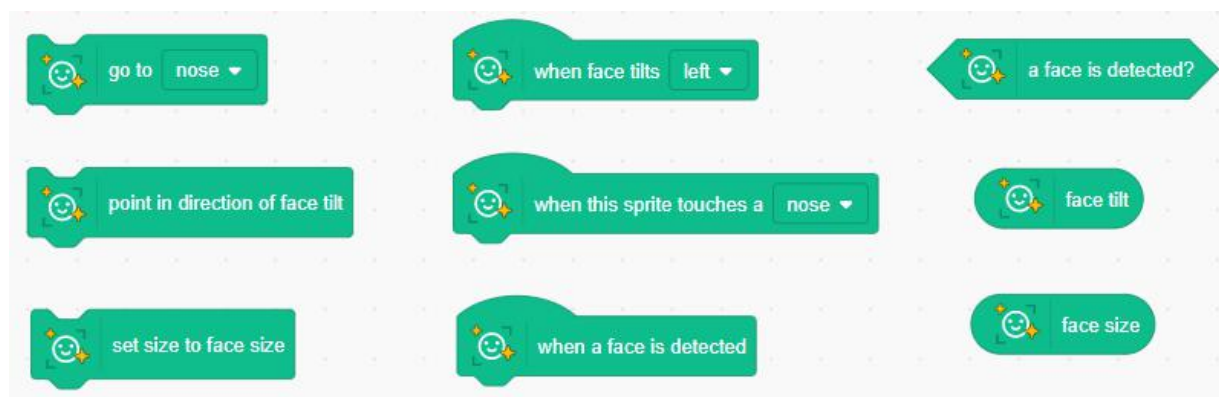
(ML4KIDS) <https://machinelearningforkids.co.uk/scratch3/>

Dostupno je proširenje za otkrivanje lica s 3 izvještajna bloka (*reporter* blokovi). U slučaju da koristite web kameru kao izvor, kombinirajte ju s proširenjem za osjet videa (*Video sensing extension*) da biste uključili/isključili videozapis s kamerom i postavili prozirnost.



Scratch (MIT) - <https://lab.scratch.mit.edu/face/>

Kliknite na „Try It Out“ (Isprobaj) i imat ćete 9 blokova za rukovanje prepoznavanjem lica.





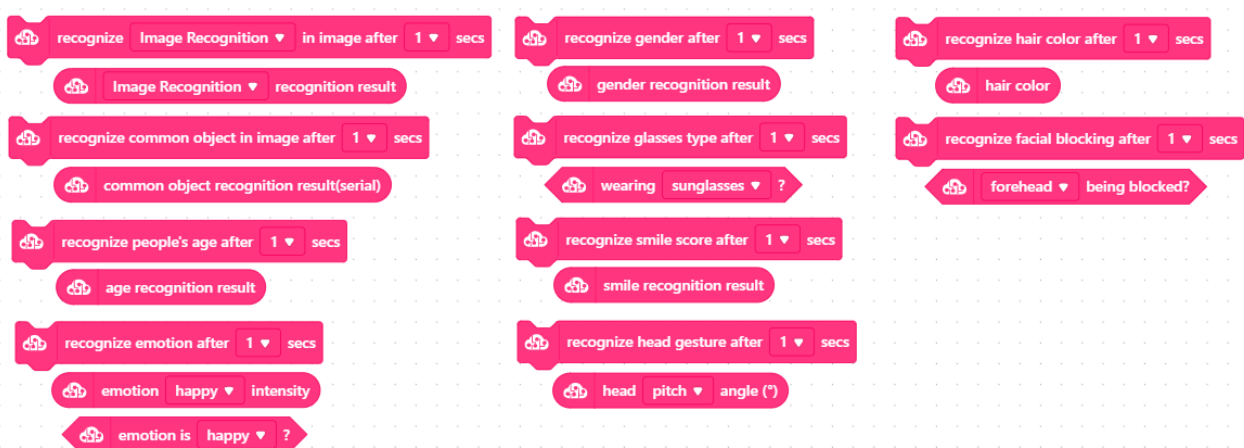
Scratch (MITMEDIALAB) - <https://mitmedialab.github.io/prg-extension-boilerplate/create/>
Učitajte proširenje *Face sensing* (osjet lica) i imat ćete 9 blokova za korištenje pri otkrivanju lica. Neki se blokovi koriste za određivanje izraza lica i prepoznavanje osjećaja. Također možete koristiti proširenje *Teachable machine* u kombinaciji s *Google Teachable machine* stranicom.



Makeblock (mBlock) - <https://ide.mblock.cc/>

Učitajte proširenje *Cognitive services* i proširenje za *Video sensing* i tu vam je na raspolaganju puno blokova. Ne postoji određen blok za otkrivanje lica, ali osobu možete otkriti s *Recognize* blokom. Ali zato imate puno blokova za određivanje određivanje emocija, dobi, spola, osmijeha, boje kose, naočala, čak i za situacije kada prekrivate dio lica.

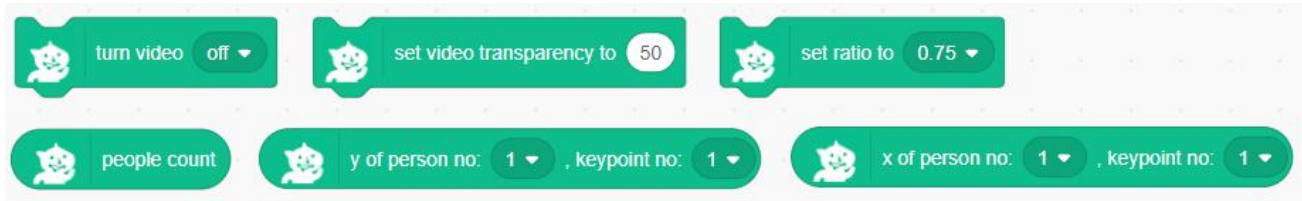
Makeblock također raspolaže sa svojim *Teachable machine* proširenjem (ne miješati to s Googleovim) gdje možete trenirati do 3 klase i koristiti ga za prepoznavanje lica ili otkrivanje objekata.





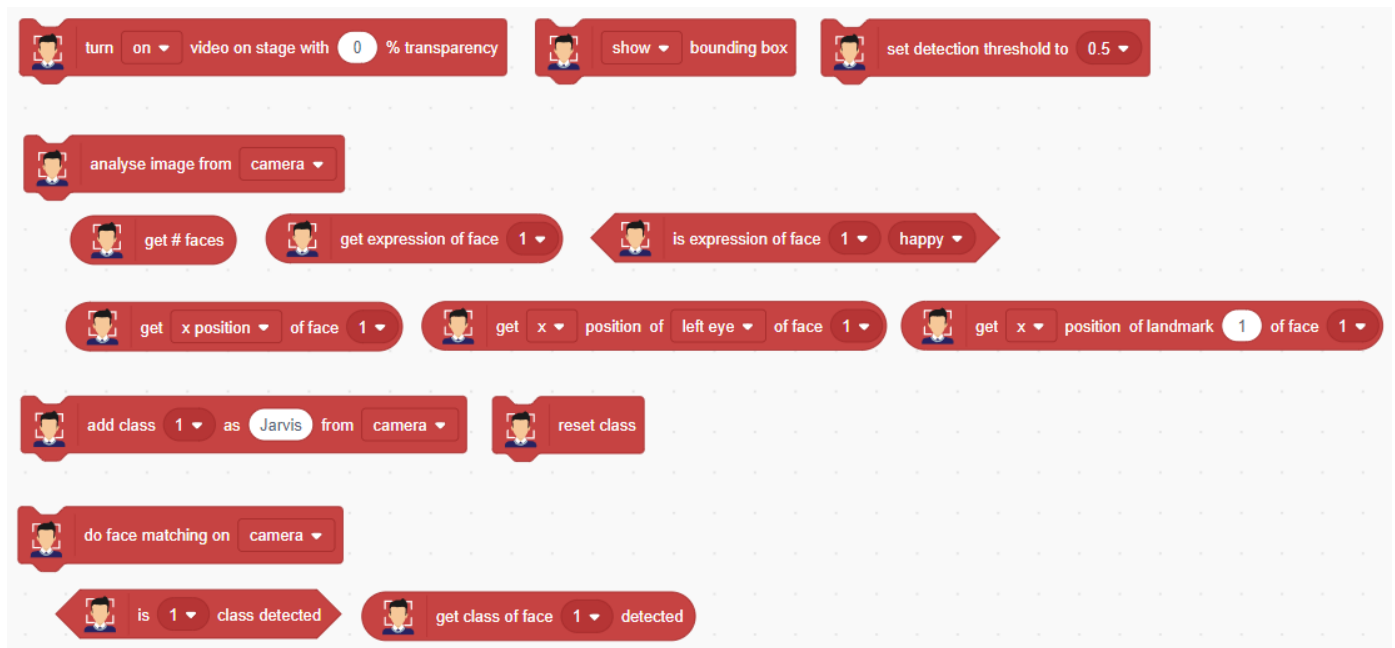
Stretch3 (github.io) - <https://stretch3.github.io/>

Učitajte **facemesh2Scratch** proširenje gdje imate 3 bloka za detekciju lica (postoje dodatna 3 bloka za rad s videozapisom). Glavna značajka je mogućnost višestrukog otkrivanja lica tako da možete otkriti više od jedne osobe u streamu kamere.



Posljednja aplikacija koja će se koristiti je **PictoBlox**, koja se prvo mora instalirati na računalo s poveznice: <https://thetempedia.com/product/pictoblox/download-pictoblox/> (427 Mb)

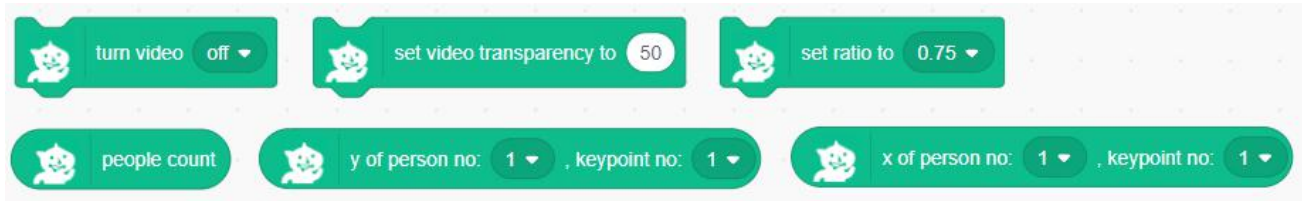
Nakon instalacije koristite proširenje *Face Detection* i tamo vas očekuje puno toga - možete otkriti više lica i njihove izraze. Također, postoji mogućnost treniranja klasa koja se koristi za prepoznavanje lica.





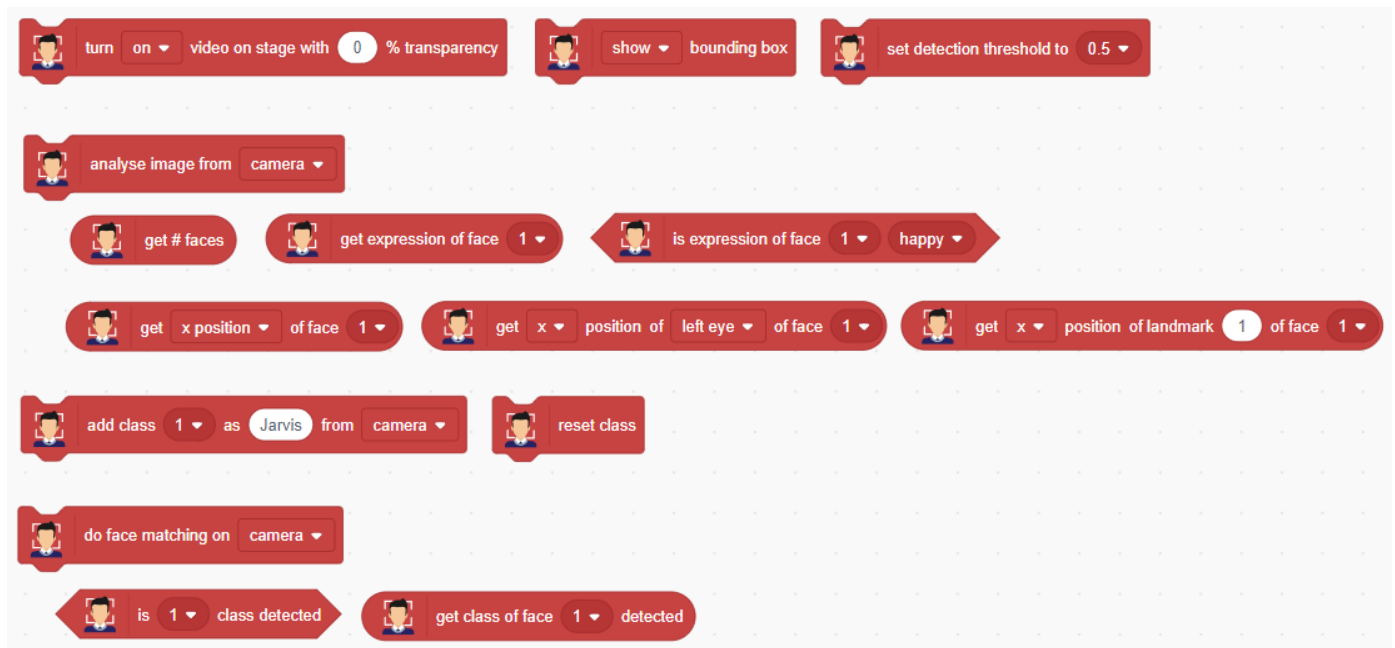
Stretch3 (github.io) - <https://stretch3.github.io/>

Učitajte **facemesh2Scratch** proširenje gdje imate 3 bloka za detekciju lica (postoje dodatna 3 bloka za rad s videozapisom). Glavna značajka je mogućnost višestrukog otkrivanja lica tako da možete otkriti više od jedne osobe u streamu kamere.



Posljednja aplikacija koja će se koristiti je **PictoBlox**, koja se prvo mora instalirati na računalo s poveznice: <https://thetempedia.com/product/pictoblox/download-pictoblox/> (427 Mb)

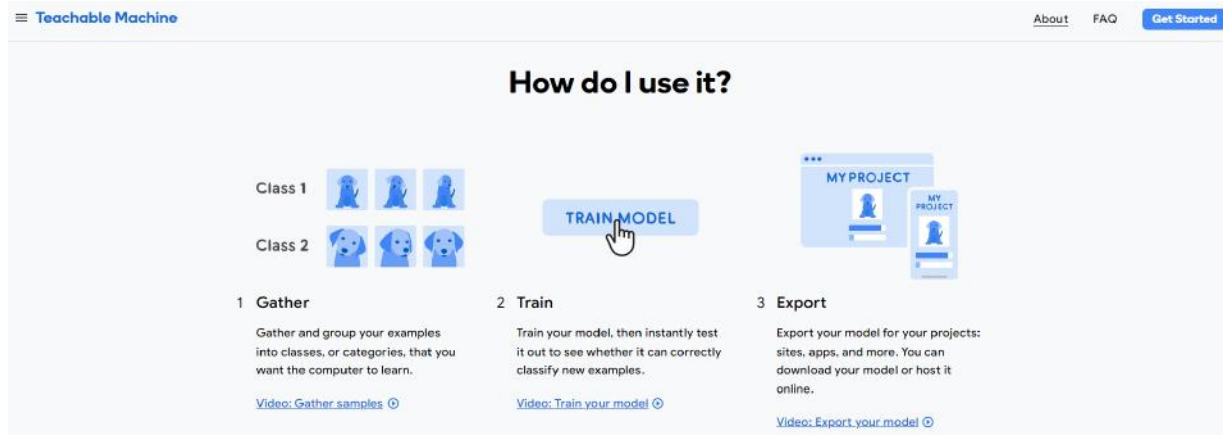
Nakon instalacije koristite proširenje *Face Detection* i tamo vas očekuje puno toga - možete otkriti više lica i njihove izraze. Također, postoji mogućnost treniranja klasa koja se koristi za prepoznavanje lica.





Teachable machine (Google) - <https://teachablemachine.withgoogle.com/>

Ova se aplikacija koristi za treniranje vašeg modela i njegovo korištenje za prepoznavanje lica u kombinaciji s *Teachable machine* proširenjem dostupnom u **Scratchu** (MITMEDIALAB)



Vidjeli smo različite aplikacije za otkrivanje i prepoznavanje lica.

Otkrivanje lica (detekcija lica) razlikuje se od prepoznavanja lica (ti pojmovi su jednoznačni i ne smiju se koristiti jedan umjesto drugog) po tome što otkrivanje lica uključuje samo otkrivanje lica unutar digitalne slike ili videozapisa. To jednostavno znači da sustav za otkrivanje lica može prepoznati da je na slici ili u videu prisutno ljudsko lice, ali ne može identificirati tu osobu.

Detekcija lica sastavni je dio sustava prepoznavanja lica – prva faza prepoznavanja lica je otkrivanje prisutnosti ljudskog lica. Detekcija lica može se koristiti i u kamerama za pomoć pri automatskom fokusiranju - vjerojatno ste primijetili da će se na nekim digitalnim fotoaparatom i telefonima pojaviti mala kutija oko lica ljudi otkrivenih unutar slike, što će kameri omogućiti da prednost daje fokusu na ta lica. Identificiranje prisutnosti ljudskog lica vrši se pomoću formula i algoritama.

Tipično, prva stvar koju će sustav za otkrivanje lica tražiti bit će oči jer su to jedna od najlakših značajki za identifikaciju. Tada bih također mogao tražiti prisutnost usta, obrva, nosa i nosnica. Otkrivanje lica važan je dio procesa prepoznavanja lica, međutim, iz sigurnosne perspektive neke posebnosti od sustava za otkrivanje lica - jednostavno prepoznaje da je lice prisutno, ali nema informacija o identitetu tog lica. Prepoznavanje lica igra ključnu ulogu u velikom broju primjena, posebno u kontroli prometa ljudi i provedbi zakona. Točno identificiranje pojedinaca pomaže u poboljšanju sigurnosnih sustava i zaštite u zračnim lukama i gradovima širom svijeta, a to se može učiniti jedino uz pomoć nekog od vodećih sustava za prepoznavanje lica dostupnih na tržištu.



Kao što možemo vidjeti, tehnologija prepoznavanja lica mijenja svijet u kojem živimo, a čini nam se kao da samo zagrebli po površini u smislu potencijala primjene programa za prepoznavanje lica. Iako se upotreba za prepoznavanja lica može činiti beskrajnom, ključno je osigurati da se ova tehnologija koristi na odgovarajući i odgovoran način.



ZAKLJUČAK

Detekcija lica je proces koji određuje prisutnost lica na slici ili u videoisječku dok se prepoznavanje lica odnosi se na identificiranje i prepoznavanje osoba.

Promisli!



Što sada znam?



Što sam naučio/la?



Što još želim znati?



Programiranje detekcije lica u Scratchu

Kako programirati detekciju lica u Scratchu?

Prođimo kroz nekoliko jednostavnih primjera programiranja detekcije lica u Scratch aplikacijama. Vjerojatno ste primijetili da neke aplikacije na vašem pametnom telefonu crtaju pravokutnik oko lica kao rezultat otkrivanja lica. To je također moguće učiniti u Scratchu.

U nastavku ćemo upoznati programe za otkrivanje lica i njegove upotrebe prolaskom kroz određene primjere.

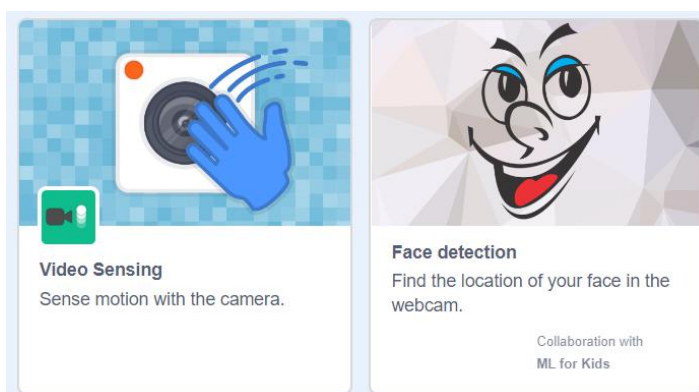




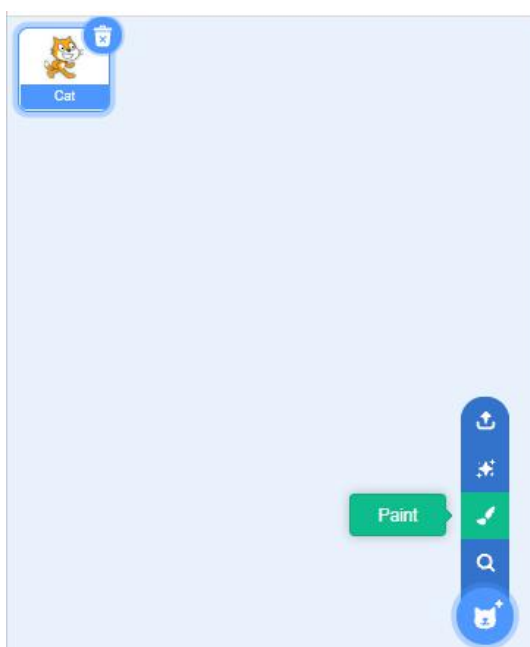
Prvi projekt - SCRATCH (ML4KIDS):

Korak 1: otvorite internetski preglednik Chrome i idite na: <https://machinelearningforkids.co.uk/scratch3/>

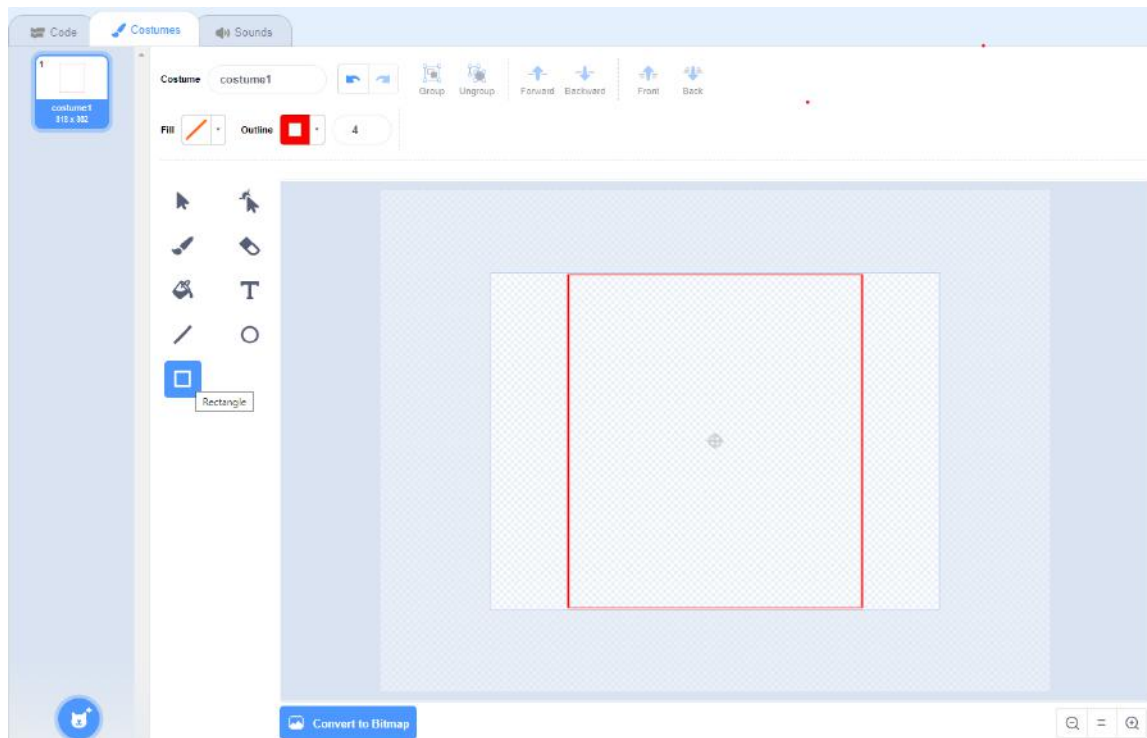
Korak 2: Učitajte proširenja za Video i Detekciju lica (*Face detection*), a zatim priključite web kameru (ako nemate ugrađenu)



Korak 3: **Izbrišite lik mačke (Cat)** klikom na ikonu kante za smeće (gornji desni kut) i odaberite opciju **bojanja** novog spritea

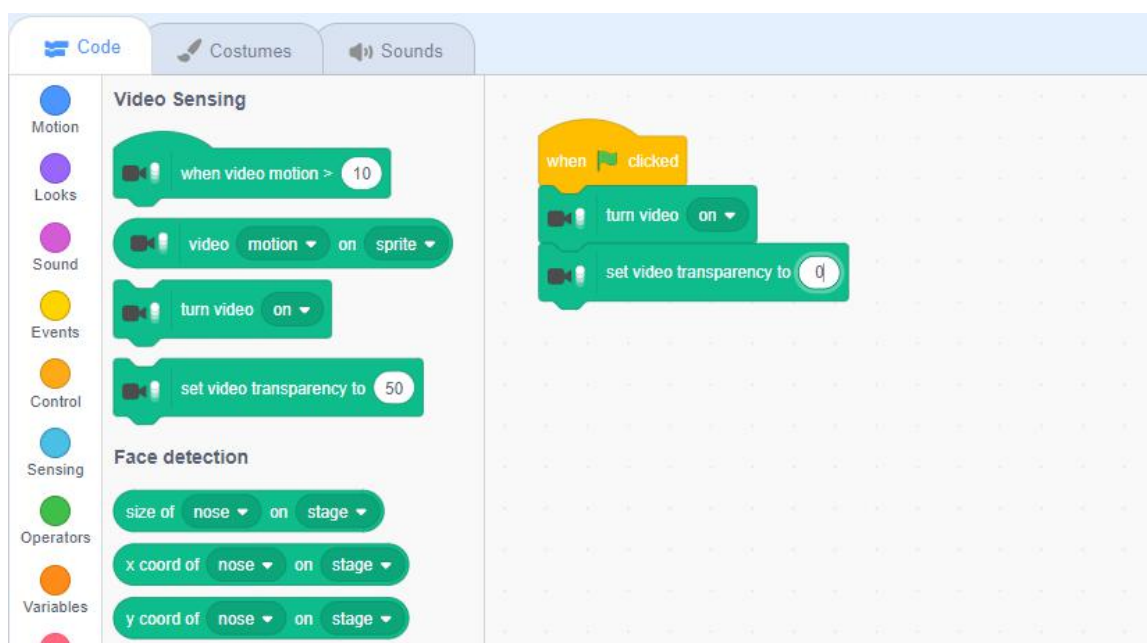


Korak 4: Naslikajte pravokutnik (koristit će se kao granični okvir) bez ispune i postavite boju okvira na crvenu i debljinu na 4 kao na slici ispod



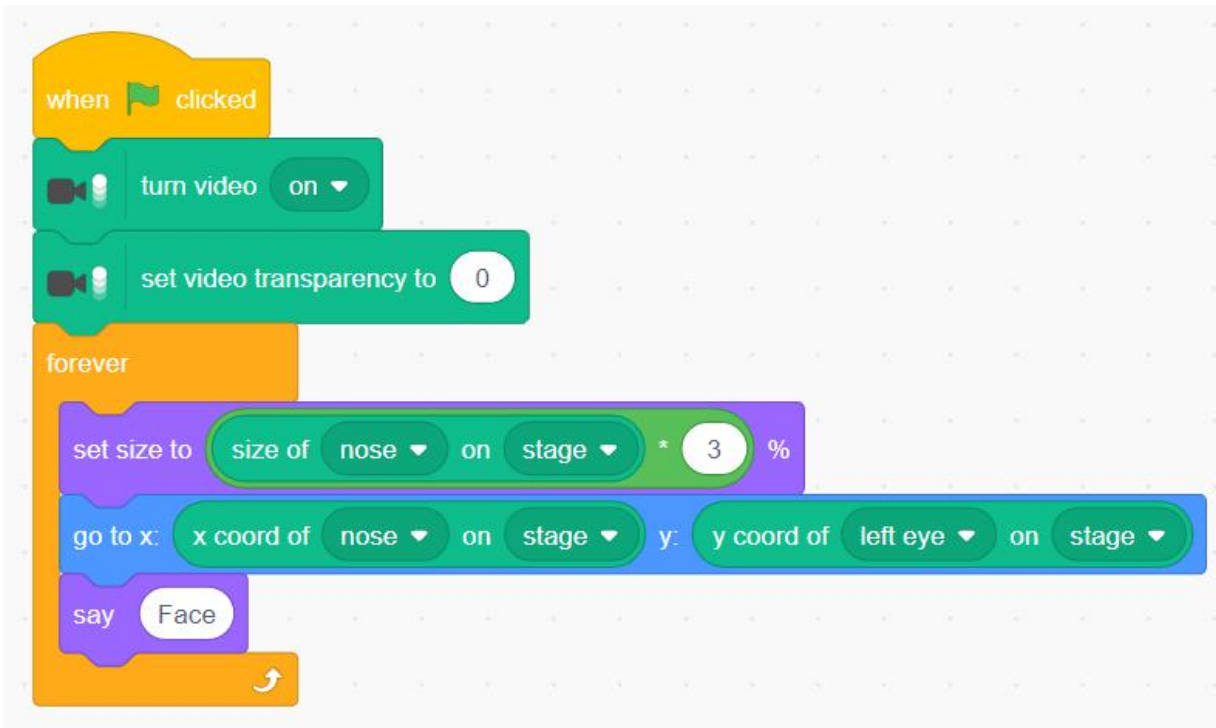
f

Korak 5: prijedite na karticu **Skripte** i započnite s programiranjem. Prvo uključujemo videozapis i postavljamo njegovu prozirnost na 0 (netransparentno).





Korak 6: Zatim slijedi petlja (ponavlja) s 3 bloka u njoj kako bi se postavila veličina pravokutnika (veličina nosa pomnožena s 3) i njegove koordinate x i y. Možda ćete htjeti prilagoditi vrijednost multiplikatora s 3 na neki drugi broj, pogotovo ako prijedete na cijeli zaslon (1,5 je bolje).



Drugi projekt - otkrivanje lica u kombinaciji s proširenom stvarnošću

Korak 1: Otvorite Scratch na poveznici <https://machinelearningforkids.co.uk/scratch3/>

Korak 2: Dodajte proširenje za detekciju lica (*Face detection*)

Korak 3: Dodavanje proširenje Video i priključite web kameru (ako nemate ugrađenu)

Korak 4: Obrišite lik mačke

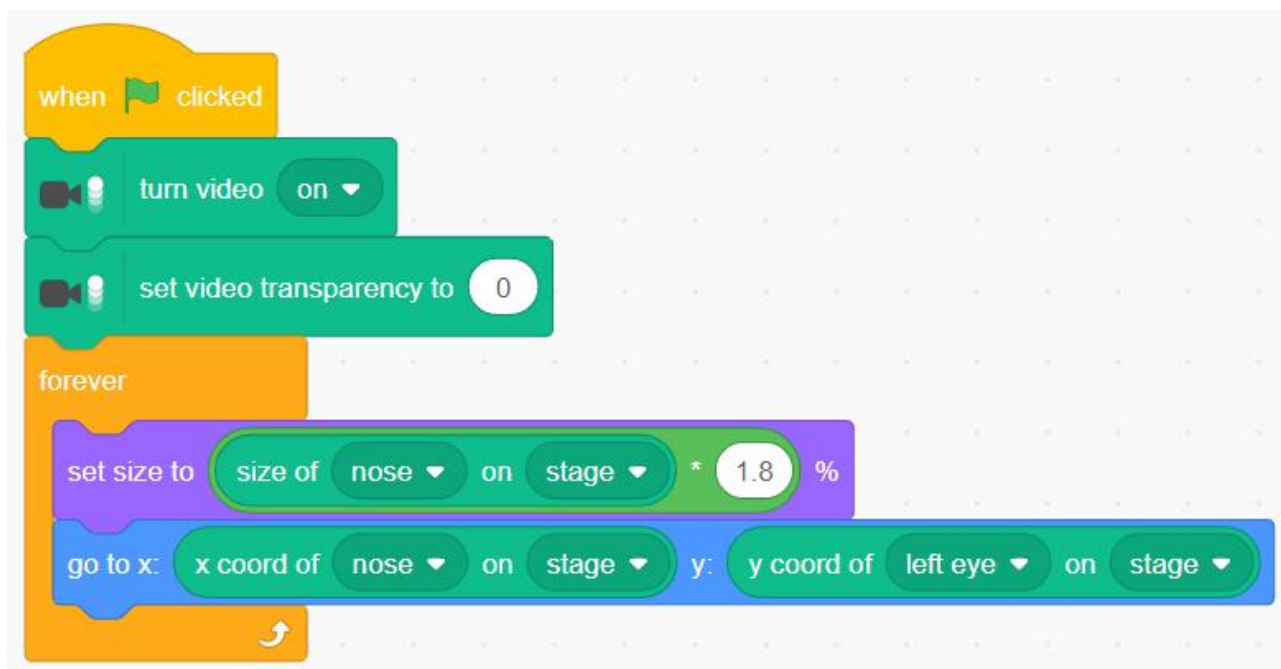
Korak 5: Preuzmite sliku s poveznice https://toppng.com/transparent-glasses-PNG-free-PNG-Images_110945

Korak 6: Prenesite tu sliku u Scratch kao lik i preimenujte je u „Naočale“ ili slično

Korak 7: Skripta za lik - unijeti prema slici ispod:

Korak 8: Pokrenite program i pomičite glavu lijevo-desno i naprijed-nazad

Korak 9: Razgovarajte o točnosti algoritma i kako ga poboljšati.



Treći projekt s PICTOBLOX-om (Desktop aplikacija):

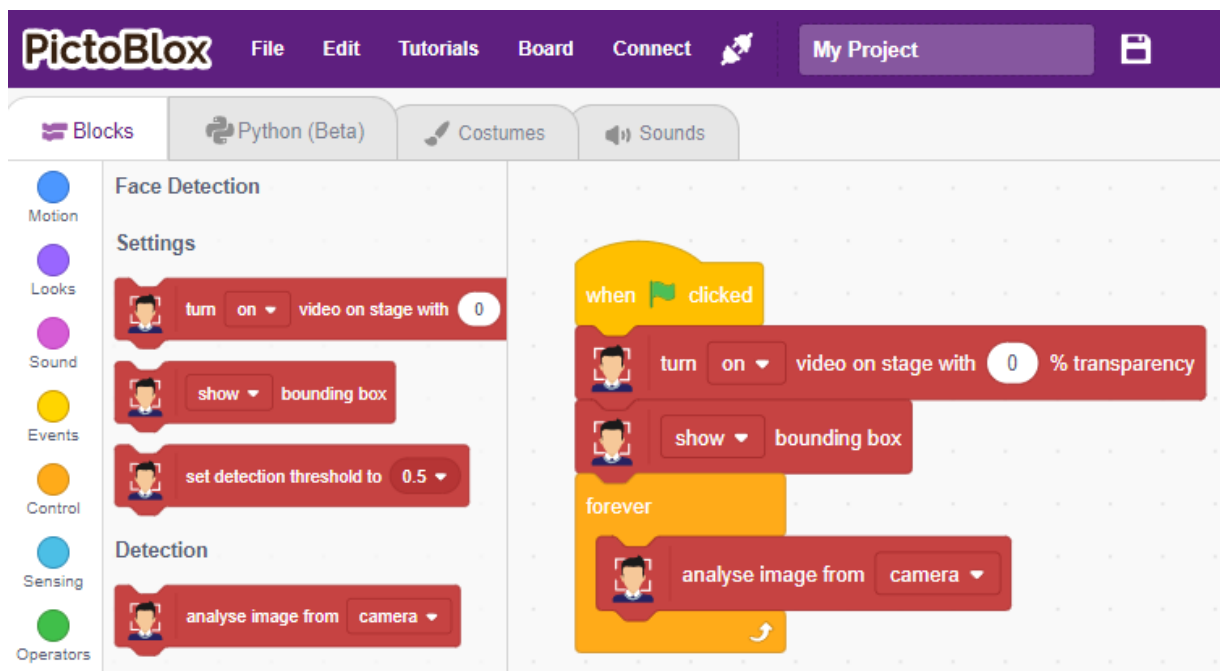


Korak 1a: Budući da nema dostupnog internetskog sučelja za ovu aplikaciju, morate instalirati PictoBlox sa poveznice: <https://thetempedia.com/product/pictoblox/download-pictoblox/> (427 Mb)

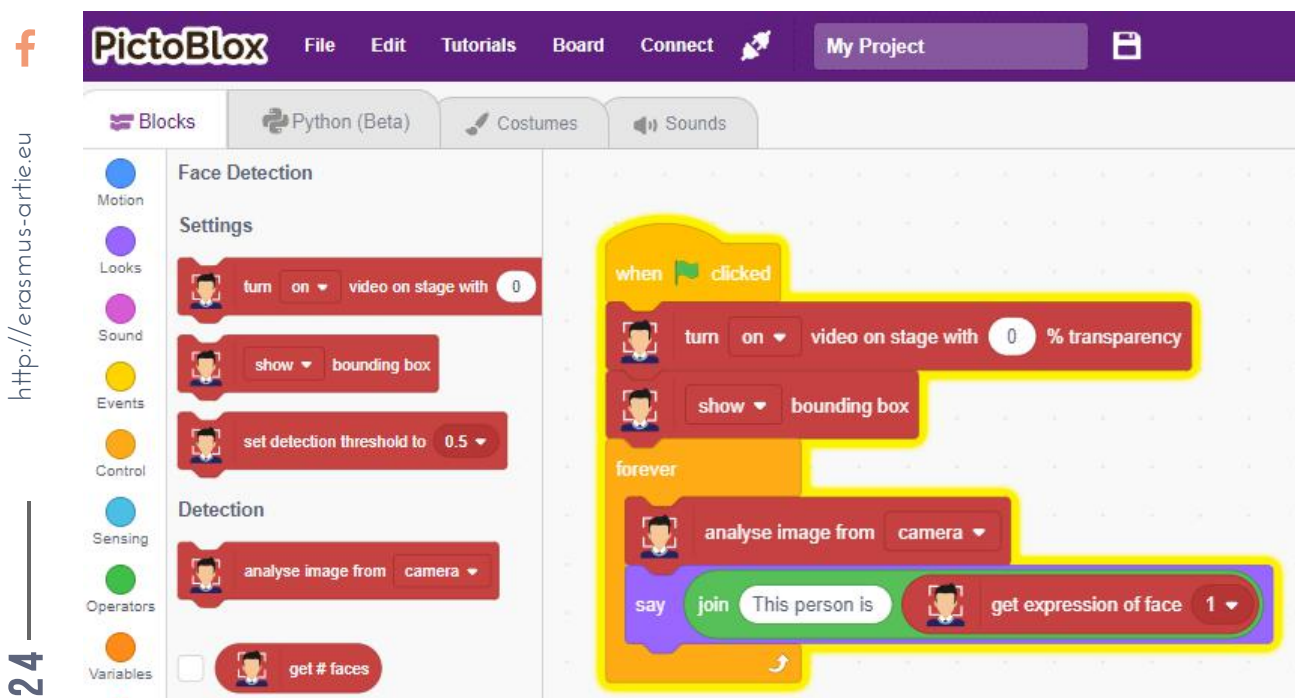
Korak 2a: Otvorite PictoBlox i odaberite proširenje za detekciju lica (*Face detection*)



Korak 3a: Koristite blokove kao na slici ispod, a vrlo je jednostavno shvatiti kako skripta funkcionira. Sada možemo uključiti iscrtavanje graničnog okvira s odgovarajućim blokom i nema potrebe crtati pravokutnik. No, glavna značajka je da **možete detektirati više lica**. Nazovite nekoga da vam se pridruži pred kamerom da vidite kako to funkcionira. Označite (čekirajte) reporter blok get # faces kako biste vidjeli koliko je lica detektirano.



Korak 4a: Poboljšajmo skriptu pomoću operatora spajanja za prikaz izraza lica osobe. Možete dodatno istražiti kako to funkcionira s više lica.





Danas se programi za otkrivanje koriste u gotovo svim poljima, od mobilnih uređaja do *snapchat* filtera za lice i različitih sigurnosnih aplikacija. Otkrivanje lica pomaže vam prepoznati lica, njihovu dob, izraze, spol, lokaciju i mnoge druge značajke. Detekcija lica je dakle širi pojam, a ima ga svaki sustav koji može identificirati prisutnost ljudskog lica na slici. Detekcija lica ima brojne aplikacije, uključujući brojanje ljudi, internetski marketing, pa čak i automatsko fokusiranje objektiva kamere. Njegova temeljna svrha je označiti prisutnost lica. Prepoznavanje lica posljednjih je godina postalo značajnije i relevantnije zbog potencijalnih primjena. Budući da su lica vrlo dinamična i predstavljaju više pitanja i izazova koje treba riješiti, istraživači u domeni prepoznavanja uzoraka, računalnog vida i umjetne inteligencije predložili su mnoga rješenja za smanjenje takvih poteškoća kako bi se poboljšala robusnost i točnost prepoznavanja.

ZAKLJUČAK

Detekcija lica koristi se u različitim sofisticiranim sustavima, a moguće je izraditi jednostavne primjere u alatima poput Scratcha i PictoBlox.



Promisli!



Što sada znam?



Što sam naučio/la?



Što još želim znati?



Programiranje prepoznavanja lica u Scratchu

Što je prepoznavanje lica?
Kako funkcionira prepoznavanje lica?
Koji praktičnu primjenu može imati?

Pomoću primjera gotovih programa steći ćete bolje razumijevanje programa prepoznavanja lica i njihove uporabe.





Korak 1: Otvorite svoj web preglednik i preuzmite svih 20 slika s poveznice: <https://bit.ly/daenerys-data>

Koristit će se za treniranje klase 1

Korak 2: Otvorite svoj web preglednik i preuzmite svih 20 slika s poveznice: <https://bit.ly/arya-data>

Koristit će se za treniranje klase 2

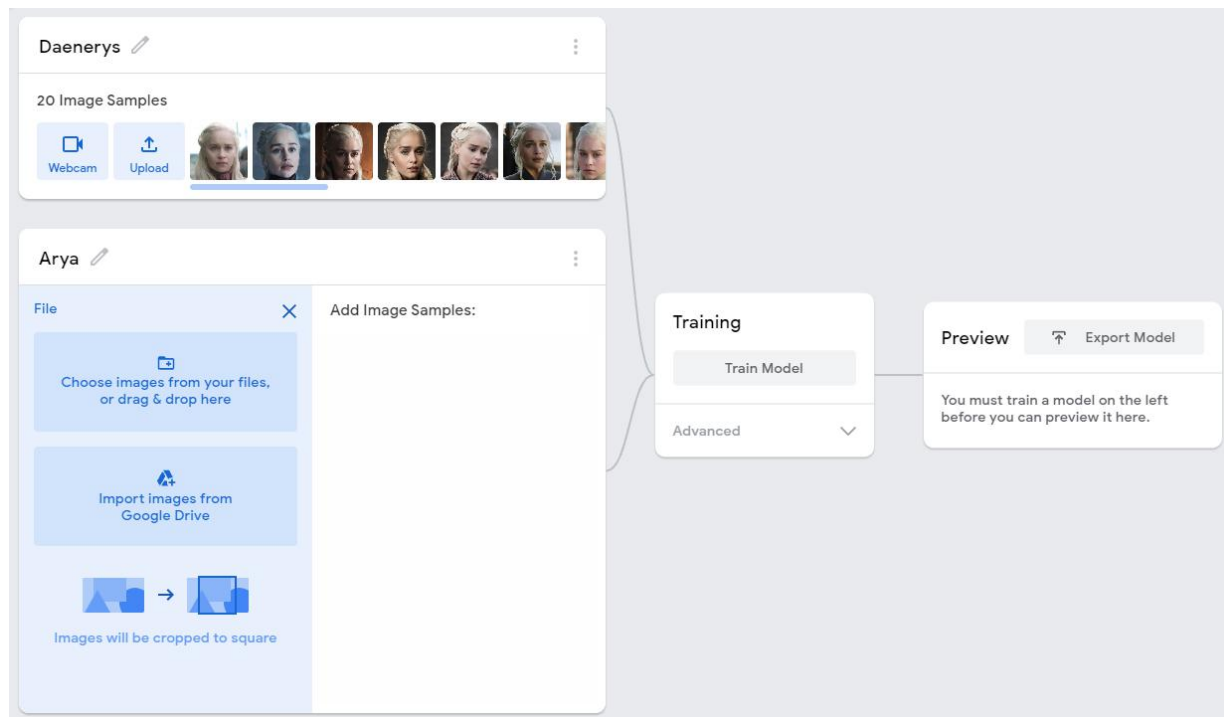
Korak 3: Otvorite web preglednik i idite na: <https://teachablemachine.withgoogle.com/>

Korak 4: Kliknite na *Get Started* (početak rada).

Korak 5: odaberite *Image project* (projekt sa slikama)

Korak 6: odaberite *Standard image model* (standardni model za slike)

Korak 7: Promijenite nazive klasa - Class 1 u Daenerys i Class 2 u Arya. Prenesite slike Daenerys u Daenerys *File* (datoteka) i Aryine slike u Arya *File* (datoteka) kao što je prikazano na slici ispod





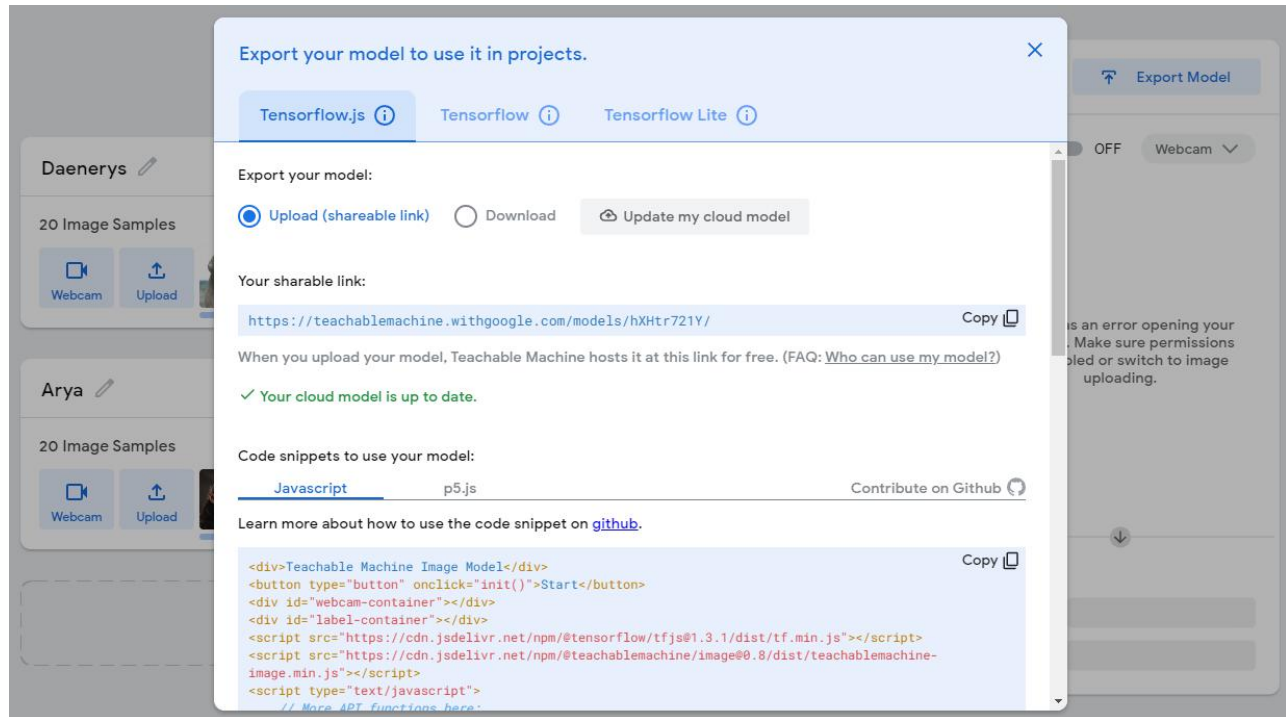
Korak 8: Trenirajte svoj model. Nemojte mijenjati kartice preglednika tijekom postupka treniranja.



Korak 9: Izvezite svoj model (*Export model*). Na skočnom prozoru odaberite *upload my model* (prijenos u oblak - treća opcija) i Google će besplatno pohraniti vaše podatke.



Korak 10: Kopirajte poveznicu istaknutu u donjem tekstualnom polju (*Your sharable link*) - ovo je URL vašeg modela. U ovom slučaju to je <https://teachablemachine.withgoogle.com/models/hXHtr721Y/>



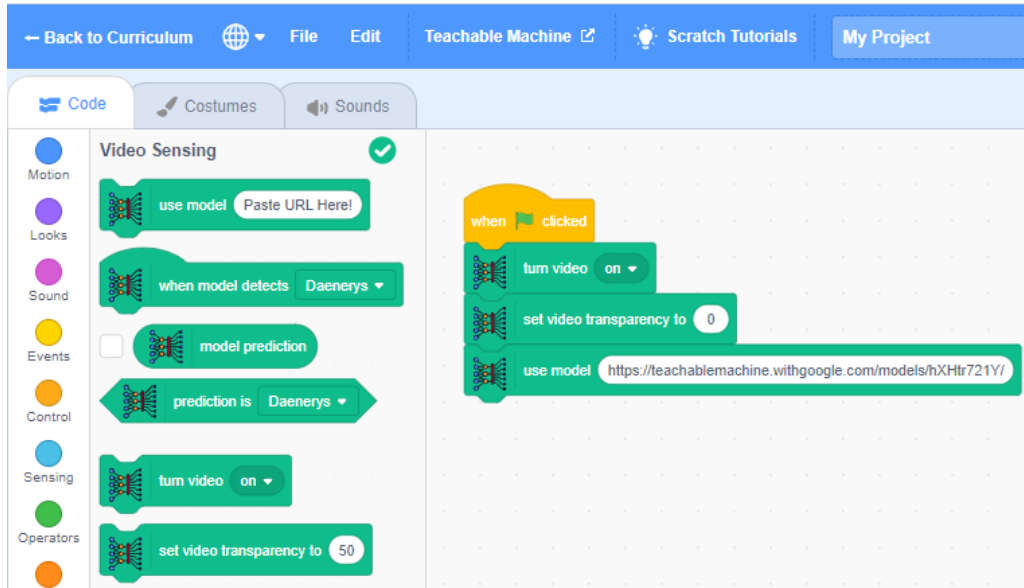
Korak 11: Vaš je model spreman za uporabu

Korak 12: Otvorite Scratch GUI na: <https://mitmedialab.github.io/prg-extension-boilerplate/create/> i učitajte *Teachable machine* proširenje.



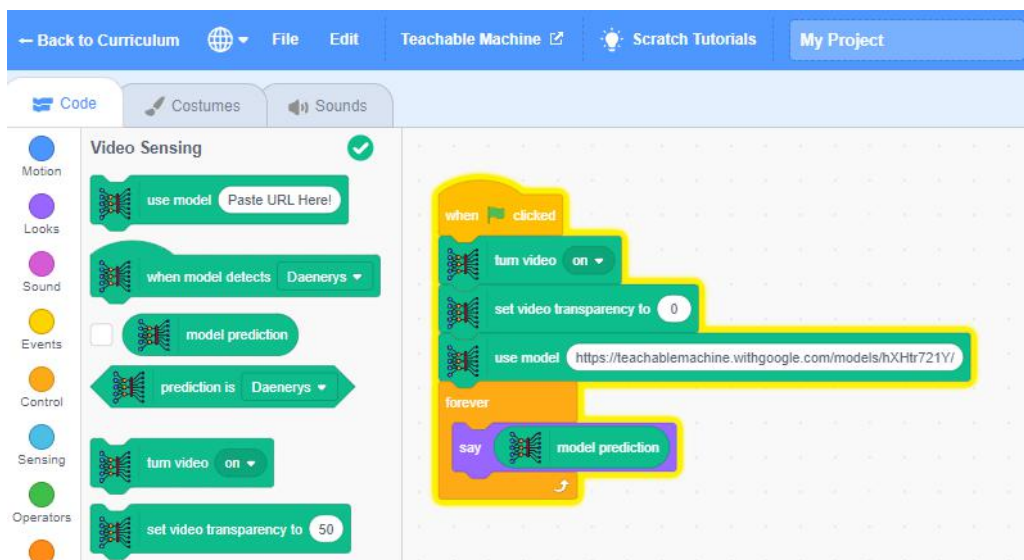


Korak 13: Prvo moramo uključiti video s kamere i postaviti prozirnost videa na 0 (netransparentno). Zatim koristimo blok *use model* i tamo zalijepimo poveznicu modela koju smo dobili u koraku 10.



Korak 14: Posljednjih nekoliko blokova su petlja (ponavljač) i blok reci u kojem se nalazi reporter blok s rezultatom predviđanja (*model prediction*). Sigurno nećete imati prave Daenerys ili Aryu ispred kamere pa upotrijebite pametni telefon s njihovim slikama i usmjerite ga prema kameri da biste vidjeli rezultate. Model možete trenirati vlastitim slikama ili slikama svojih prijatelja.

*** Pazite da ne snimate nečiju fotografiju bez dopuštenja.





PICTOBLOX (Desktop aplikacija):

Korak 1a: Budući da ovdje radimo s aplikacijom na računalu, morate prethodno instalirati najnoviju verziju PictoBlox-a s

poveznice: <https://thetempedia.com/product/pictoblox/download-pictoblox/> (577 Mb)

Korak 2a: Otvorite PictoBlox i odaberite proširenje za detekciju lica (*Face detection*)



Korak 3a: Otvorite svoj web preglednik i preuzmite svih 20 slika s

poveznice: <https://bit.ly/daenerys-data>

Koristit će se za treniranje klase 1

Korak 4a: Otvorite svoj web preglednik i preuzmite svih 20 slika s

poveznice: <https://bit.ly/arya-data>

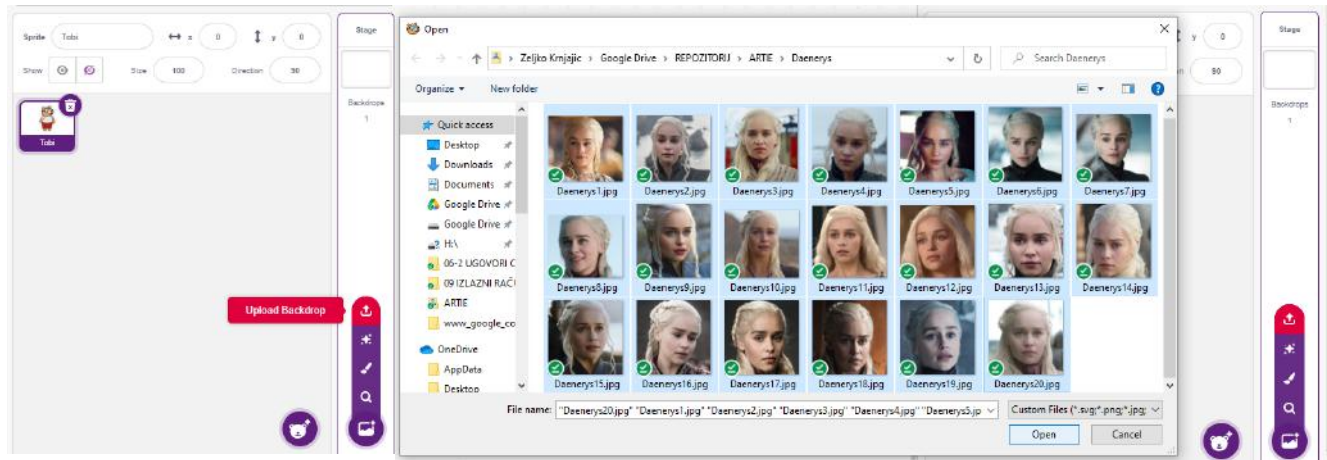
Koristit će se za obuku klase 2

Korak 5a: Sakrijte lik Tobi-ja s pozornice (*Stage*) klikom na ikonu kao što je prikazano na slici ispod.

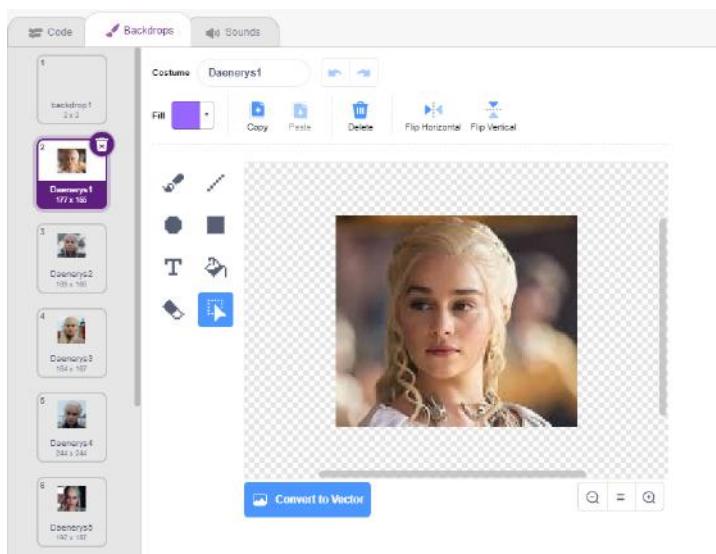




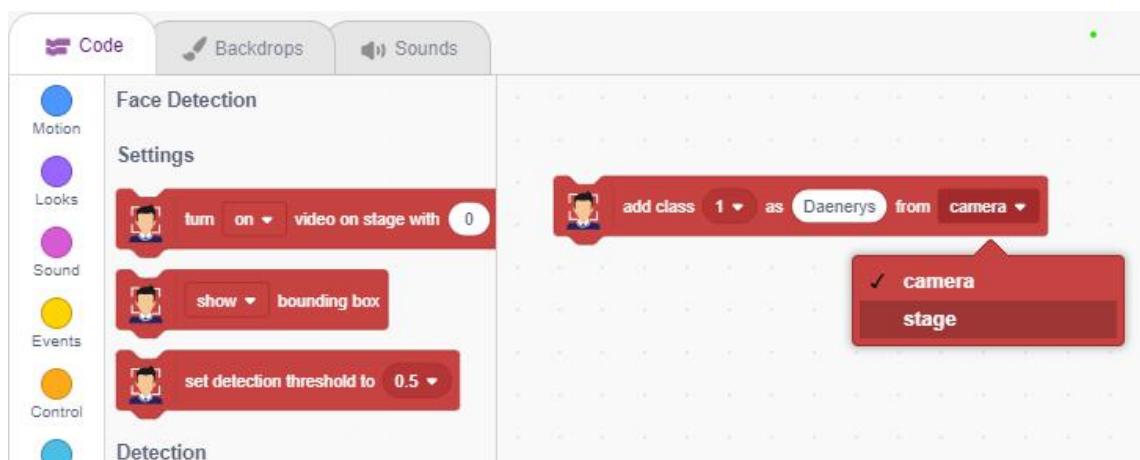
Korak 6a: Prenesite sve Daenerys slike u pozadinu (*Upload Backdrop* - Odaberite sve slike - Otvori)



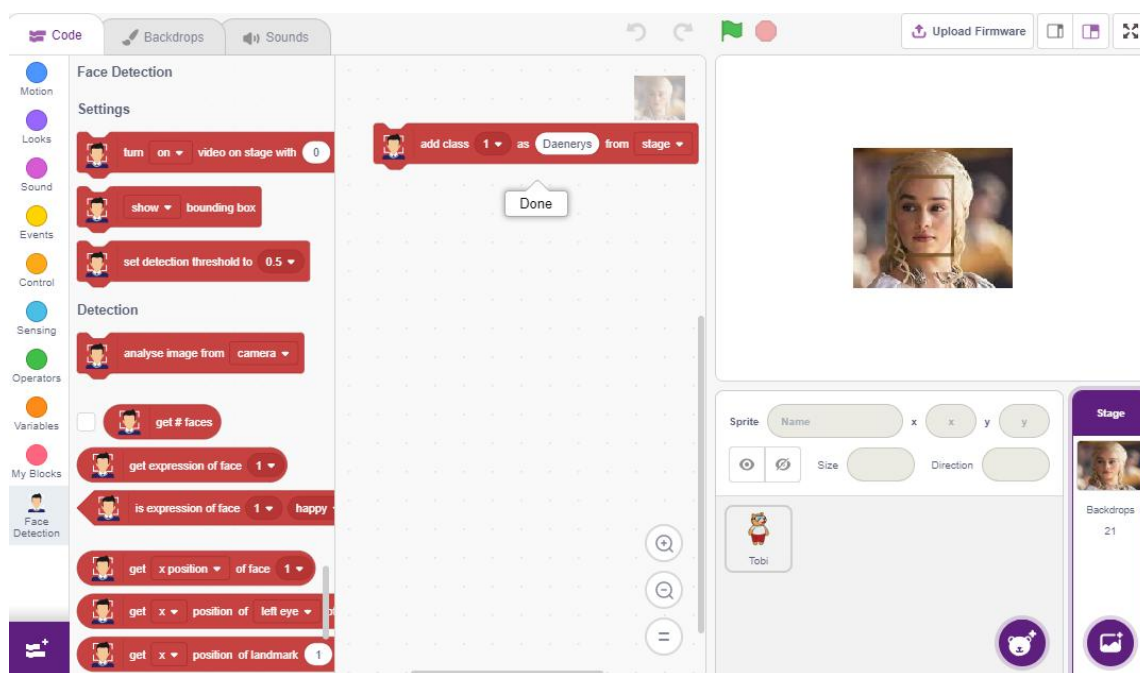
Korak 7a: Odaberite sliku Daenerys1 i prijedite na karticu *Code (Blocks)*



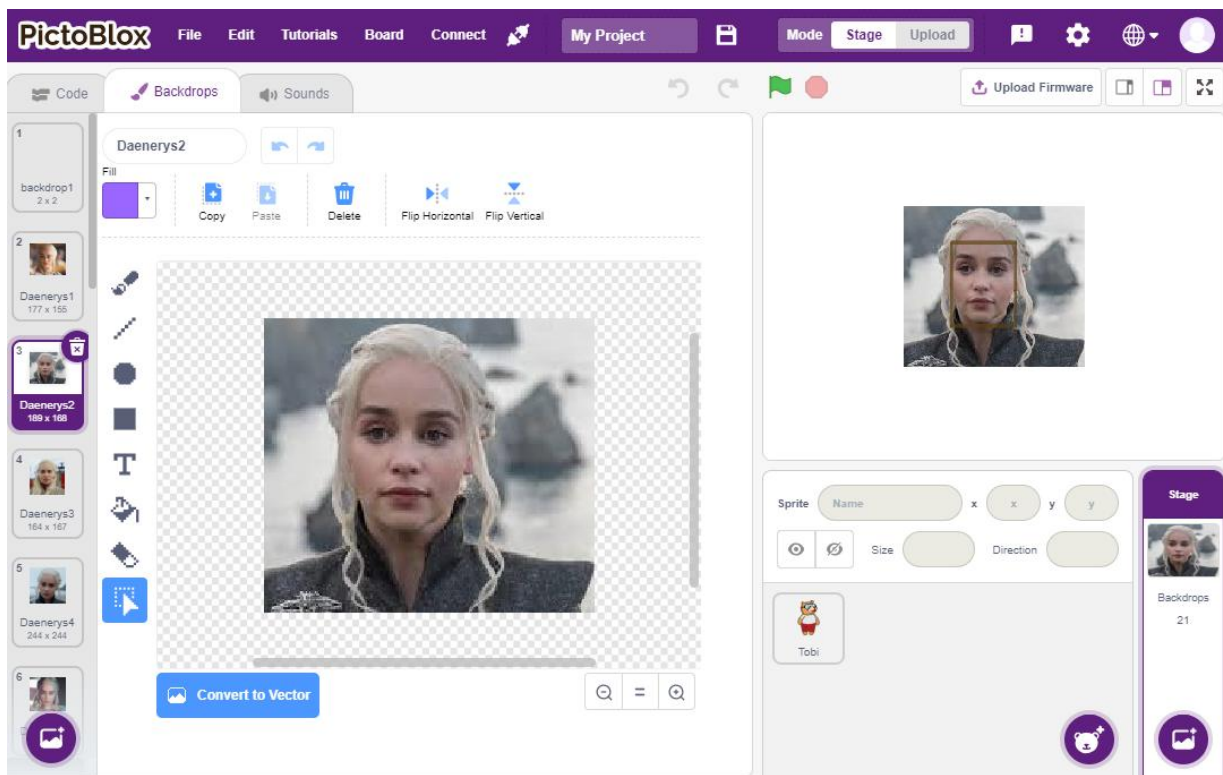
Korak 8a: U grupi blokova za detekciju lica (*Face detection*) nađite blok dodaj klasu (*add class*), povucite ga i ispustite na radni prostor i preimenujte naziv klase iz Jarvis u **Daenerys** i promijenite izvor (*from*) s kamere (*camera*) u pozornicu (*stage*) kao što je prikazano na slici ispod.



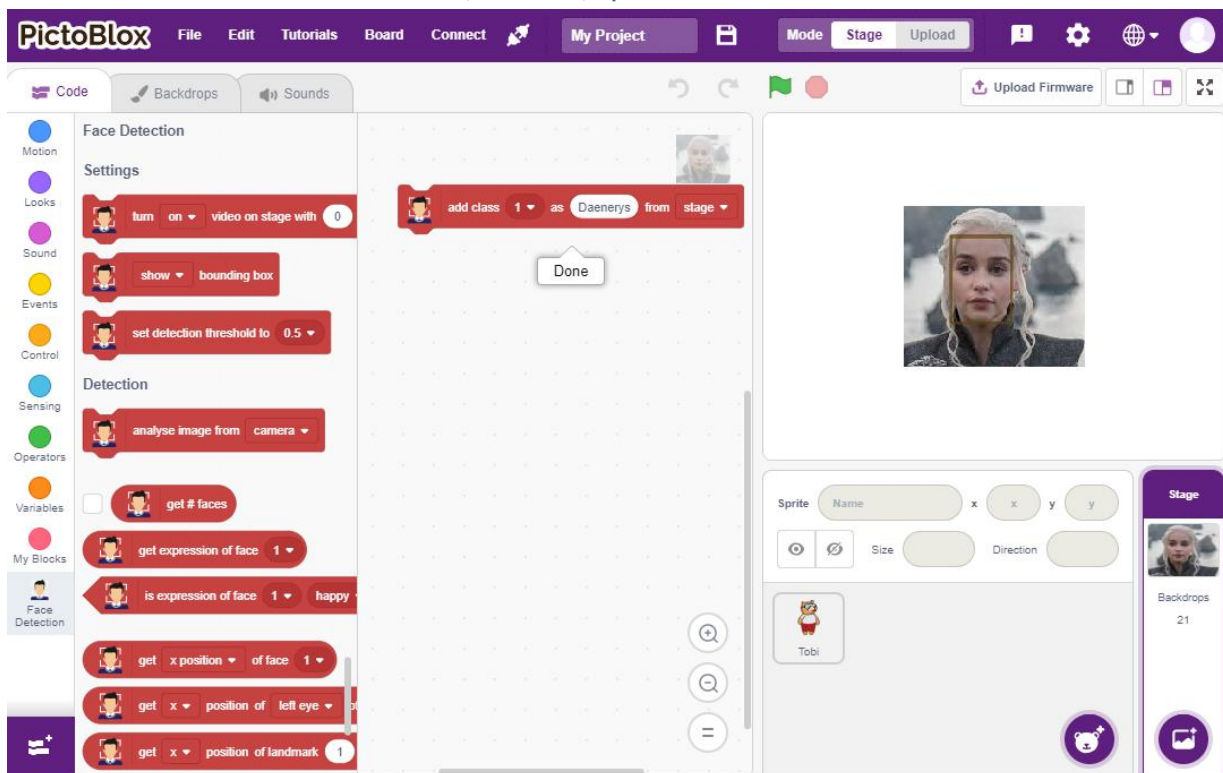
Korak 9a: Nakon što su promjene izvršene - samo kliknite na *add class* blok za početak treninga. Primit ćete poruku "Done" (gotovo) i vidjeti granični okvir na licu Daenerys. Treniranje s prvom slikom je dovršeno i morat ćete ponoviti ove korake za sve ostale slike.



Korak 10a: Vratite se na karticu *Backdrops* i odaberite Daenerys2 sliku.



Korak 11a: Vratite se na karticu *Code* (ili *Blocks*) i ponovno kliknite na *add class* blok

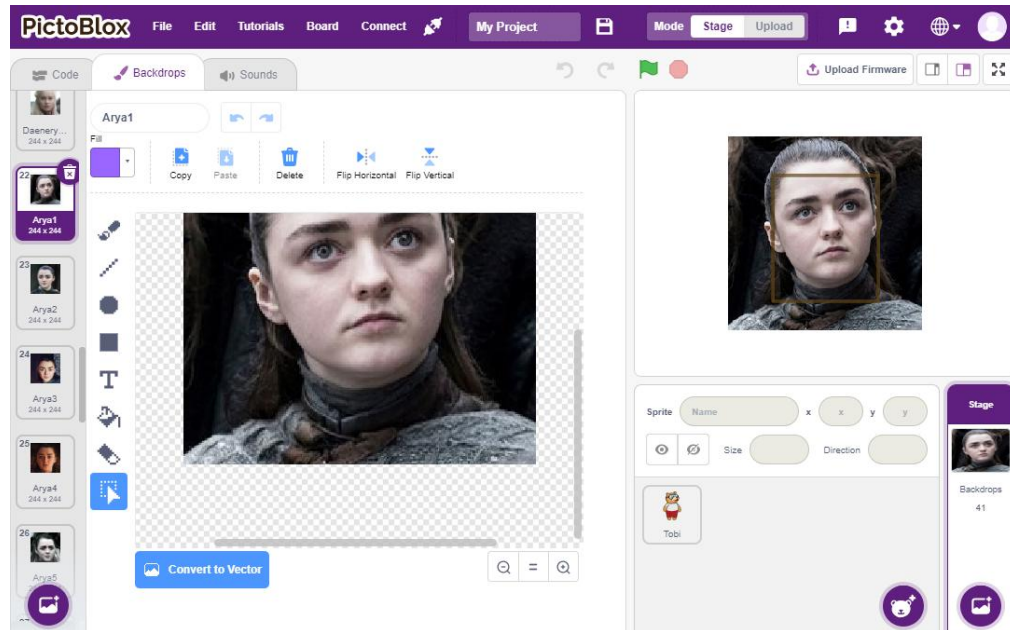




Korak 12a: Ponovite korake 10a i 11a za svaku pojedinu sliku (do Daenerys20)

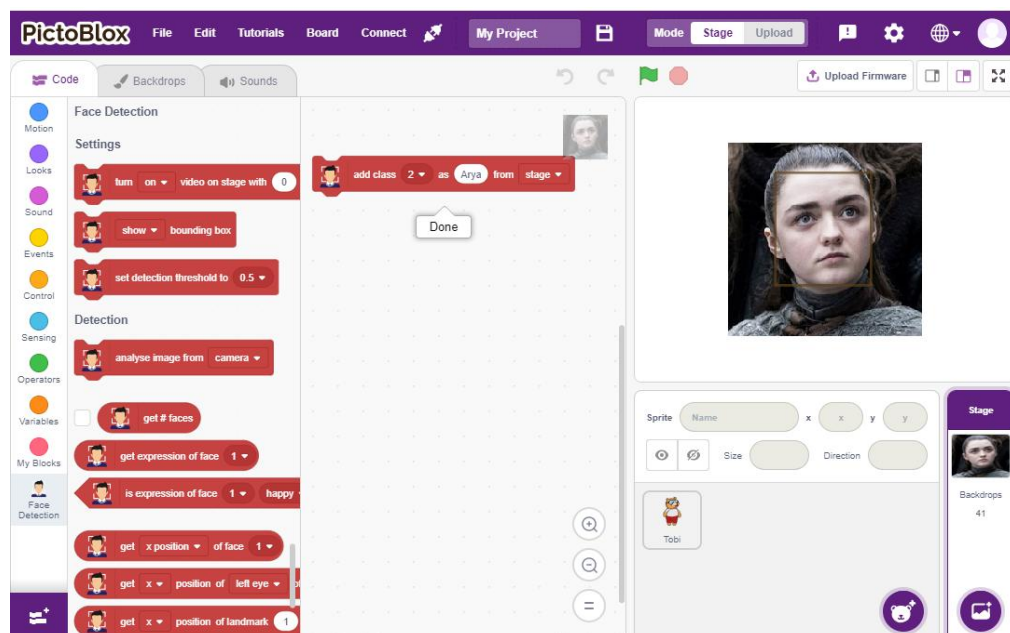
Korak 13a: Sada prenesite sve Arya slike u pozadinu (*Backdrop*) na isti način kao što ste to učinili u koraku 6a

Korak 14a: Odaberite sliku Arya1



f

Korak 15a: Promijenite karticu u *Code (Blocks)* i **promijenite** (to je važno jer treniramo drugu klasu) **add class** u 2 i naziv iz Daenerys u **Arya**. Izvor ostaje isti - pozornica (stage). Nakon toga kliknite na taj blok za treniranje Arya klase prvom slikom.





Korak 16a: prijedite na karticu *Backdrop* i odaberite sliku Arya2

Korak 17a: Prijedite na karticu *Code (Blocks)* i kliknite na **add class** blok

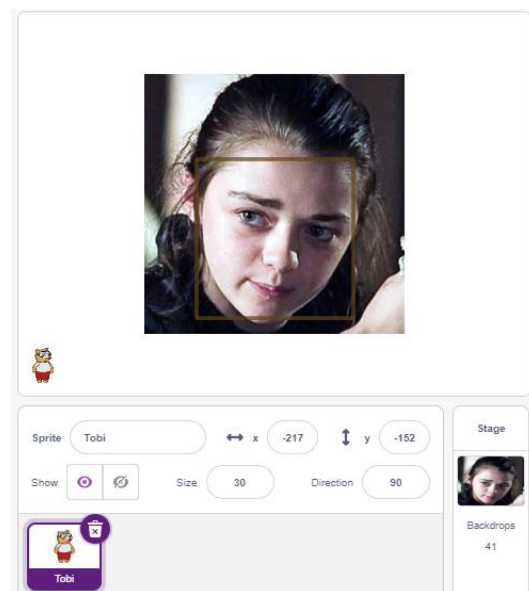
Korak 18a: Ponovite korake 16a i 17a za svaku pojedinu sliku (sve do Arya20)

Korak 19a: Vaš je model sada spreman za testiranje. Odaberite Tobi lik (*sprite*) i učinite ga vidljivim (*Show*).

Postavite veličinu (*Size*) Tobija na 30% i pomaknite je iz središta u donji lijevi kut pozornice (*stage*).

Korak 20a: Priključite kameru (ako nemate ugrađenu) i počnite programirati. Uključite video s 0% prozirnosti i prikažite granični okvir. Sljedeći blok je petlja zauvijek (*forever*) i Tobi će reći (*say*) rezultat prepoznavanja lica. Tu imamo dvostruki ako-onda blok i u drugom bloku će ako lice nije prepoznato - oblačić za govor biti prazan.

Korak 21a: Pokrenite program. Koristite svoj pametni telefon sa slikama Daenerys ili Arye i usmjerite ga prema kameri da biste vidjeli rezultate. Model možete trenirati vlastitim slikama ili slikama svojih prijatelja. *** Pazite da ne snimate nečiju fotografiju bez dopuštenja.





PROJEKT S PREPOZNAVANJEM LICA

Evo malog projekta koji je u osnovi korištenje prepoznavanja lica u praksi. Za vas smo pripremili trenirani model s 8 poznatih žena u *Teachable Machine*-u kako bismo ga usporedili s vašim licem. Za ovu priliku model je treniran slikama poznatih osoba: Adriane Lime, Emilie Clarke, Gal Gadot, Natalie Portman, Selene Gomez, Emme Stone, Zoe Saldana i Maisie Williams. Tako možete saznati koje lice s popisa ima najviše sličnosti s vašim licem.

Korak 1b: Otvorite Scratch sučelje na poveznici:

<https://mitmedialab.github.io/prg-extension-boilerplate/create/> i učitajte *Teachable machine* proširenje.

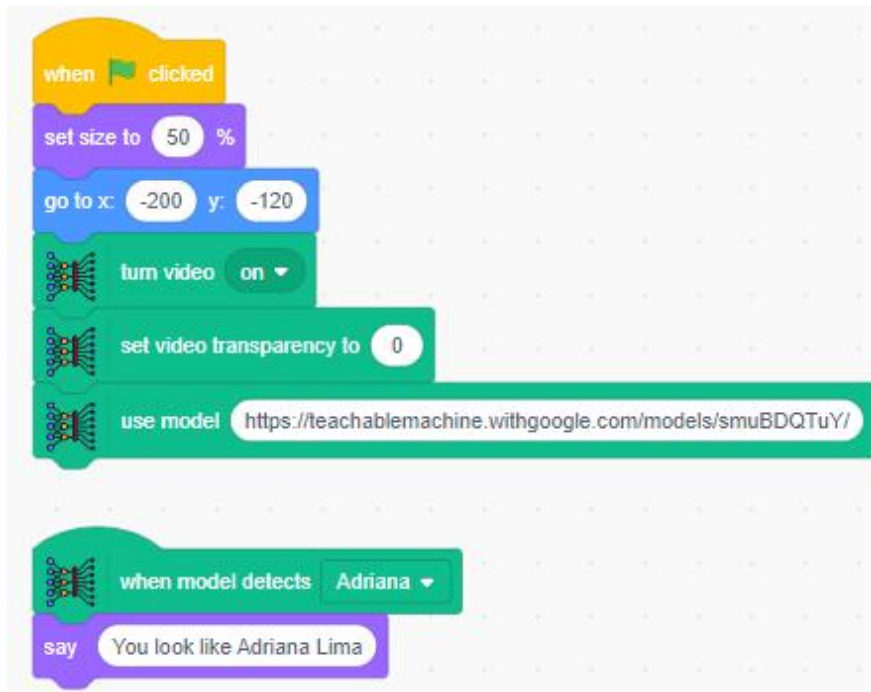


Korak 2b: Prvo postavite veličinu i položaj lika, a zatim uključite video i postavite njegovu prozirnost na 0 (netransparentno). Zatim koristimo blok *use model* i u polje za poveznicu zalijepimo: <https://teachablemachine.withgoogle.com/models/smuBDQTuY/>

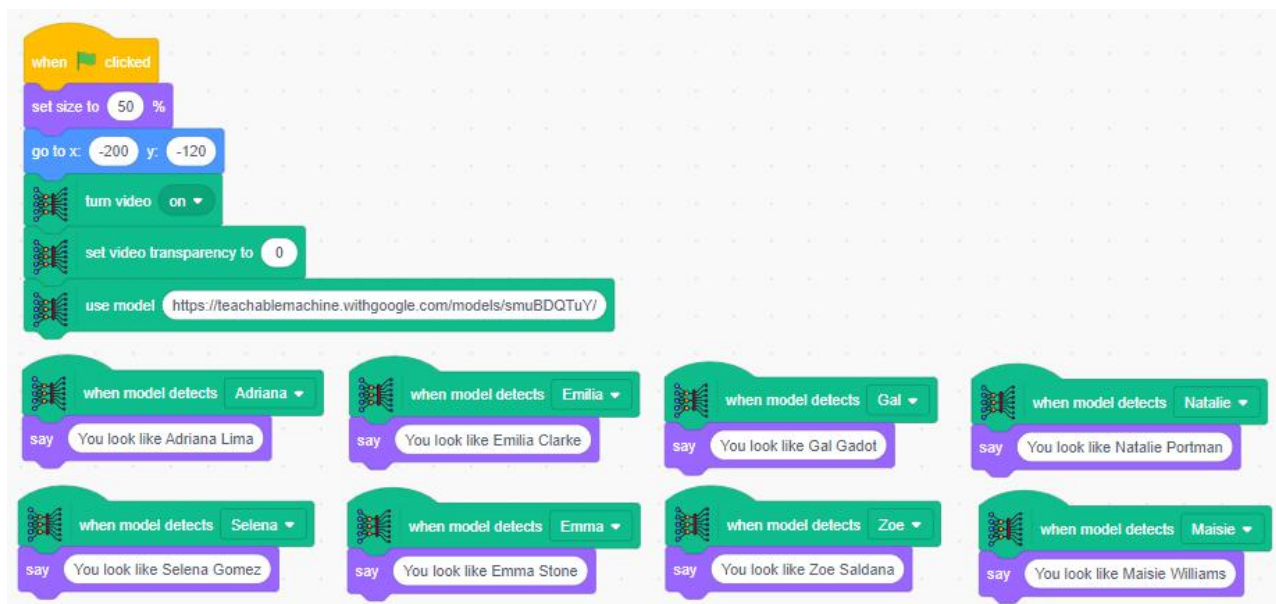




Korak 3b: Zatim ćemo koristiti blok za vrste događaja koji se aktivira uz podudaranje lica. Na njega treba dodati blok reci (say) "Izgledaš kao <jedna od 8 osoba s popisa>".



Korak 4b: Dodajte ovu kombinaciju blokova za svaku poznatu osobu na popisu.





Korak 5b: Priključite/omogućite web kameru, pokrenite kôd i pogledajte tko vam s popisa odgovara!

Korak 6b: Želite li modelirati s vlastitim? Postoji sjajan skup podataka (*dataset*) sa slikama slavnih osoba na: <https://www.kaggle.com/hereisburak/pins-face-recognition>

Možda ćete se morati registrirati na Kaggleu prije nego što ga preuzmete. Ili možete ručno prikupljati slike putem Google pretraživanja. Nakon što prikupite sve slike koje želite - upotrijebite Teachable machine na poveznici: <https://teachablemachine.withgoogle.com/> za treniranje svog modela na način na koji ste to učinili u **programiranju prepoznavanja lica** u prvom dijelu ovog scenarija.

Prepoznavanje lica je tehnologija koja može identificirati ili potvrditi subjekta putem slike, videozapisa ili bilo kojeg audiovizualnog elementa njegovog lica. Općenito, ova se identifikacija koristi za pristup aplikaciji, sustavu ili usluzi. Prije nego što počnemo programirati prepoznavanje lica, moramo prikupiti fotografije određenog lica i trenirati model. Koristili smo aplikaciju pod nazivom Poučna mašina. Teachable Machine je web alat koji stvaranje modela strojnog učenja čini brzim, jednostavnim i dostupnim svima. Jeste li primijetili da je ovaj projekt u potpunosti temeljen na webu i da nije potrebna instalacija softvera?

Danas mnogi potrošači širom svijeta redovito komuniciraju s tehnologijom prepoznavanja lica.

Značajke korištenja prepoznavanja lica

Provjera identiteta:

Identificirajte pojedince i primijenite određena pravila na temelju kategorije u koju spadaju, na primjer, VIP, registrirani posjetitelj, netko s liste blokiranih, zaposlenik ili student. Koristite informacije za poboljšanje i automatizaciju procesa kao što su (1) kontrola pristupa, (2) sigurnosna zaštita, (3) pozdravi kupaca ili posjetitelja i (4) radno vrijeme zaposlenika.

eKYC i prevencija podmetanja:

Potvrdite identitet osobe pomoću fotografije uživo ili snimanja videozapisa skeniranim (i provjerenim) ID-om. To se zove eKYC (elektronički upoznajte svog kupca) i široko se koristi u BFSI-ju ili sličnim slučajevima.

Autorizacija:

Utvrđite nalazi li se ovlaštenu pojedinac u bazi podataka kako bi (1) podigao gotovinu s bankomata, (2) pristupio medicinskom ormariću koji sadrži osigurane lijekove ili (3) otključao skupe strojeve koji zahtijevaju dobro obučene operatere.

Segmentacija i analitika kupaca:

Za pametno oglašavanje analizirajte karakteristike osobe koja stoji ispred digitalnog znaka, kao što su spol, dob i emocije.



Zdravstvene mjere:

Potvrdite da osoba pravilno nosi masku, kao i da provjerite da nema temperaturu prije nego što odobrite pristup zgradi ili restoranu.

Kada se promatraju vertikalna tržišta, 10 industrija ističe se kao zrelo za integraciju prepoznavanja lica i, u mnogim slučajevima, već ga prihvaćaju:

1. Proizvodnja i skladištenje
2. Bankarstvo, financijske usluge i osiguranje (BFSI)
3. Pametni uredi
4. Pametne kuće i stambeni kompleksi
5. Ponavljati
6. Javni prijevoz i zračne luke
7. Zdravstvene ustanove
8. Škole i sveučilišta
9. Gostoprimstvo
10. Restorani i barovi

ZAKLJUČAK

Prepoznavanje lica koristi se u različitim sofisticiranim sustavima, a moguće je izraditi jednostavne primjere u alatima poput Scratcha i PictoBlox.

Promisli!



Što sada znam?



Što sam naučio/la?



Što još želim znati?





Detekcija i klasifikacija objekata za početnike

Detekcija i klasifikacija objekata privukli su veliku pozornost u proteklim desetljećima.

U polju računalnog vida najčešća pitanja vezana su uz razlike između klasifikacije, detekcije i segmentacije objekata na slikama.

Najava cilja nastavnog sata:

Uvod u detekciju objekata i klasifikaciju za početnike kroz praktične primjere primjene.

Počnimo s razumijevanjem što je klasifikacija slike:
Pogledajte donju sliku :



Sufinancirano sredstvima
programa Europske unije
Erasmus+



<http://erasmus-artie.eu>



Odmah ga prepoznajete. To je pas. Vratite se unatrag i analizirajte kako ste došli do ovog zaključka. Prikazana vam je slika i klasificirali ste što je na slici (pas, u ovom slučaju). I to je, ukratko, bit klasifikacije slike.

Kao što ste vidjeli, ovdje je samo jedan objekt - pas. Lako možemo klasificirati model i predvidjeti da na zadanoj slici postoji pas. Ali što ako imamo i mačku i psa na jednoj slici?



U tom slučaju možemo trenirati višestruki klasifikator. Ali i dalje ne znamo lokaciju životinje ili objekta na slici. Tu dolazimo do lokalizacije objekta na slici. Ona nam pomaže identificirati točnu poziciju objekta na slici. U slučaju da imamo više prisutnih objekata, tada se moramo osloniti na koncept **detekcije objekata**. Ona nam služi da predvidimo točnu lokaciju i klasu svakog objekta.



Classification



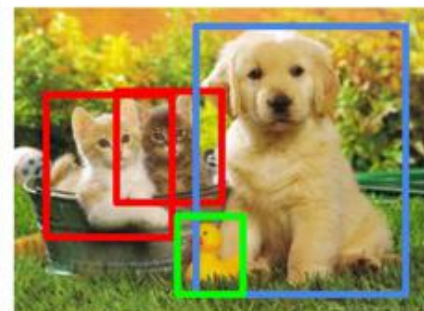
CAT

Classification + Localization



CAT

Object Detection



CAT, DOG, DUCK

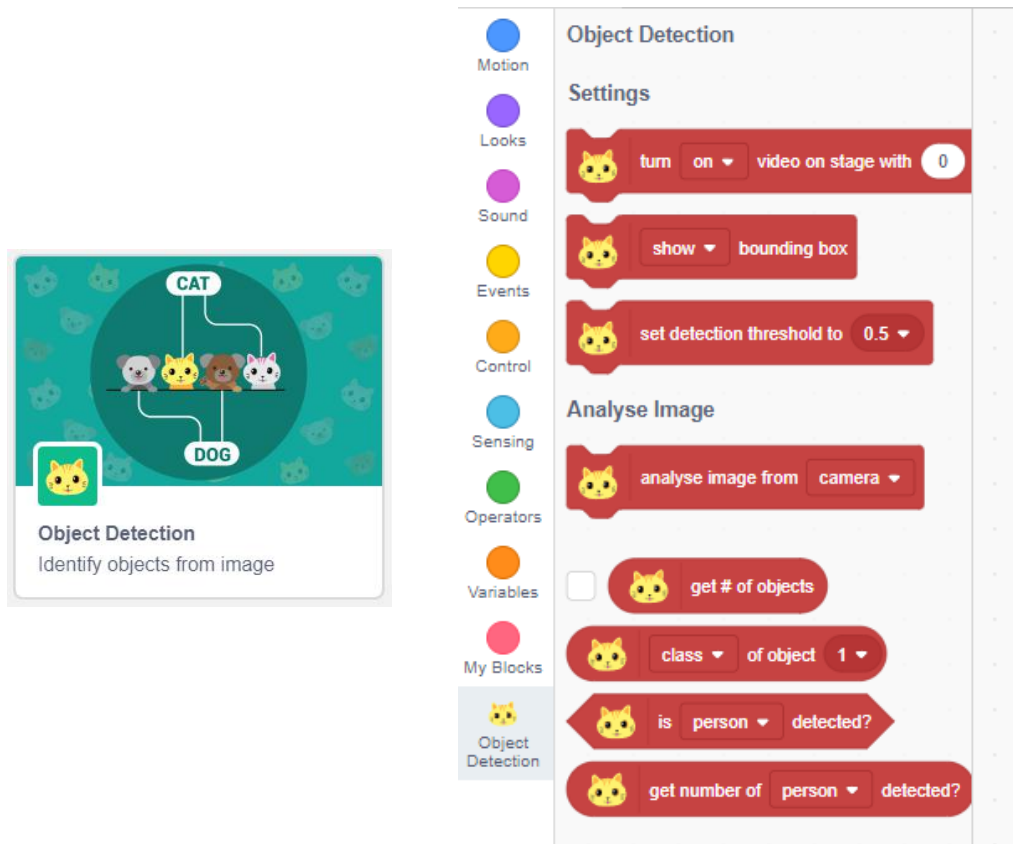


Aplikacije za detekciju objekata

PictoBlox je trenutno jedina aplikacija s mogućnošću detekcije objekata. To je računalni program i prvo ga morate instalirati s poveznice <https://thetempedia.com/product/pictoblox/download-pictoblox/> (577 Mb)

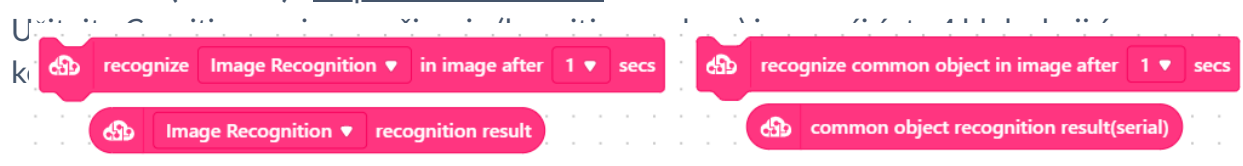
Korak 1: Nakon instalacije učitajte proširenje za detekciju objekata (*Object Detection*)

Korak 2: Ovdje su blokovi dostupni za programiranje.



Prva 3 bloka su postavke kamere, a ostali se koriste za analizu i prikazivanje rezultata. Situacija s klasifikacijom objekata puno je bolja jer imamo više aplikacija za korištenje (osim prethodno spomenutog PictoBlox)

Makeblock (mBlock) - <https://ide.mblock.cc/>





Makeblock također ima svoje Teachable Machine proširenje (različito od Google-ovog) gdje možete trenirati do 3 klase i koristiti ga za klasifikaciju objekata.

Scratch (ML4KIDS) - <https://machinelearningforkids.co.uk/scratch3/>

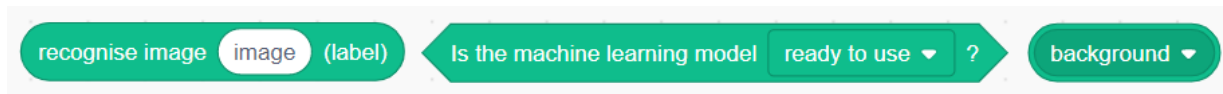
Učitajte proširenje *Imagenet*.



U njemu su nam dostupna 3 bloka. Koristite ga u kombinaciji s proširenjem za *Video sensing* proširenjem kako biste uključili/isključili sliku s kamere i postavili njenu prozirnost. Model je treniran je za prepoznavanje oko tisuću uobičajenih predmeta (objekata) i temelji se na *MobileNetu* (ML modelu dizajniranom za mobilne uređaje, tako da ne zahtijeva puno računalne snage).

Popis svih objekata *MobileNet* modela dostupan je ovdje:

<https://storage.googleapis.com/download.tensorflow.org/data/ImageNetLabels.txt>



Scratch (MITMEDIALAB) - <https://mitmedialab.github.io/prg-extension-boilerplate/create/>

Učitajte *Teachable Machine* proširenje



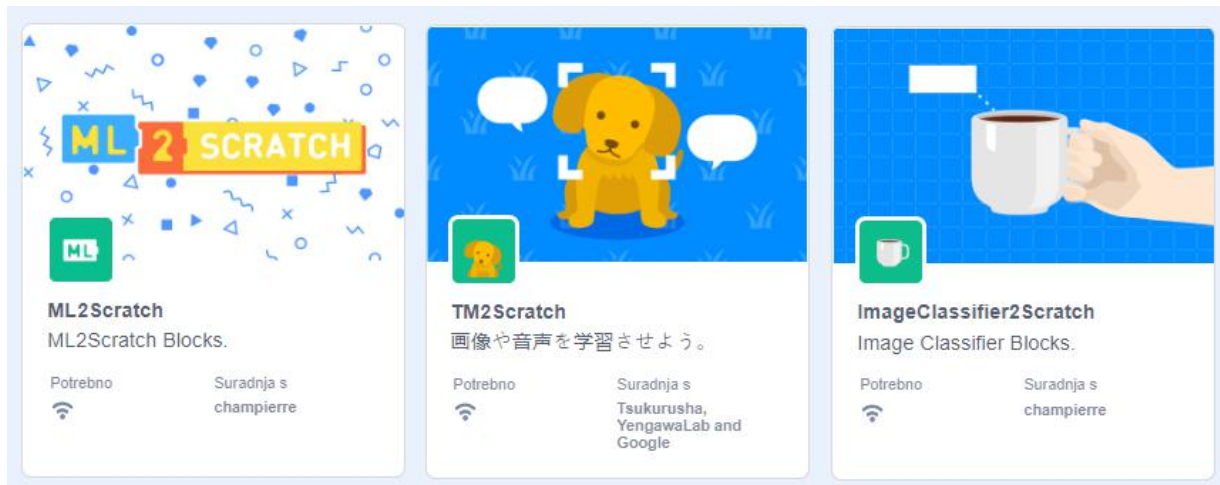


Koristite ove blokove u kombinaciji s *Google Teachable machine*.



Stretch3 (github.io) - <https://stretch3.github.io/>

Učitajte proširenja *ML2Scratch*, *TM2Scratch* i *ImageClassifier2Scratch* za korištenje većeg broja blokova za klasifikaciju objekata i treniranje.

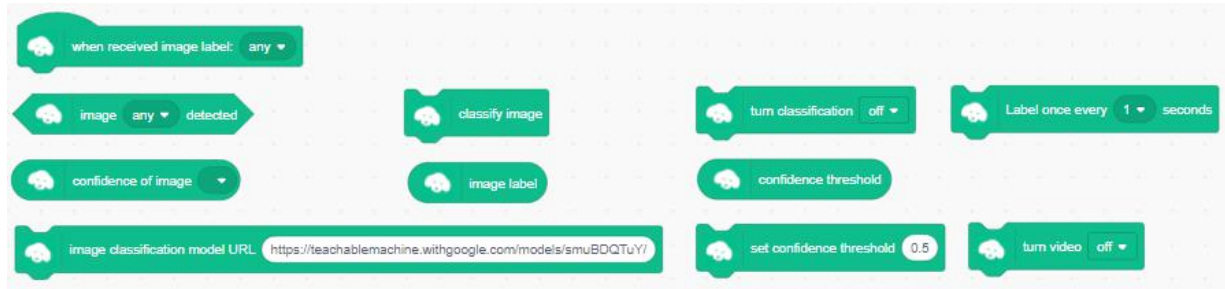


ML2Scratch blokovi





TM2Scratch blokovi

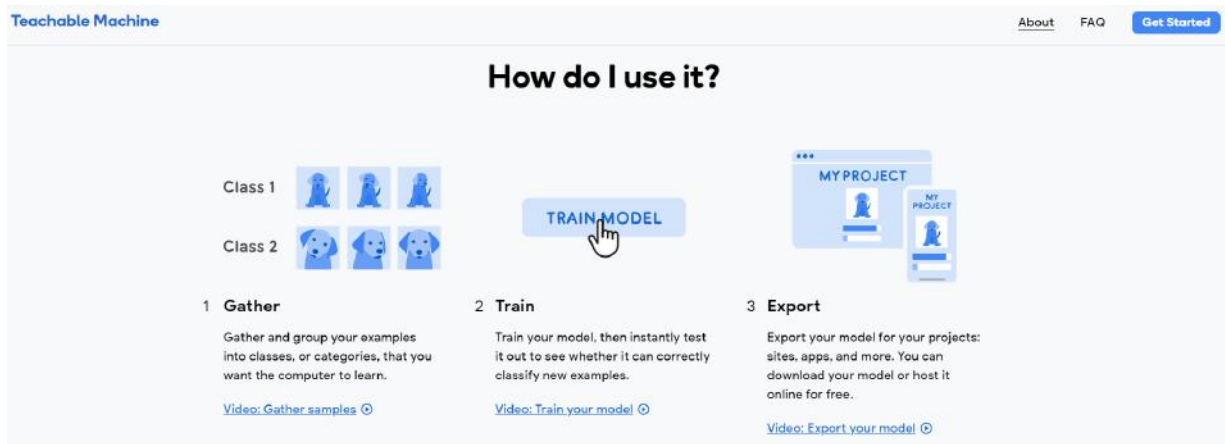


ImageClassifier2Scratch blokovi



Teachable Mchine (Google) - <https://teachablemachine.withgoogle.com/>

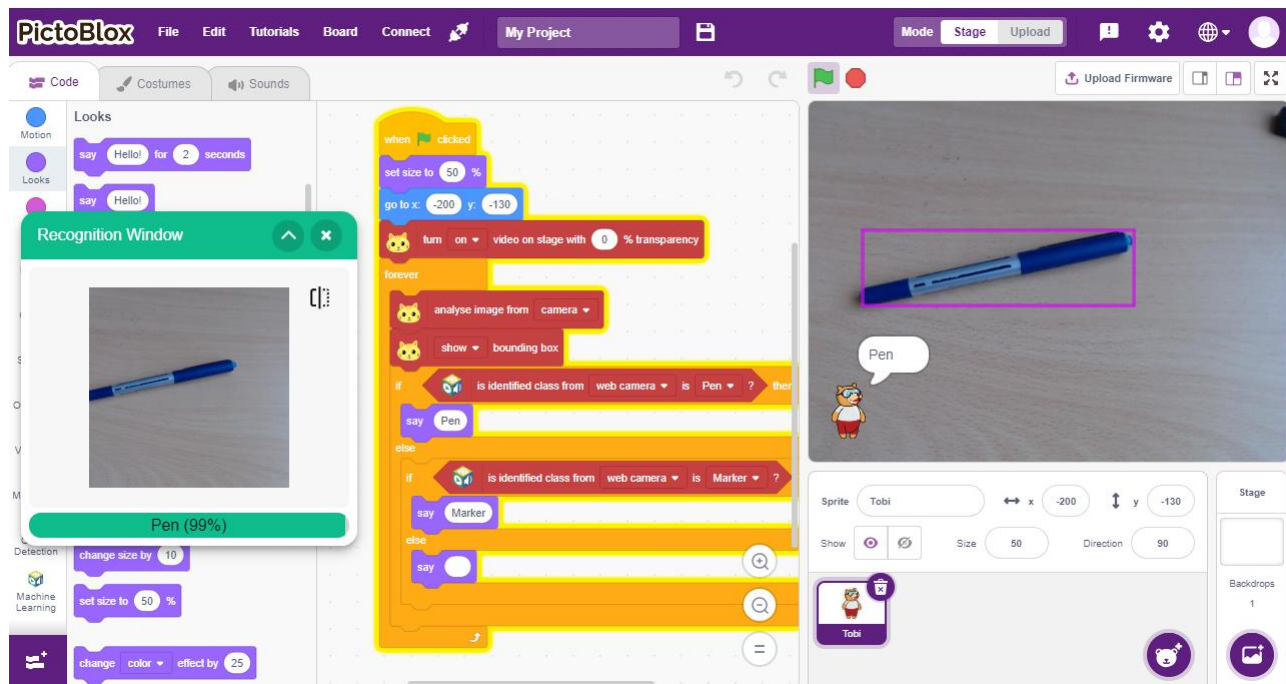
Ova se aplikacija koristi za treniranje vašeg modela i njegovo korištenje za prepoznavanje objekata u kombinaciji s Teachable Machine proširenjima dostupnim u **PictoBloxu**, **Scratchu** (**MITMEDIALAB**) i **Stretch3**



Jedan primjer za detekciju objekata - olovka ili marker. Teacahable Machine koristi se za obuku 2 klase. Poveznica na trenirani model je:

<https://teachablemachine.withgoogle.com/models/FdWn0CA2a/>

Proširenja u PictoBloxu koja ćemo koristiti u ovom primjeru su *Object Detection* i *Machine Learning*.



Detekcija i klasifikacija objekata dva su ključna zadatka za analizu slike. Detekcija objekata na slici zahtijeva kombiniranje mnogo različitih signala iz neobrađenih slikovnih podataka. Često se koriste dvije vrste informacija: lokalna pojava koja opisuje sam objekt i globalni prikaz koji bilježi informacije specifične za sliku. Ove dvije vrste informacija često se koriste u dva zadatka: detekcija i klasifikacija objekata.

Klasifikacija

Klasifikacija je zadatak strojnog učenja za određivanje objekata na slici ili videozapisu. Odnosi se na trenirane modele strojnog učenja koji prepoznaju prisutnost klasa (objekata).

Klasifikacija je korisna na razini da-ne odlučivanja sadrži li slika objekt odnosno anomaliju ili ne. Zaseban zadatak od klasifikacije je lokalizacija ili određivanje položaja klasificiranih objekata na slici ili videu.





Detekcija objekata

Otkrivanje objekata kombinira klasifikaciju i lokalizaciju kako bi se utvrdilo koji se objekti nalaze na slici ili videozapisu te odredilo gdje se nalaze na slici. Klasifikacija se primjenjuje na različite objekte i za svaki se koristi granični okvir. Detekcija objekata koristi se za prepoznavanje objekata na slici ili u videozapisu. Slučajevi uporabe za detekciju objekata uključuju detekciju lica s dodatnim analizama, kao što je na primjer izraz lica, procjena dobi ili detekcija pospanosti osobe. Postoje mnoge aplikacije za detekciju objekata u stvarnom vremenu za upravljanje prometom, kao što su sustavi za detekciju vozila koji se primjenjuju u raznim situacijama u prometu.

Kao što je gore opisano, najpopularniji pristupi računalnom vidu su klasifikacija i detekcija za identifikaciju objekata prisutnih na slici i određivanje njihovog položaja.

ZAKLJUČAK

Detekcija i klasifikacija objekata dva su ključna zadatka za analizu slike.

Promisli!



Što sada znam?



Što sam naučio/la?



Što još želim znati?



Programiranje detekcije objekata u Scratchu

Što je detekcija objekata?
Kako funkcionira detekcija objekata?

Detekcija objekata je tehnika računalnog vida čija je svrha identificirati i locirati objekte unutar slike ili videozapisa. Naime, detekcija objekata crta granične okvire oko detektiranih objekata, što nam omogućuje da lociramo gdje se navedeni objekti nalaze (ili kako se kreću) u određenoj sceni. Prije nego što počnemo programirati detekciju objekata, moramo prikupiti određene fotografije i istrenirati model. Koristit ćemo aplikaciju pod nazivom Teachable machine. Teachable Machine je web alat koji stvaranje modela strojnog učenja čini brzim, jednostavnim i dostupnim svima.

U nastavku ćemo upoznati programe za detekciju objekata i njegove uporabe kroz primjere.



Sufinancirano sredstvima
programa Europske unije
Erasmus+



ZADATAK: Mačka ili pas?

Stvorite model i program koji će otkriti je li mačka ili pas u slici s kamere.

WEB APLIKACIJA (nema instalacije softvera)

Korak 1: Otvorite web-preglednik, odaberite i preuzmite slike s: <https://bit.ly/cats-image-dataset>

Koristit će se za treniranje klase 1

Korak 2: otvorite web-preglednik, odaberite i preuzmite slike s: <https://bit.ly/dogs-image-dataset>

Koristit će se za treniranje klase 2

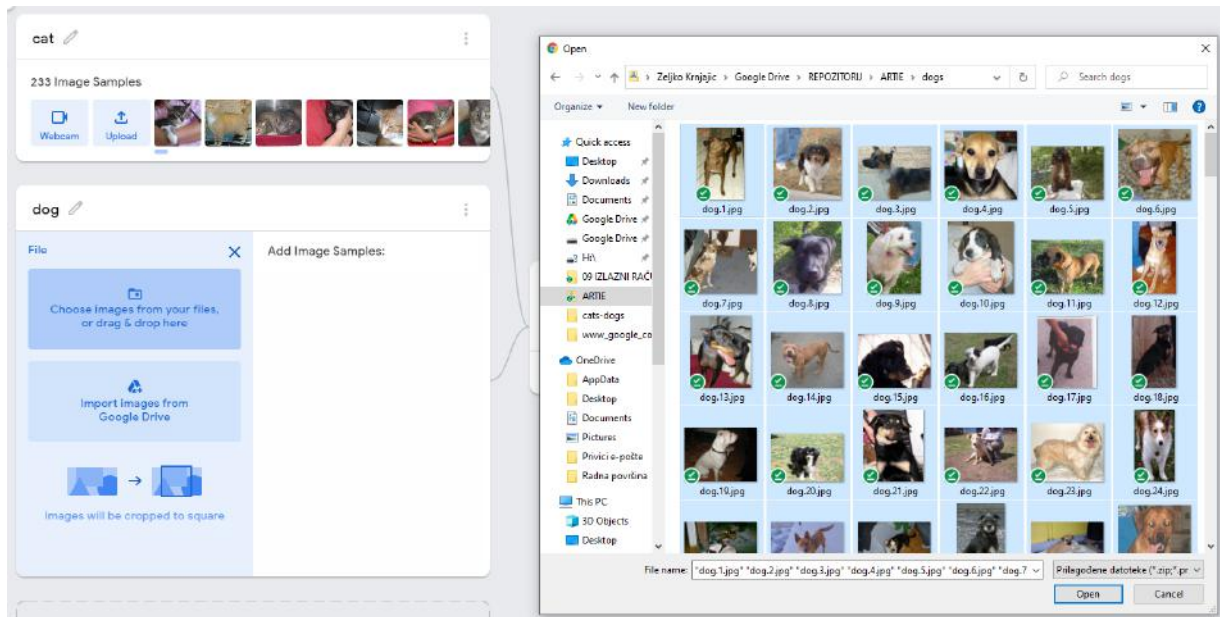
Korak 3: Otvorite web preglednik i idite na poveznicu: <https://teachablemachine.withgoogle.com/>

Korak 4: Kliknite na *Get started*.

Korak 5: Birajte *Image project*

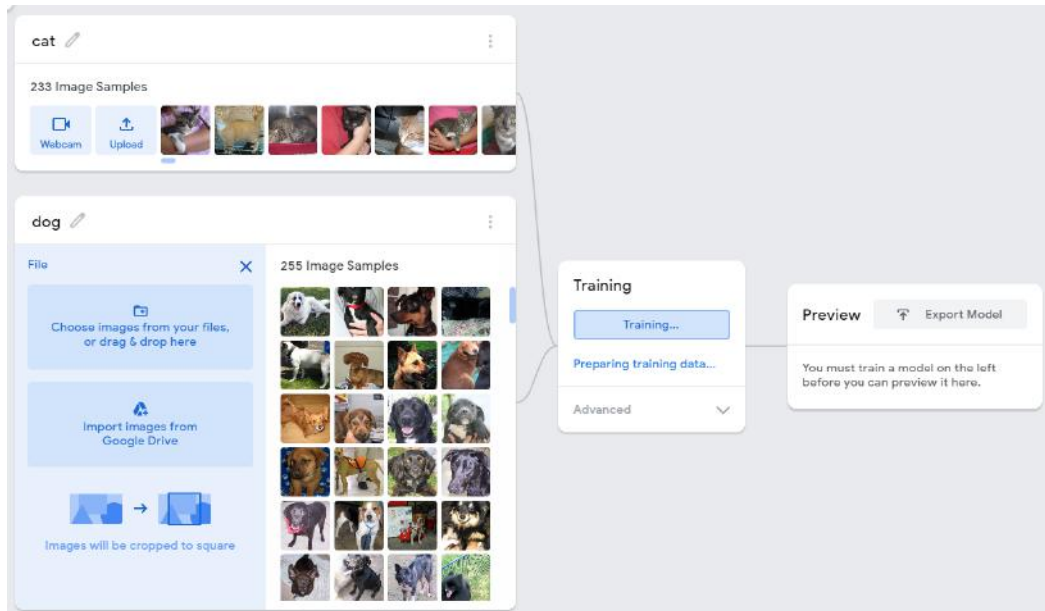
Korak 6: Birajte *Standard image model*

Korak 7: Promijenite imena klase 1 u mačka i klase 2 u psa. Prenesite slike mačaka u datoteke mačaka i slike pasa u datoteke pasa kao što je prikazano na slici ispod

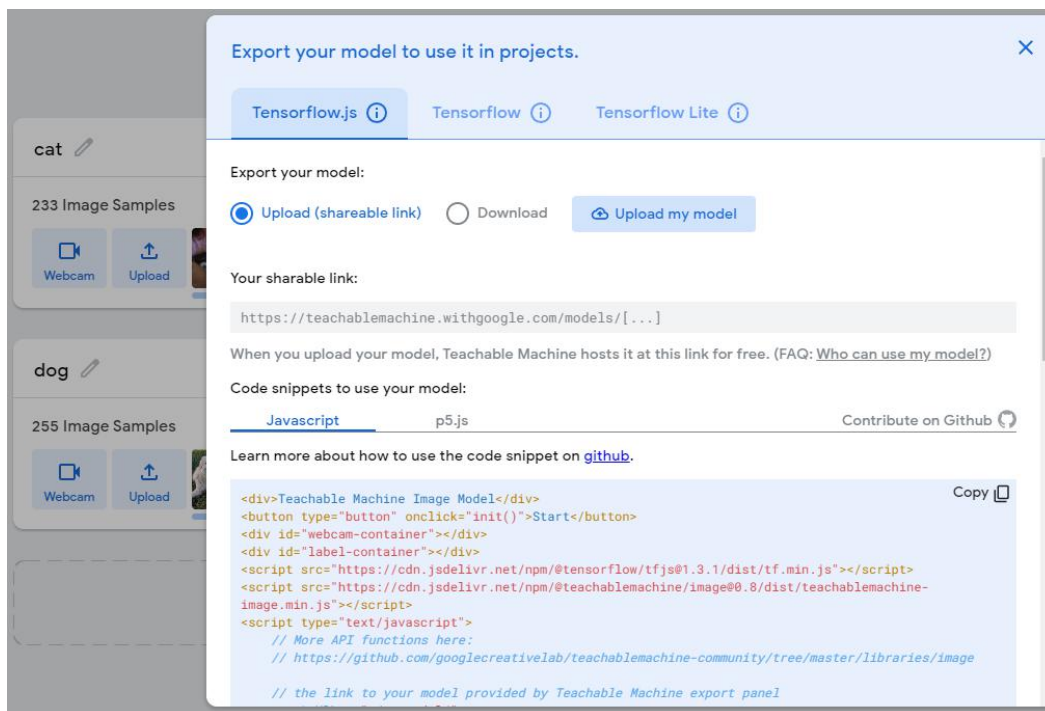




Korak 8: Trenirajte svoj model. Nemojte mijenjati kartice preglednika tijekom treniranja modela.



Korak 9: Izvezite svoj model. Na skočnom prozoru odaberite *Upload my model* (treća opcija - Prenesi moj model u oblak) i Google će pohraniti vaš model besplatno.





Korak 10: Kopirajte poveznicu dobivenu u donjem tekstualnom polju - ovo je poveznica vašeg modela. U našem slučaju bila je <https://teachablemachine.withgoogle.com/models/gS4NT1mgE/>

☒ Upload (shareable link) ☐ Download

Your sharable link:

<https://teachablemachine.withgoogle.com/models/gS4NT1mgE/>

Copy

When you upload your model, Teachable Machine hosts it at this link for free. (FAQ: [Who can use my model?](#))

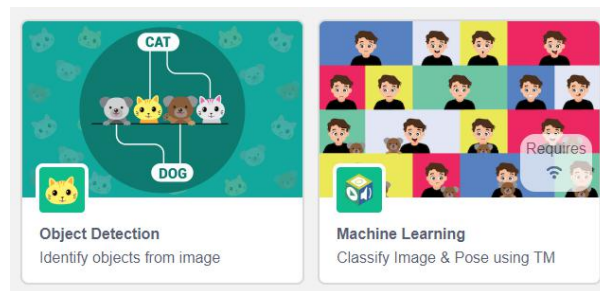
✓ Your cloud model is up to date.

Korak 11: Vaš je model spreman za uporabu

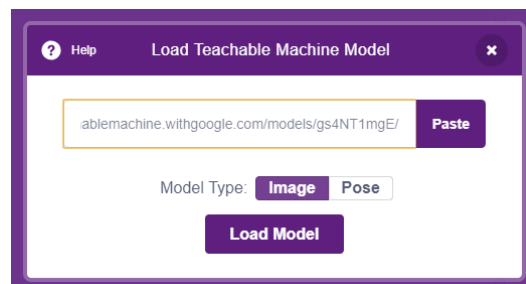
PICTOBLOX (Desktop aplikacija):

Korak 1a: Preuzmite i instalirajte **PictoBlox** jer je trenutno jedini ima podršku za detekciju objekata. To je računalni program i prvo ga morate instalirati s poveznice <https://thetempedia.com/product/pictoblox/download-pictoblox/> (577 Mb)

Korak 2a: Učitajte proširenja *Object Detection* i *Machine Learning*.



Korak 3: Kliknite na grupu blokova *Machine learning* i odaberite *Load model* (učitaj model). Zalijepite poveznicu vašeg modela dobivenog od *Teachable Machine-a* (vidi dolje): <https://teachablemachine.withgoogle.com/models/gS4NT1mgE/> i kliknite na *Load Model*.





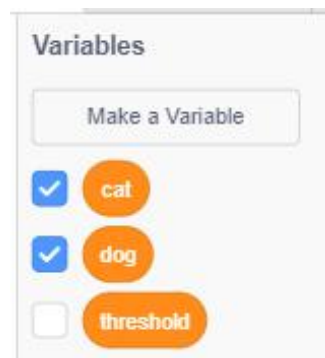
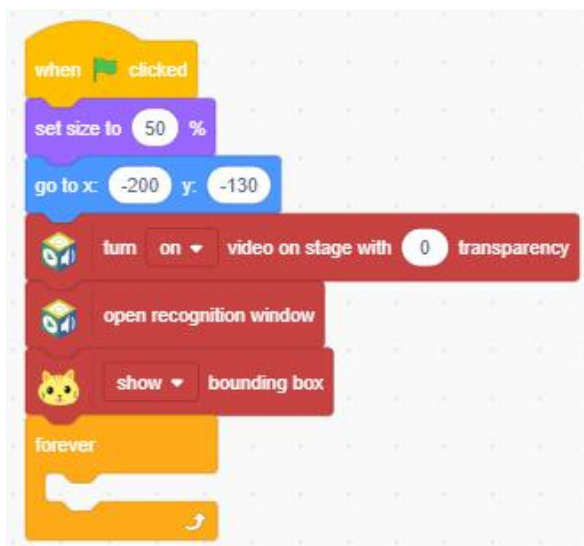
Korak 4a: Istovremeno ćemo izvoditi klasifikaciju i detekciju objekata. Prvo moramo pokrenuti program, promijeniti veličinu i pomaknuti Tobi lik (*sprite*) u donji lijevi kut. Slijedećih nekoliko blokova iz Machine Learning proširenja koriste se za uključivanje videa i otvaranje prozora za prepoznavanje (*open recognition window*). Ako je videozapis preokrenut na bloku birajte preokrenuti video (*turn on flipped video on stage*) s 0% prozirnosti (*transparency*).



Korak 5a: Iz proširenja za detekciju objekata (*object detection*) koristite *show bounding box* blok za prikaz položaja objekta u slici s kamere i dodajte blok za beskonačnu petlju (*forever*).



Korak 6a: Napravite 3 varijable **mačka (cat)**, **pas (dog)** i **prag (threshold)**. Označite varijable **mačka** i **pas** za prikaz na pozornici (*stage*). Ostavite varijablu **prag** neoznačenu.

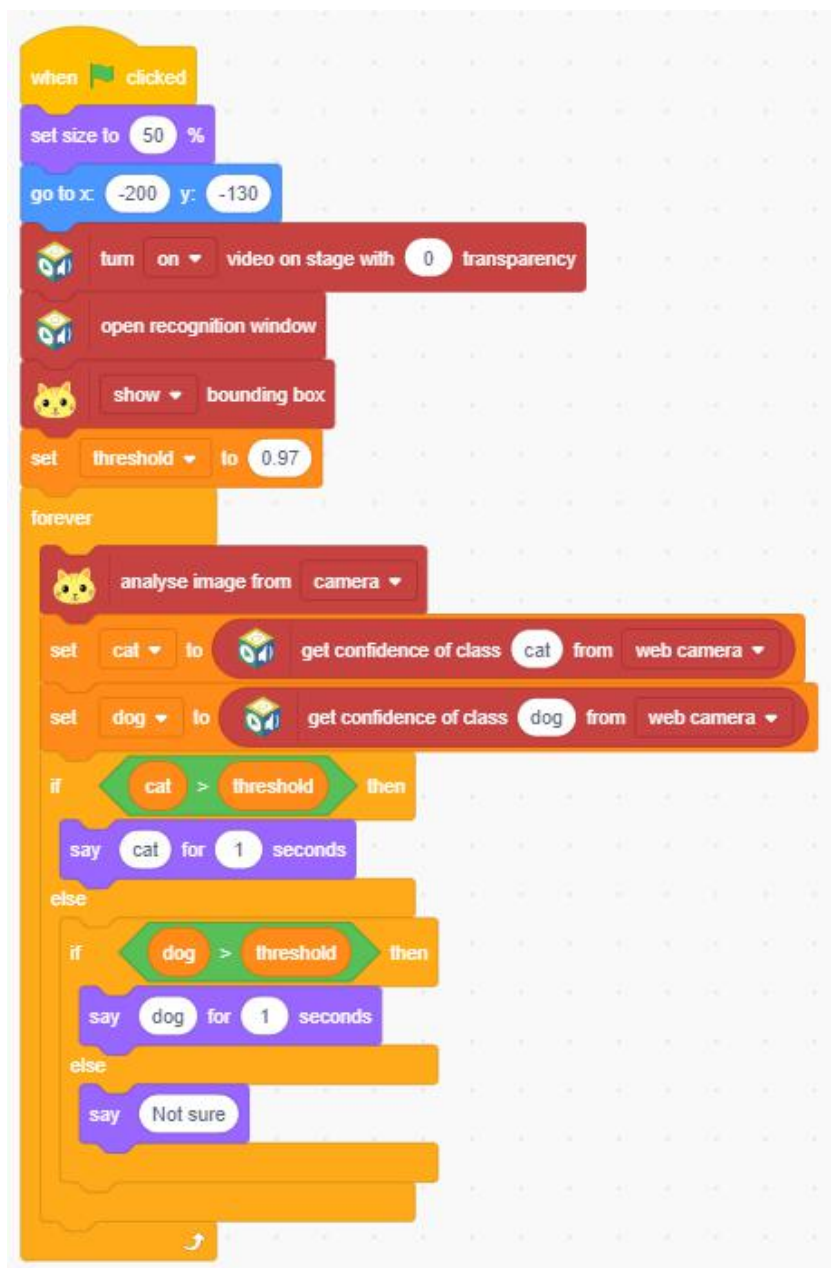




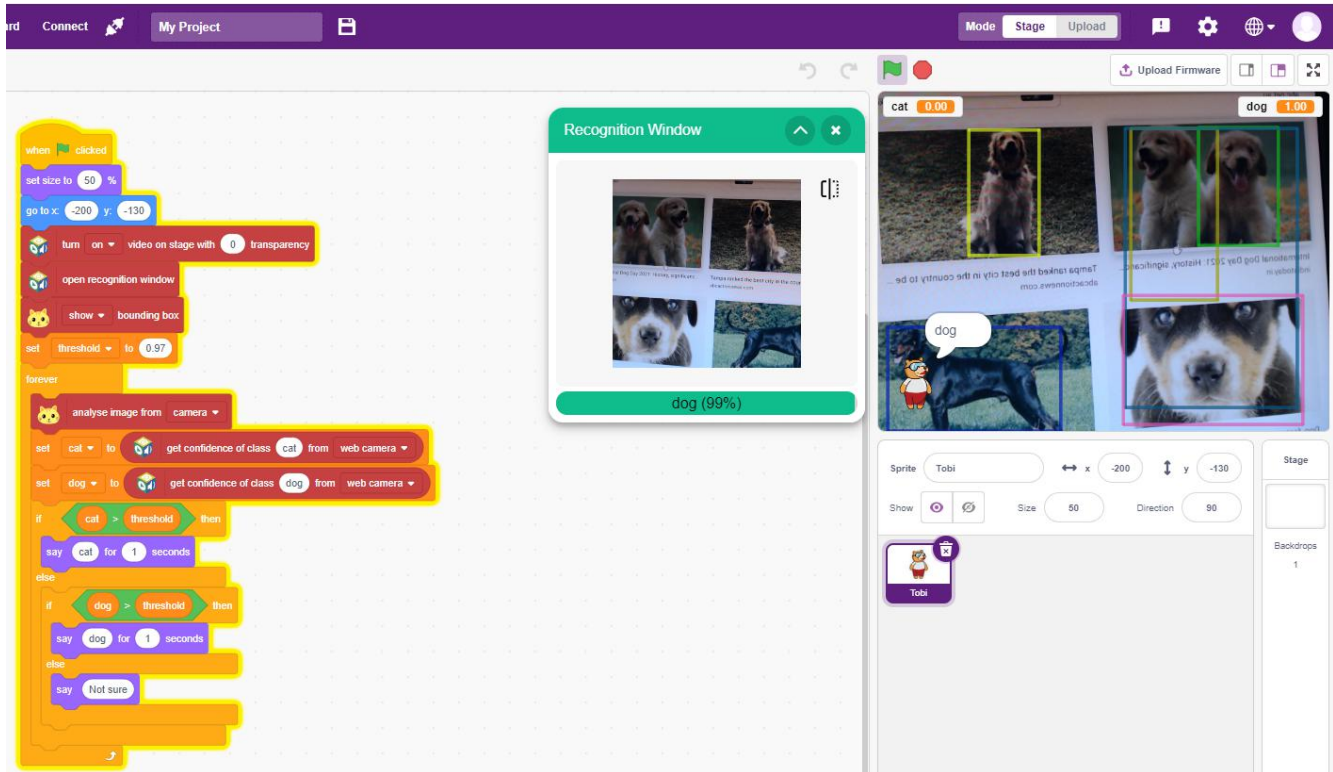
Korak 7a: Prvo s blokom postavite vrijednost varijable **prag** na **0,97** neposredno prije beskonačne petlje. A sada slijedi glavni dio u kojem analiziramo sliku iz kamere (*analyse image from camera*) i pohranjujemo vrijednost pouzdanosti za svaku klasu u pripadajuće varijable (*get confidence of class ... from web camera*).



Korak 8a: Slijedi dvostruki ako-onda blok s ispitivanjima pouzdanosti za svaku kasu, nakon čega se u balončiću za govor (*say ... for 1 seconds*) ispisuje "mačka" (*cat*) ili "pas" (*dog*). Ako je vrijednost pouzdanosti za obje klase ispod postavljenog praga, tekst govornog balončića je "Nisam siguran" (*Not sure*).



Korak 9a: Pokrenite program i testirajte ga na vašoj mački ili psu. Ako ih nemate, na raspolaganju su vam rezultati pretrage za Google slike (*Google images*), samo trebate na tražilici naći slike pasa ili mačaka te usmjeriti svoju web kameru na zaslon s rezultatima pretrage. Vidjet ćete granične okvire oko detektiranih objekata. Pokušajte smanjiti ili povećati vrijednost praga ili pokušajte zbuniti algoritam otkrivanja s nekim slikama specifičnih pasmina mačaka ili pasa.



Dakle, što je točno detekcija objekata?

Da bismo odgovorili na to pitanje, počnimo s prvo klasifikacijom slika. U ovom zadatku imamo sliku i želimo je dodijeliti jednoj od mnogih različitih kategorija (npr. automobil, pas, mačka, čovjek,...), što znači da u osnovi želimo odgovoriti na pitanje "Što je na ovoj slici?". Imajte na umu da je jednoj slici dodijeljena samo jedna kategorija. Jednostavnim riječima, detekcija objekata je tehnika klasifikacije slika, a osim što klasificira, ova tehnika identificira i poziciju objekta pronađenog iz velikog broja unaprijed definiranih kategorija.

ZAKLJUČAK

Detekcija objekata je tehnika klasifikacije slika, a osim što klasificira, ova tehnika identificira i poziciju objekta pronađenog iz velikog broja unaprijed definiranih kategorija.

Promisli!



Što sada znam?



Što sam naučio/la?



Što još želim znati?



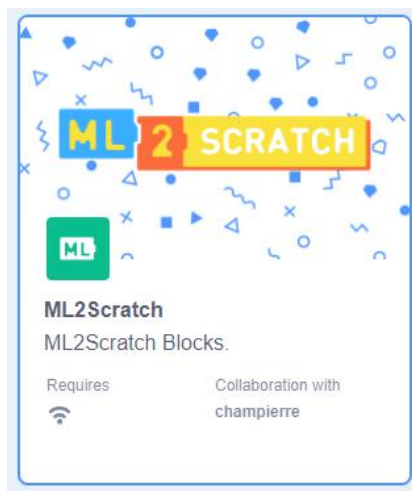
Projekt s klasifikacijom objekata

Što je otkrivanje objekata?
Kako funkcionira otkrivanje objekata?

U nastavku ćemo upoznati postupak klasifikacije objekata i njegove uporabe kroz projekt s praktičnim radom.

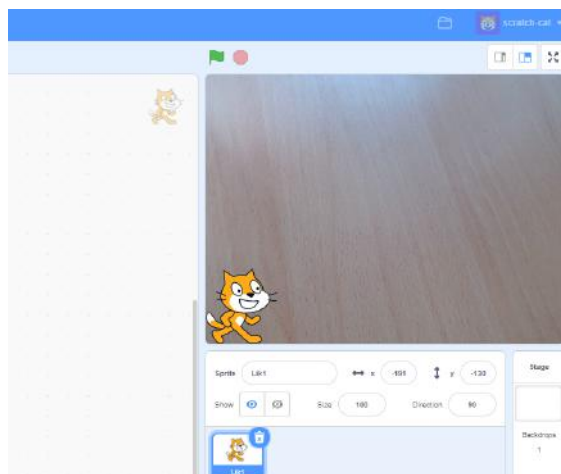
Korak 1: Spojite web-kameru na računalo

Korak 2: Otvorite Scratch sučelje na <https://stretch3.github.io/> i učitajte proširenje "ML2SCRATCH"

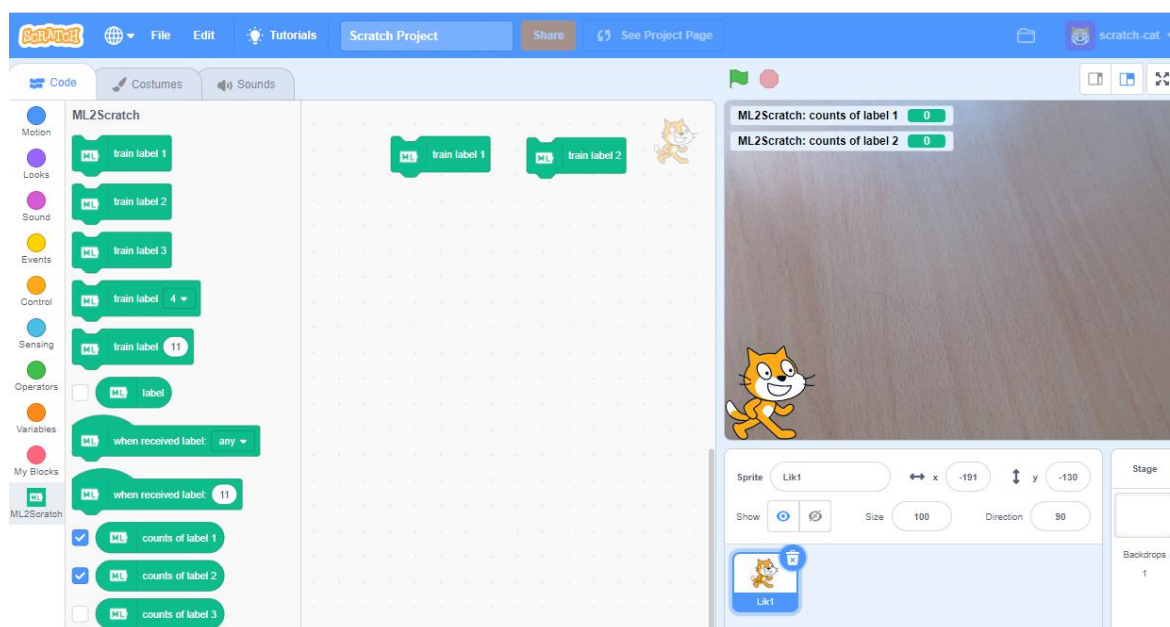




Korak 3: Najprije pospremite površinu stola i priključite te usmjerite svoju web kameru na stol tako da se na njemu ne vide nikakvi predmeti, a zatim u aplikaciji premjestite lik mačke u kut kako je prikazano.



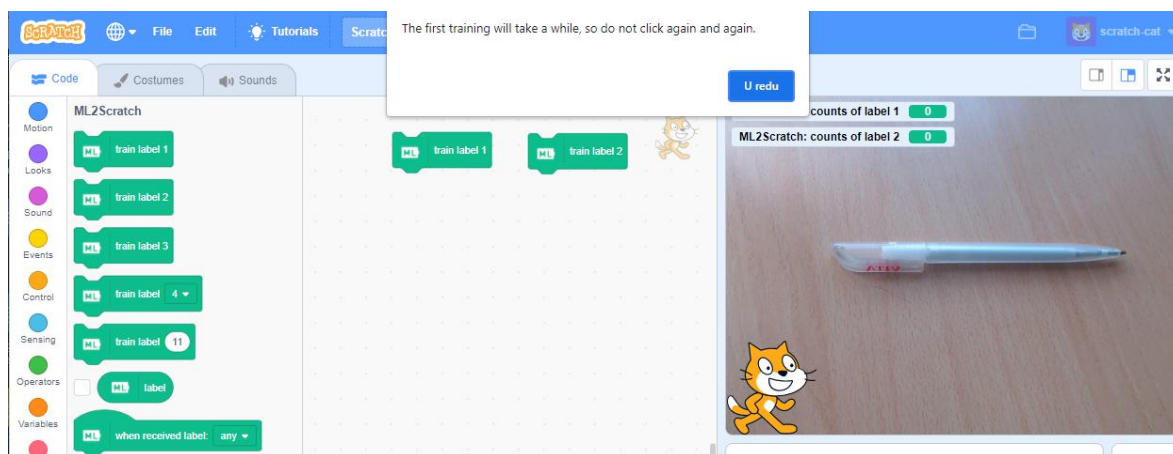
Korak 4: Iz grupe blokova ML2SCRATCH odaberite *train label 1* i *train label 2* blokove i stavite ih na programsko područje i označite za prikaz blokove *counts of label 1* i *counts of label 2* kako je prikazano na slici ispod.



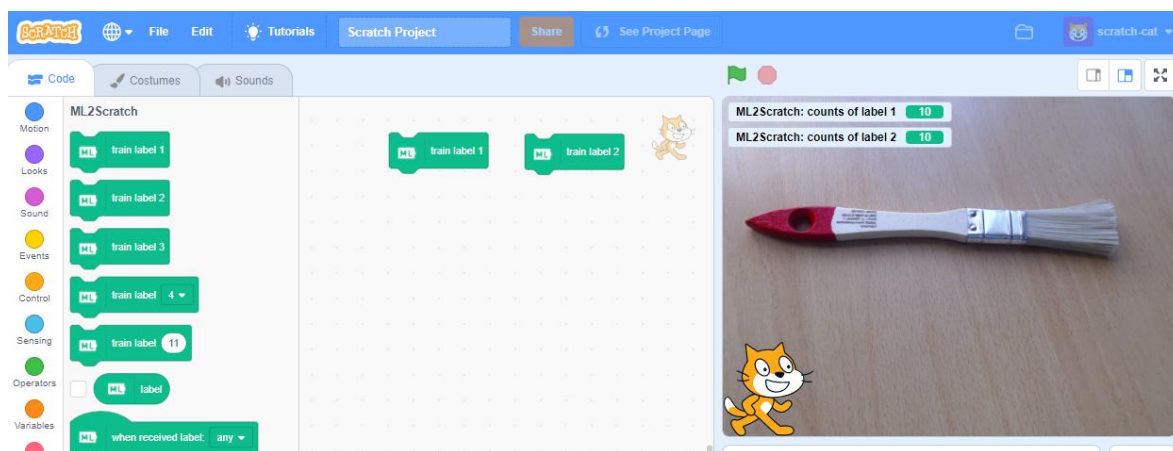


Korak 5: Pripremite dvije skupine objekata za strojno učenje. U našem slučaju kistovi i olovke koristit će se za treniranje oznaka (*label*)

Korak 6: Postavite prvi objekt iz prve grupe predmeta na područje stola u koje je web kamera usmjerena, a zatim kliknite na train label 1 blok - dobiti ćete obavijest da malo pričekate kao što je prikazano i broj oznaka (*counts of label 1*) će se promijeniti na 1

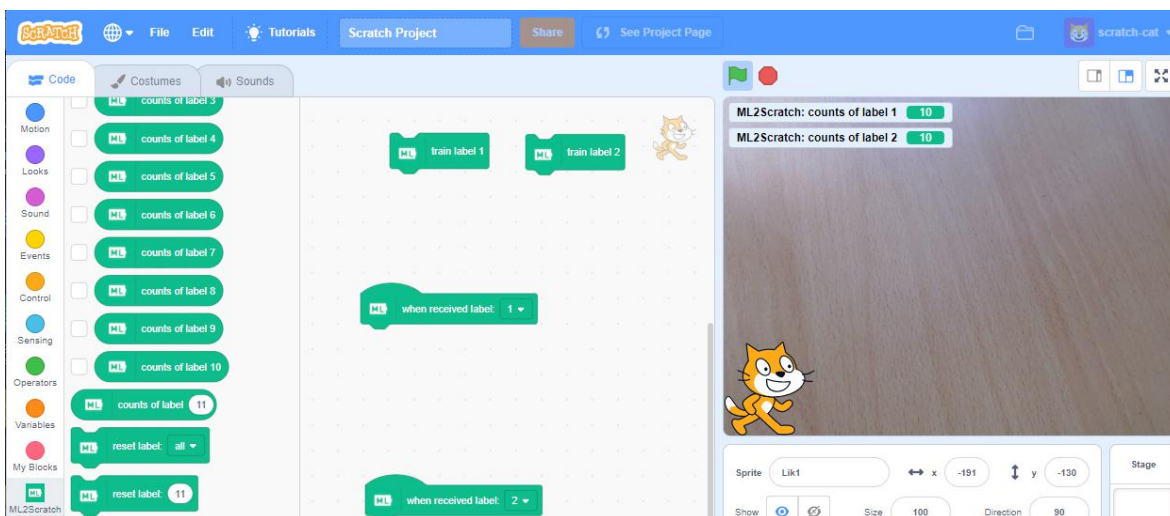


Korak 7: Trenirajte oznake s otprilike 10 slika svakog objekta mijenjajući im položaj. Vodite računa da trenirate ispravnu oznaku - ne miješajte predmete i oznake.

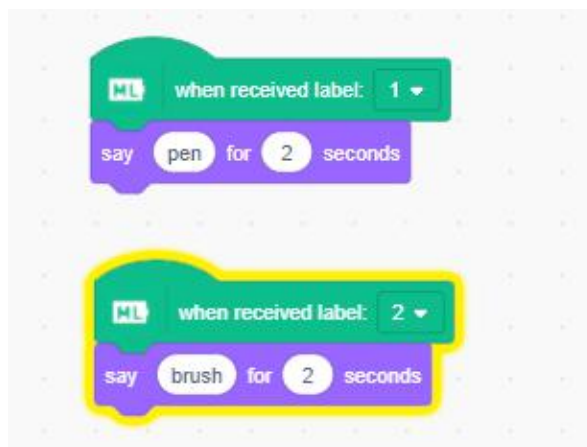




Korak 8: Sada na područje za programiranje postavite dva bloka *when received label: any* i s padajućeg izbornika na sami blokovima promijenite *any* u 1 na prvom, a zatim u 2 na drugom bloku kao što je prikazano.



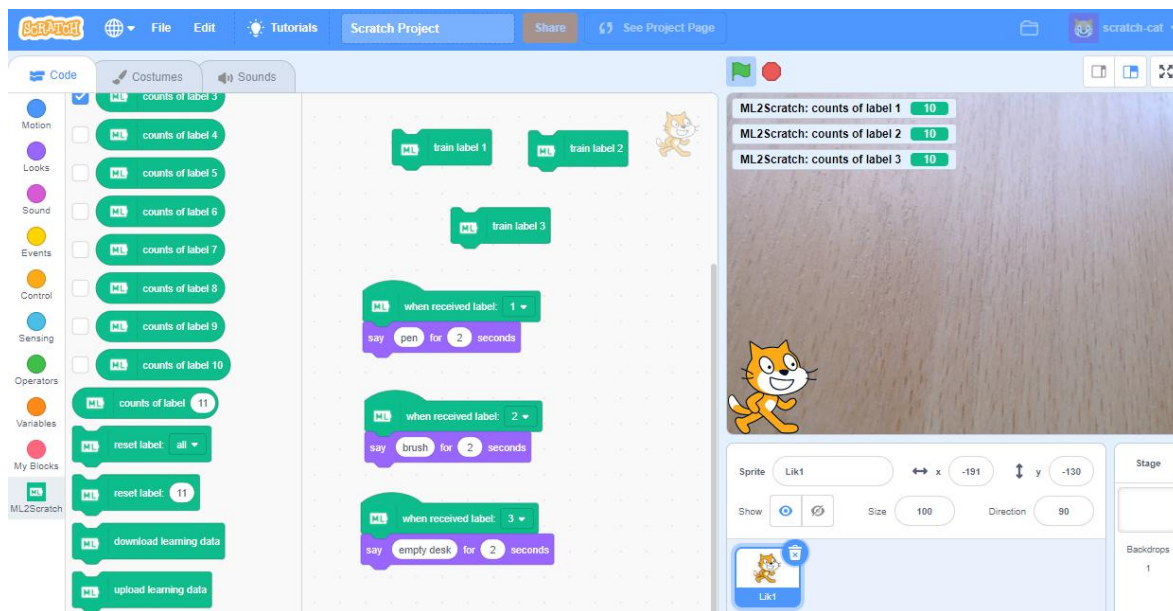
Korak 9: Iz grupe Looks (*Izgledi*) odaberite dva bloka *say Hello! for 2 seconds* i promijenite tekst "Hello!" u "Olovka" za prvu oznaku, a zatim promijenite tekst "Hello!" u "Kist" za drugu oznaku kao što je prikazano





Korak 10: Na dio stola nasumično postavljajte objekte iz dviju grupa korištenih za treniranje i pratite što se događa. Radi li program ispravno? Što se događa kad na stolu nema objekata? Kako riješiti problem s praznim stolom?

Korak 11: Rješenje: Upotrijebite treću oznaku (*train label 3*), trenirajte ju na praznom stolu i dodajte treći blok *when received label: 3* i dodajte na njega blok *say Nema objekta for 2 seconds* kao što je prikazano

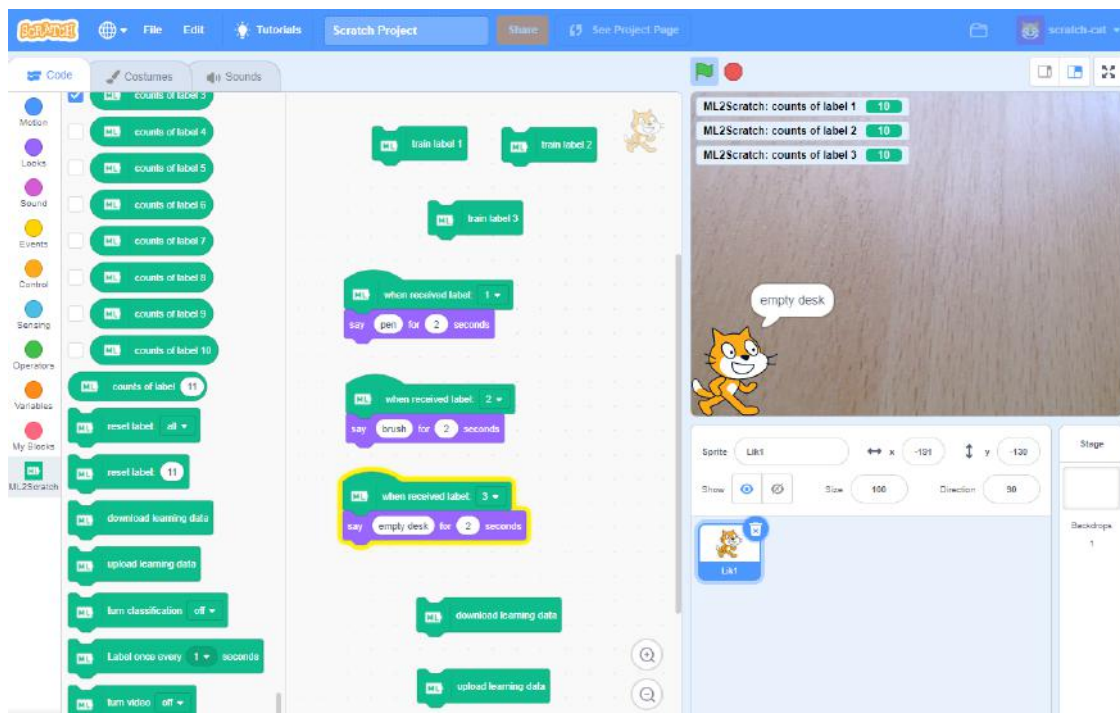


Korak 12: Razgovarajte o točnosti klasifikacije. Pokušajte trenirati svaku od oznaka s dodatnim predmetima i usporedite s prethodnim rezultatima. Poboljšava li se točnost klasifikacije?

Korak 13: Upotrijebite više grupa objekata za treniranje više oznaka. Preuzmite i prenesite trenirani podtkovni model s blokovima *download learning data* i *upload learning data* (kliknete na odgovarajući blok da biste spremili ili prenijeli .json datoteku)



ARTIE



ZAKLJUČAK

Klasifikacija slike je tehnika koja se koristi za klasificiranje ili predviđanje klase određenog objekta na slici. Glavni cilj ove tehnike je točno identificirati značajke na slici.

Promisli!



Što sada znam?



Što sam naučio/la?



Što još želim znati?



Sufinancirano sredstvima
programa Europske unije
Erasmus+



<http://erasmus-artie.eu>



Prepoznavanje i generiranje govora za početnike u Scratchu

Uvod u prepoznavanje govora

Prepoznavanje govora je sposobnost prevođenja diktata ili izgovorene riječi u tekst. Poznat je i kao govor-u-tekst (*Speech-To-Text STT*) i prepoznavanje glasa (*Voice Recognition*).

Prepoznavanje se postiže slijedom određenih radnj, a program koji vrši prepoznavanje govora poznat je kao "Sustav prepoznavanja govora" (*Speech Recognition system*). Ovi sustavi obično se implementiraju u obliku programa za diktiranje i inteligentnih asistenata u osobnim računalima, pametnim telefonima, web preglednicima i mnogim drugim uređajima.

Uvod u generiranje govora

Generiranje ili sinteza govora (skraćeno i kao TTS, što je kratica od engleskih riječi [Text-to-Speech](#)), za razliku od prepoznavanja govora, nije tehnologija koja iskorištava glas, ona ga proizvodi. Sintetizirani govori općenito su završna faza globalnih procesa koji postaju sve demokratskiji zbog važnosti cjelokupnog iskustva vezanog uz pojam "glasa".

Sinteza govora (TTS) definira se kao umjetna proizvodnja ljudskih glasova. Glavna uporaba (i ono što je potaknulo stvaranje te sinteze) je sposobnost automatskog prevođenja teksta u izgovorenu riječ.

Uvod u prepoznavanje i generiranje govora za početnike kroz jednostavan primjer programa u Makeblock aplikaciji.



Sufinancirano sredstvima
programa Europske unije
Erasmus+

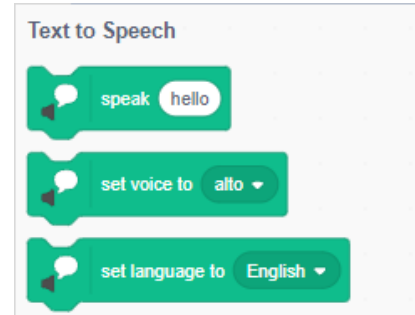


Aplikacija

Scratch (MIT)

<https://scratch.mit.edu/projects/editor/>

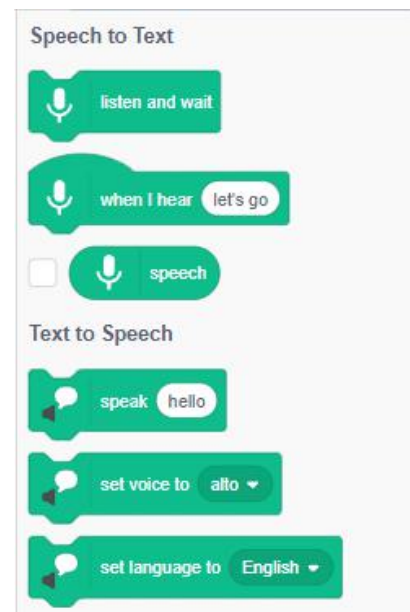
Dostupno je samo proširenje "Text-to-Speech" (3 bloka)



Scratch (ML4KIDS)

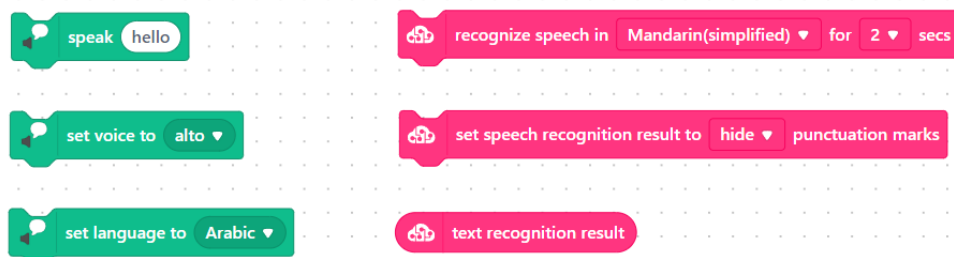
<https://machinelearningforkids.co.uk/scratch3/>

Dostupna su proširenja pretvaranja teksta u govor (Text-to-Speech, 3 bloka) i pretvaranje govora u tekst (Speech-to-Text, 3 bloka)



Makeblock - <https://ide.mblock.cc/>

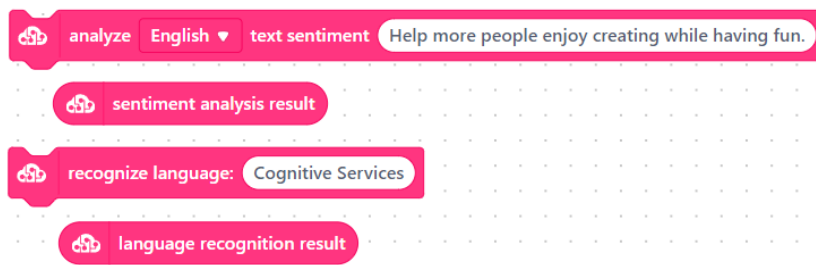
Dostupna su proširenja za pretvaranje teksta u govor (Text-to-Speech, 3 bloka), i kognitivne usluge (Cognitive Services, 3 bloka za prepoznavanje govora)





* DODATAK

Makeblock u *Cognitive Services* proširenju ima i blokove za prepoznavanje jezika i analizu raspoloženja u tekstu



Analiza raspoloženja je proces otkrivanja pozitivnog ili negativnog raspoloženja u tekstu. Tvrtke ga često koriste za otkrivanje osjećaja u društvenim podacima, procjenu ugleda robnih marki i razumijevanje kupaca. Modeli analize sentimenta fokusiraju se na polaritet (*pozitivan, negativan, neutralan*), ali i na osjećaje i emocije (ljuti, sretni, tužni itd.), hitnost (hitni, manje hitni), pa čak i namjere (zainteresirani ili nisu zainteresirani).

Saznajte više o analizi sentimenta na poveznici: <https://monkeylearn.com/sentiment-analysis/>

Da bismo to ilustrirali, uzmimo primjer s Makeblockom.



Korak 1: Otvorite Makeblock sučelje na poveznici: <https://ide.mblock.cc/>

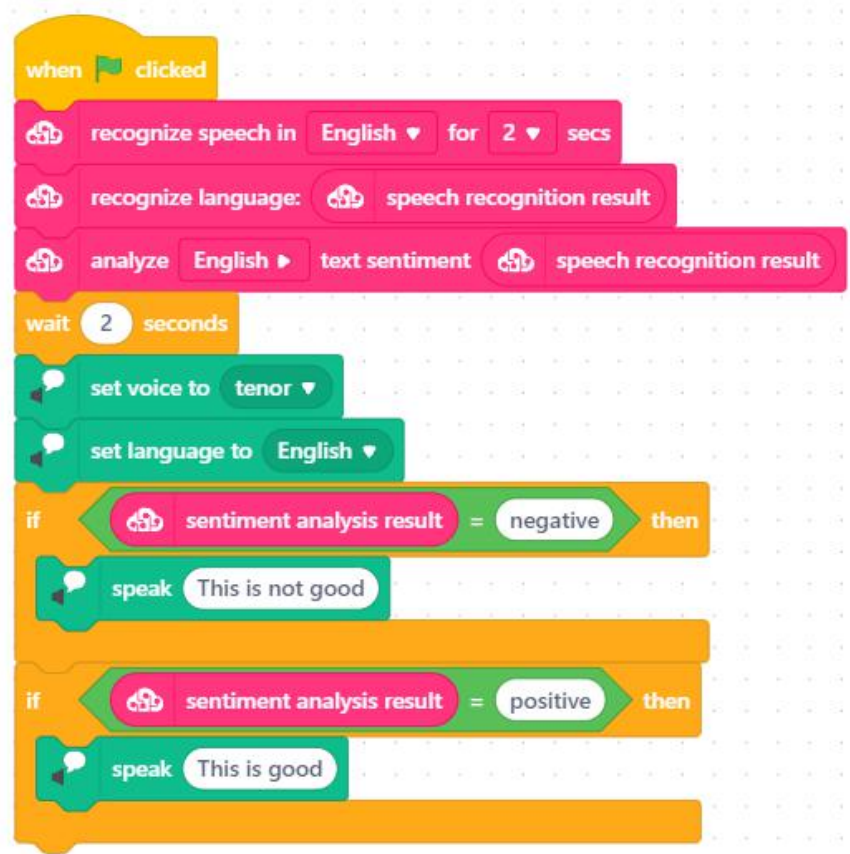
Korak 2: Dodajte proširenja: *Cognitive services* i *Text-to-Speech*

Korak 3: Označite za prikaz sljedeće blokove izvještajnog tipa (*reporter type block*)

- ✓ *speech recognition result*
- ✓ *language recognition result*
- ✓ *sentiment analysis result*

Korak 4: Koristimo blok *speech recognition result* unutar *recognize language* i *analyse text sentiment* blokova

Korak 5: Programsko rješenje koje ćemo koristiti nalazi se na slijedećoj slici:

**f**

Korak 6: Uključite mikrofonski i zvučnik, pokrenite skriptu i recite "Sretan sam" (vidjet ćete skočni prozor koji snima vaš glas 2 sekunde - ako vam je potrebno više vremena, povećajte vrijednost u bloku recognize speech in English for 3 secs (ili više sekundi). Program će zatim izreći "Ovo je dobro" (*This is good*) ili "Ovo nije dobro" (*This is not good*) ovisno o rezultatu analize raspoloženja.





Korak 7: Ponovno pokrenite ovu skriptu, recite nešto drugo i pričekajte rezultat analize sentimenta.

Prepoznavanje govora je sposobnost prevođenja diktata ili izgovorene riječi u tekst. Poznat je i kao Tekst-u-Govor i Prepoznavanje govora (*Speech-to-Text* i *Voice recognition*). To se postiže pomoću određenih koraka, a sustav koji to radi poznat je i kao "Sustav prepoznavanja govora". Sustavi prepoznavanja govora obično se primjenjuju u obliku aplikacija za diktiranje i inteligentnih asistenata u osobnim računalima, pametnim telefonima, web preglednicima i mnogim drugim uređajima.

ZAKLJUČAK

Prepoznavanje govora je sposobnost prevođenja diktata ili izgovorene riječi u tekst. **Generiranje govora** je tehnologija koja stvara glas.



Promisli!



Što sada znam?



Što sam naučio/la?



Što još želim znati?



Programiranje prepoznavanja govora u Scratchu

Posljednjih godina tehnologija prepoznavanja govora postaje sve raširenija. Ovu tehnologiju često koriste i tvrtke i pojedinci zbog brojnih prednosti koje donosi.

"Hej, Siri", "OK Google" i tako dalje – prepoznavanje glasa, često poznato kao tehnologija prepoznavanja govora, nije novi koncept (*SRT, Speech recognition technology*). Odnosi se na vrstu tehnologije koja izgovorene riječi može pretvoriti u strojno čitljive oblike. Sada možete komunicirati sa svojim uređajima i učiniti da djeluju prema vašim glasovnim naredbama, vrlo slično kao u znanstveno-fantastičnim pričama.

Budući da većina tehnologija prepoznavanja govora ima postotak točnosti od preko 95%, nije ni čudo da najnovija statistika glasovnog pretraživanja otkriva da je gotovo 50% svih pretraživanja u 2022. godini zatraženo govorom.

Upoznat ćemo program za prepoznavanje govora i njegovo korištenje na primjeru jednog programa.



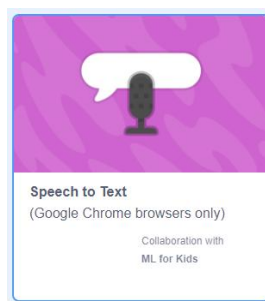
Sufinancirano sredstvima
programa Europske unije
Erasmus+

Ako želimo raditi s prepoznavanjem govora u Scratch i Makeblock aplikacijama imamo proširenja za tu svrhu.

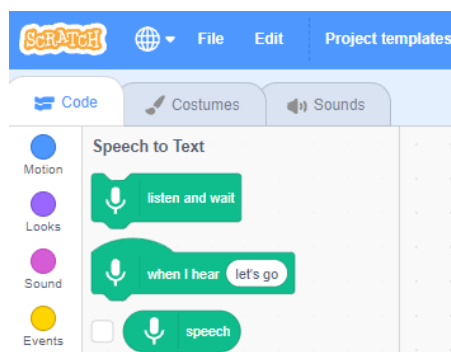
SCRATCH (ML4KIDS):

Korak 1: Otvorite poveznicu u vašem Chrome pregledniku: <https://machinelearningforkids.co.uk/scratch3/>

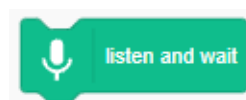
Korak 2: Učitajte proširenje govor u tekst (*Speech to Text*)



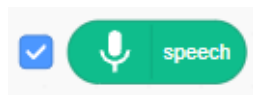
Korak 3: Vidjet ćete novu grupu u blokova u paleti pod nazivom *Speech to Text* sa 3 bloka



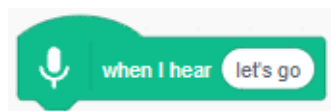
Korak 4: ***listen and wait*** sluša očekujući govor, a zatim obrađuje izgovorene riječi



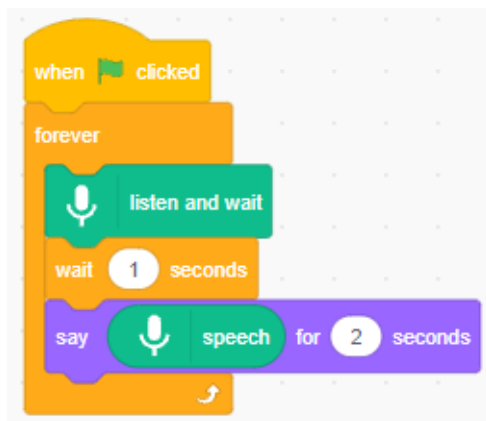
Korak 5: Rezultat prepoznavanja govora prikazan je u *reporter* bloku **speech**. Označite ako želite da se na pozornici (*stage*) prikaže rezultat prepoznavanja govora.



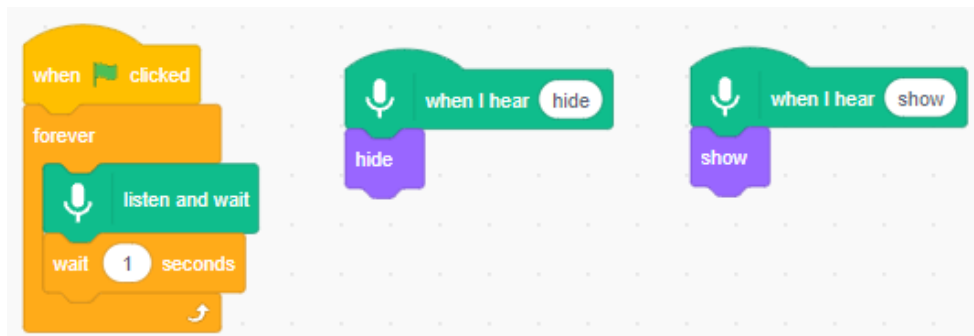
Korak 6: Posljednji blok koji se aktivira događajem (*event trigger*). Ovaj blok čeka riječ u tekstualnom balonu, kao što je "nešto" (*something*) u ovom primjeru, a zatim izvršava slijed naredbi u blokovima koji se nastavljaju na njega.



Korak 7: Napravimo jednostavan slušaj i reci program. Sve što trebate učiniti je postaviti **listen and wait** blok u petlju (*forever*), pričekati 1 sekundu i sa **say** blokom ispisati izgovoreno u trajanju od 2 sekunde.



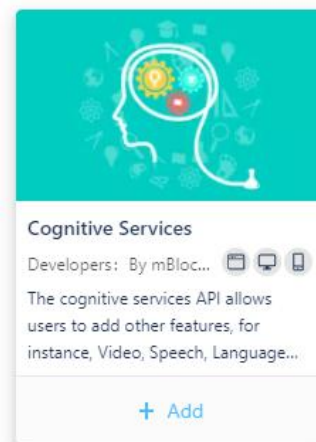
Korak 8: Također možete napraviti program koji očekuje određene riječi da bi pokrenuo događaje. Kao što je ovaj sakrij (*hide*) i pokaži (*show*) program.



MAKEBLOCK

Korak 1a: Otvorite web preglednik i idite na poveznicu:
<https://ide.mblock.cc/>

Korak 2a: Učitavajte proširenje za likove
(sprite extension) pod nazivom Cognitive Services



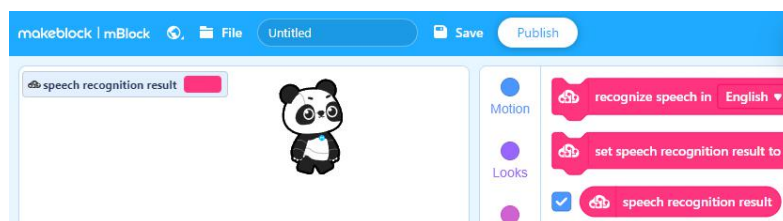
Korak 3a: U ovoj skupini postoji mnogo blokova, ali samo nekoliko njih je vezano uz prepoznavanje govora.



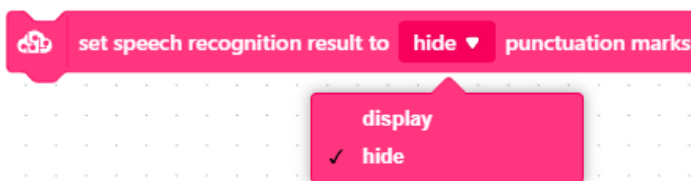
Korak 4a: recognize speech in <language> for <x> seconds blok započinje prepoznavanje govora određeni broj sekundi. Pojavit će **se skočni prozor za prepoznavanje** i vidjet ćete valni oblik zvuka (waveform) dok govorite.



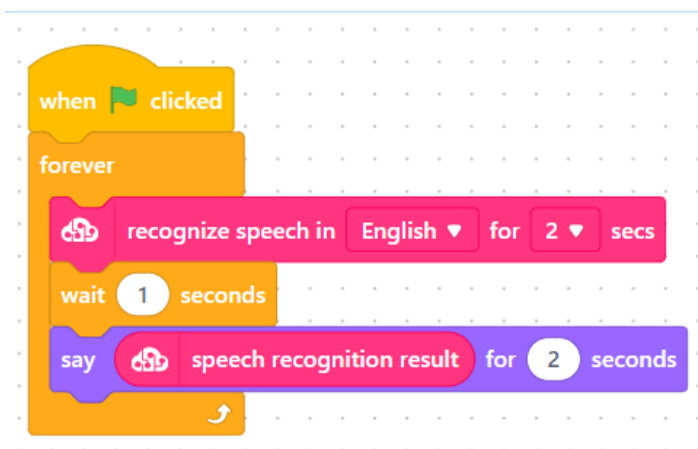
Korak 5a: Rezultati prepoznavanja govora prikazuju se u speech recognition result bloku. Označite želite li da se na pozornici (stage) prikaže rezultat prepoznavanja govora.



Korak 6a: Zadnji blok prikazuje ili sakriva interpunkcijske znakove u rezultatu prepoznavanja govora



Korak 7a: Dakle, napravimo jednostavan slušaj i reci program u Makeblocku. Vidjet ćete da je sličan onom prethodnom (u Scratch verziji).



ZAKLJUČAK

Tehnologija prepoznavanja govora omogućuje glasovnu komunikaciju korisnika i računala.

Promisli!



Što sada znam?



Što sam naučio/la?



Što još želim znati?



Programiranje generiranja govora u Scratchu

Pretvaranje teksta u govor (*Text To Speech - TTS*) vrsta je pomoćne tehnologije koja naglas čita digitalni tekst. Ponekad se zove „*read aloud*“ (čitaj naglas) tehnologija. Klikom na gumb ili dodir prsta TTS može uzeti riječi s računala ili drugih digitalnih uređaja te ih pretvoriti u zvuk. TTS je vrlo koristan za djecu i odrasle koji imaju problema s čitanjem. Uz to može pomoći u pisanju i uređivanju, pa čak i u fokusiranju.

Sustav koncept-govor (*Concept-to-Speech, CTS*) pretvara konceptualni prikaz rečenice koja se izgovara u govor. Dok se neki CTSsustavi sastoje od samostalnih modula za generiranje teksta i pretvaranje teksta u govor (*TTS*), većina postojećih CTS sustava poboljšava vezu između ova dva modula s prosodičnim modulom predviđanja koji koristi jezično znanje iz generatora teksta za predviđanje prozodijskih značajki za TTS generaciju. (U prozodijske značajke ubrajaju se naglasak i/ili akcent, ritam, intonacija te varijacije u brzini i glasnoći govora, kadšto i trajanje fonema)

Generiranje govora može prevesti bilo koji tekst u govor.

Generiranje govora proizvodi izgovorene poruke kao odgovor na signale iz sustava za obradu ili kontrolu podataka. Odabir poruka proizvodi se sastavljanjem govora iz osnovnog skupa zvukova koji mogu biti umjetnog podrijetla ili su izdvojeni obradom zvukova koje je govorom stvorio čovjek.

Uvod u program generiranja govora i njegovo korištenje kroz primjer jednog programa.

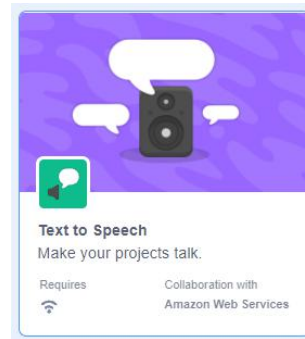


Sufinancirano sredstvima
programa Europske unije
Erasmus+

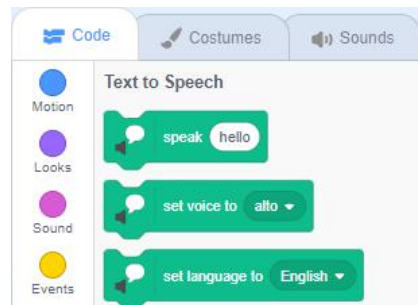


Korak 1: Otvorite Chrome preglednik i idite na poveznicu: <https://machinelearningforkids.co.uk/scratch3/>

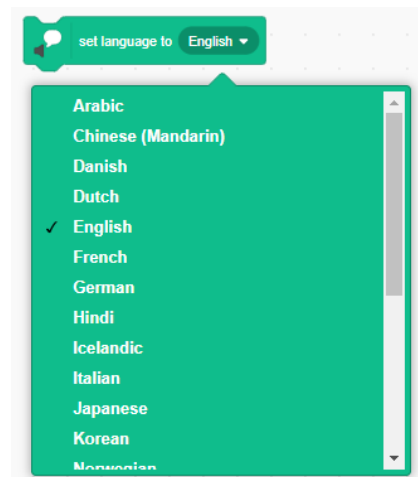
Korak 2: Učitajte proširenje za pretvaranje teksta u govor (Text to Speech)



Korak 3: U paleti blokova vidjeti ćete novu grupu pod nazivom „Text to Speech“ s 3 nova bloka



Korak 4: Idemo od zadnjeg - **set Language to** bloka. Ovaj blok postavlja jezik kojim će se govoriti – možete ga odabrati s padajućeg liste

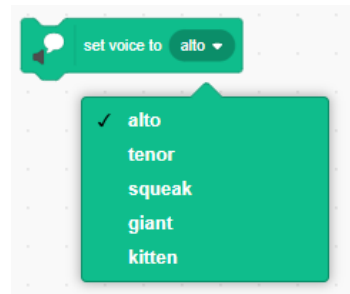




Korak 5: Sljedeći blok – **set voice to** postavlja vrstu glasa.

Možete odabrati glas:

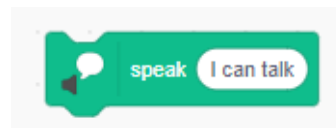
alt, tenor, „skvičanje“, div ili mačka.



Korak 6: Najbitniji blok ovdje je **speak** blok.

Ovaj blok "izgovara" tekst u bijelom balonu, poput ovog "Mogu govoriti" (*I can talk*).

Promijenite tekst u bilo koji drugi i kliknite na blok da biste čuli izgovor. Prije testiranja provjerite jeste li pojačali zvuk na vašem računalu.



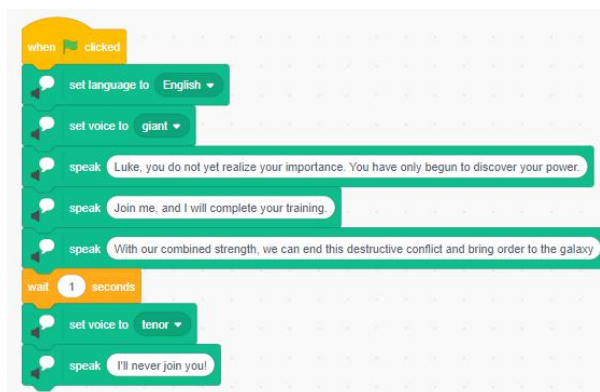
Korak 7: U osnovi i nije neki težak posao postići likovi u Scratchu govore.

Sve što trebate učiniti je postaviti jezik, glas te početi govoriti.



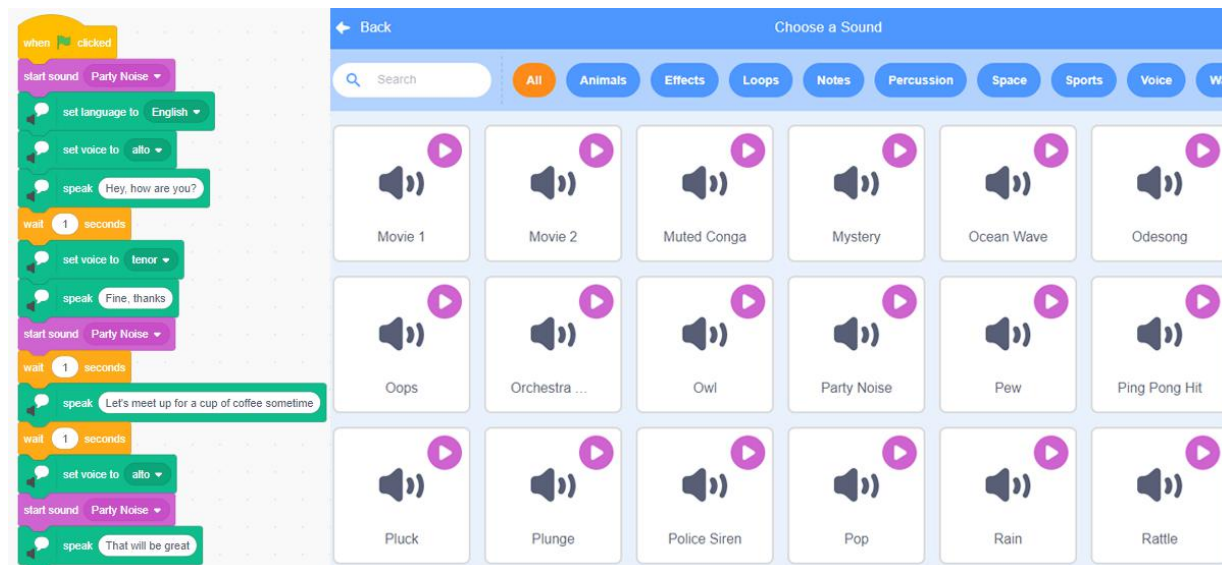
Korak 8: Što možete učiniti s time?

Možete ponovno stvoriti poznate filmske sekvence, poput ove iz Ratova zvijezda:



Korak 9: Ili možete napraviti svoj vlastiti film, ispričati priču ...

Kombinirajte TTS s drugim zvukovima iz galerije kako biste scenu učinili realističnijom.



Danas imamo mnogo uređaja za generiranje govora. Uređaji za generiranje govora omogućuju ljudima da "izgovaraju" riječi i rečenice elektroničkim putem. Uređaji za generiranje govora danas su mali elektronički uređaji koji reproduciraju riječi ili fraze kada korisnik pritisne nekakav prekidač ili dodirne tipku. Neki uređaji 'izgovaraju' riječi dok se te riječi upisuju na tipkovnici. Uređaji za generiranje govora omogućuju ljudima koji ne mogu koristiti govorni jezik da 'govore' elektroničkim putem. Takvi uređaji se koriste i za pomoć u komunikaciji kod autistične djece od 1990-ih.

ZAKLJUČAK

Pretvaranje teksta u govor (*Text To Speech - TTS*)
vrsta je tehnologije koja naglas čita digitalni tekst.

Promisli!



Što sada znam?



Što sam naučio/la?



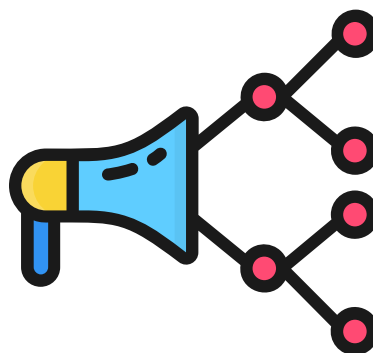
Što još želim znati?



Projekt glasovnog upravljanja objektom

Može li se upravljati nekim objektom pomoću glasa.
Je li moguće na neki način imati koristi od glasovnog upravljanja objektima?

U nastavku ćemo upoznati algoritme glasovnog upravljanja objektom i njega uporaba u praktičnom radu.



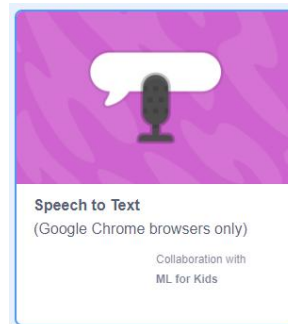


Kroz projekt, učitelj upoznaje učenike sa Scratch naredbama i vještinama programiranja, kao i na to kako se koristi prepoznavanje govora.

Korak 1: otvorite internetski preglednik Chrome i idite na poveznicu: <https://machinelearningforkids.co.uk/scratch3/>

Korak 2: Učitajte proširenje Govor u tekst
(*Speech to text*, dostupno samo za Chrome preglednike)

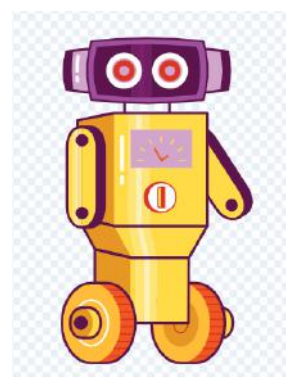
Korak 3: Uklonite lik mačke klikom na ikonu kante za smeće



Korak 4: Preuzmite sliku (labirint.png) sa poveznice: <https://drive.google.com/file/d/11YBBhQclhVfHYMWELkhgAYKSswfv33pT5/view?usp=sharing> i učitajte ju kao novi lik u Scratch i nazovite ga **labirint**.



Korak 5: Iz galerije likova odaberite **Retro Robot** i koristite drugi kostim (*Retro Robot b*)



Korak 6 : iz galerije likova odaberite **Home Button**





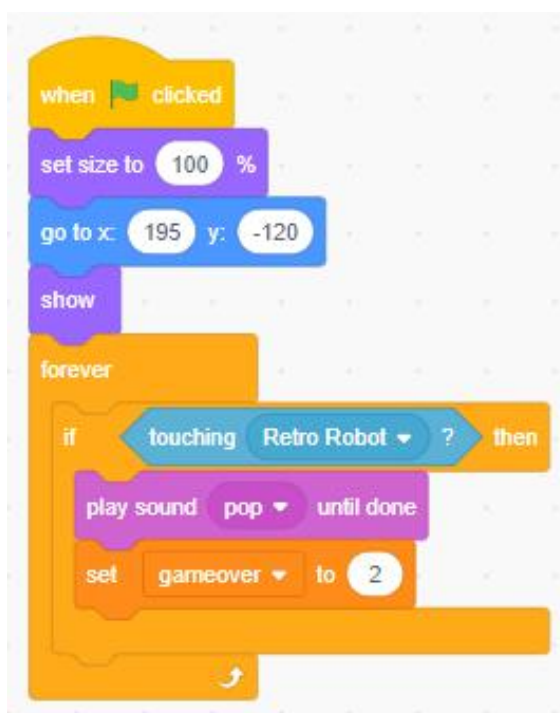
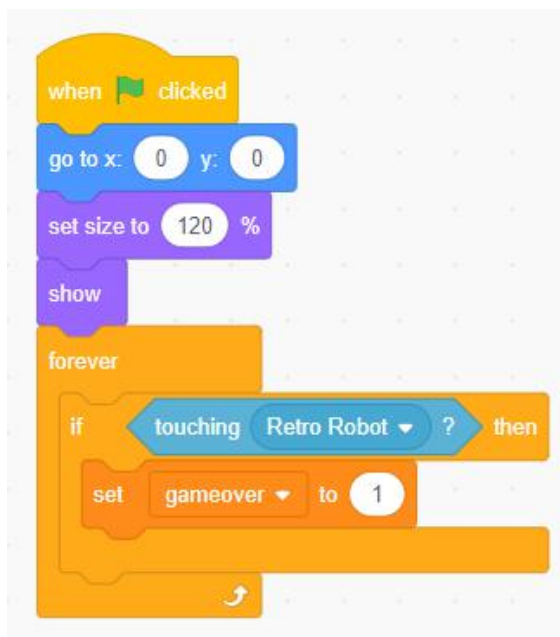
Korak 7: Napravite 3 varijable (za sve likove):

- **gameover** (pokazuje kako je igra završila)
- **xm** (x položaj robota)
- **ym** (y položaj robota)

Korak 8: kreirajte listu i promijenite njeno ime u **endtalk** (prikazuje poruku za kraj igre, ovisno o tome kako je igra završila)

Korak 9: Prebacite se na lik **labirint** i počnite s programiranjem skripte za njega. Prvih nekoliko blokova postavlja položaj, veličinu i vidljivost labirinta. U beskonačnoj petlji koja slijedi nalazi se provjera dodira labirinta s likom robota. U slučaju dodira (robot se sudario s zidom labirinta), varijabla **gameover** postavlja se na 1.

Korak 10: prijedite na lik **Home Button**. Skripta je gotovo ista, osim što se u slučaju dodira s likom robota reproducira **pop** zvuk i vrijednost **gameover** varijable se postavlja na 2.





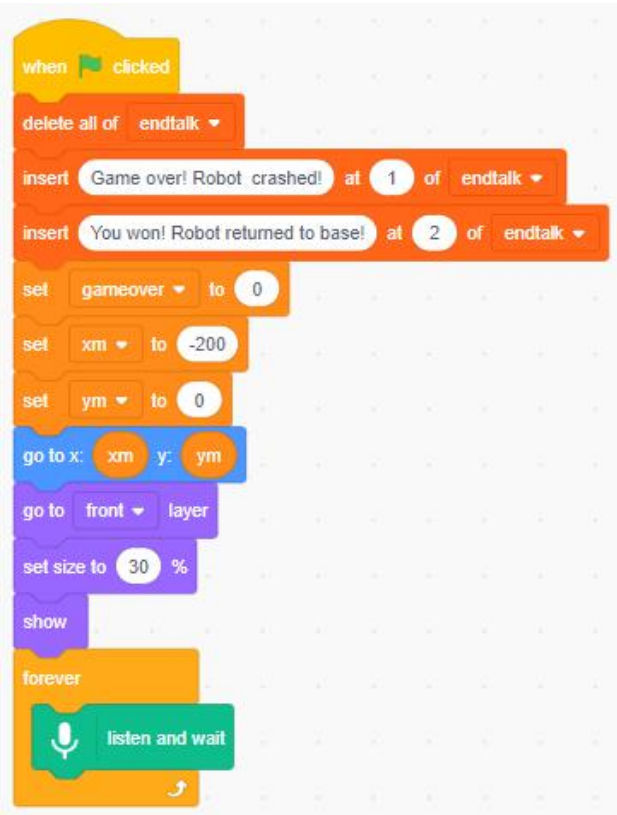
Korak 11: Glavni skripta pripada liku **Retro robot** i postoji 5 događaja koji počinju istovremeno.

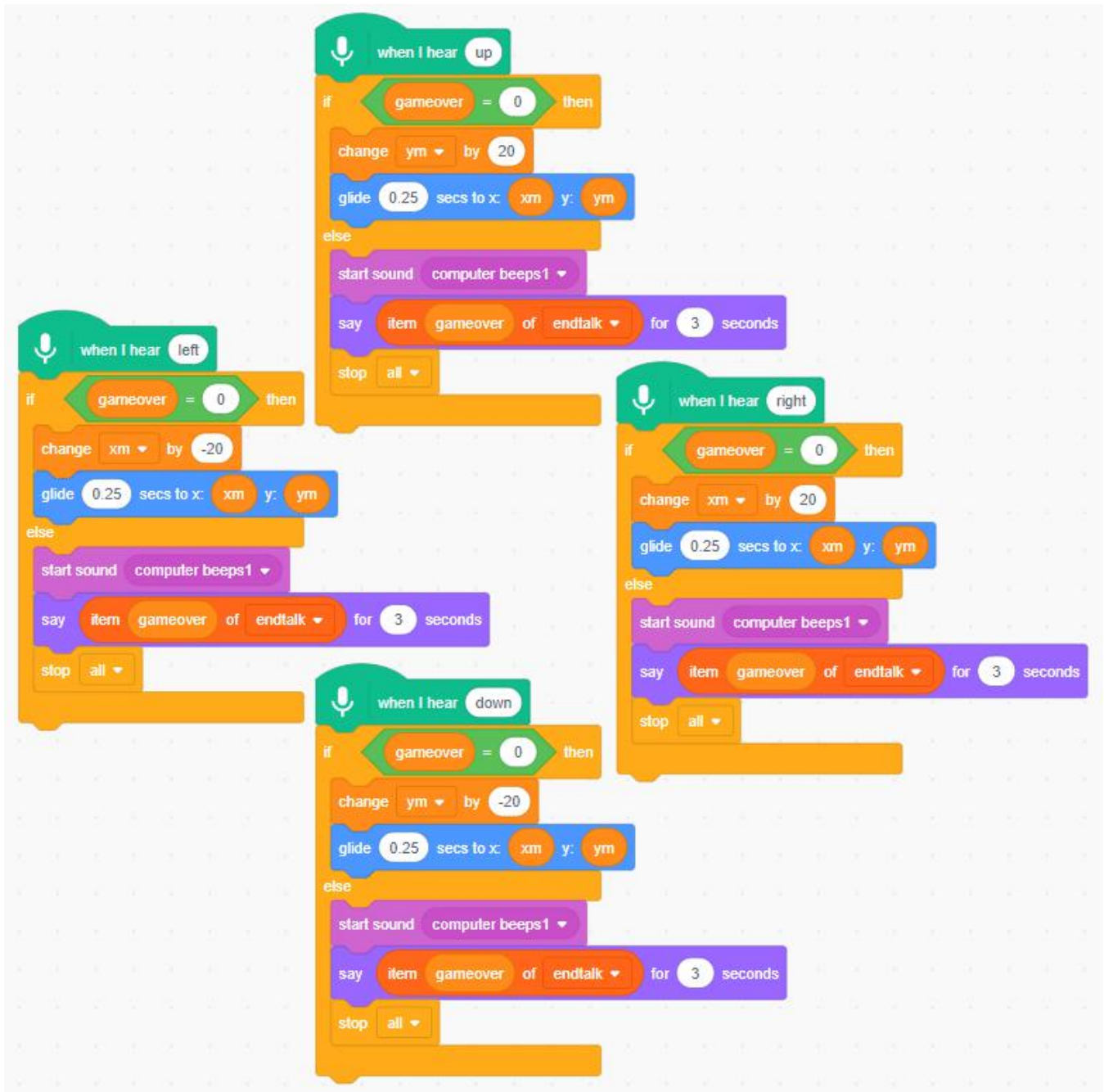
11.1 kada se klikne na zastavica (start programa) postavljaju se varijable i unosi u listu (prvo se brišu svi prethodni unosi) te položaj i vidljivost lika. Na kraju se nalazi blok za slušanje (iz *Speech to Text* proširenja) u beskonačnoj petlji.

11.2 Sljedeća 4 događaja pokreću se prepoznavanjem govora i svaka prepoznata glasovna naredba obrađuje se nizom blokovima.

Ako se ne otkrije dodir lika robota s likovima **labirint** ili **Home Button** (vrijednost varijable *gameover* ostaje 0) robot se kreće u određenom smjeru (promjenom vrijednosti koordinata *xm* ili *ym* i pomicanjem robota u izračunati položaj).

Ako se otkrije sudar (vrijednost varijable **gameover** je 1 ili 2) prikazat će se pripadajuća poruka (iz liste **endtalk**), a igra će se zaustaviti.

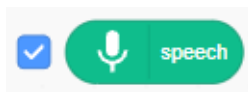






Korak 12: Pokušajte koristiti glasovne naredbe na materinjem jeziku i pogledajte kako to funkcionira.

Razgovarajte o tome kako (ne)ispravan izgovor utječe na prepoznavanje govora. Uključite prikaz rezultata prepoznavanja govora označavanjem bloka **speech**.



Roboti za praćenje objekata, ako se njima može pametno upravljati glasom, mogu biti od ogromne pomoći ljudima s fizički poteškoćama. Sustav prepoznavanja govora koristi se za prepoznavanje skupa unaprijed definiranih naredbi kao što su naprijed, natrag, lijevo, desno i rotacija pod određenim kutom. Robot se kreće prema glasovnim naredbama, a istovremeno prati željeni objekt. Obrada signala glasovnih naredbi provodi se u stvarnom vremenu, koristeći poslužitelj u oblaku koji ga u kratkom vremenu pretvara u tekstualni obrazac. Tekst naredbenog signala zatim se prenosi robotu putem Bluetooth mreže kako bi se kontroliralo njegovo kretanje. Prototip pametnog robota sastoji se od tri podsustava: sustava prepoznavanja govora, sustava praćenja objekata i sustava za upravljanje kretanjem. Točnost i učinkovitost sustava prepoznavanja govora ispituje se kroz praktične eksperimente. Ispituje se učinak čimbenika kao što su buka i udaljenost itd., a rezultati su ohrabrujući. Prototip robota može prepoznati glasovne naredbe unutar dometa bluetooth uređaja. Raspravlja se i o mogućim proširenjima koja bi mogla dovesti do širenja mogućnosti primjene.

ZAKLJUČAK

Glasom možemo upravljati objektima u računalnom programu.

Promisli!



Što sada znam?



Što sam naučio/la?



Što još želim znati?



Uvod u hardware - mikrokontroler, kamera i upravljač motora

Dio ovog kurikuluma povezan je s pravim, fizičkim uređajem koji ima mogućnost primjene umjetne inteligencije. U tu svrhu Hrvatski robotički savez razvio je malog mobilnog robota za implementaciju programskih znanja iz prethodnih lekcija. Prvo, razmotrimo od čega bi se naš robot trebao sastojati. Jasno je da će se mehanički dijelovi koristiti u kućištu, a kotači i motori kao pogon. Što je s elektroničkim dijelovima? Slično kao i većina živih bića, naš robot će trebati mozak (mikrokontroler), oči (AI kamera) i živce / impulse (upravljač motora). Pogledajmo te dijelove.

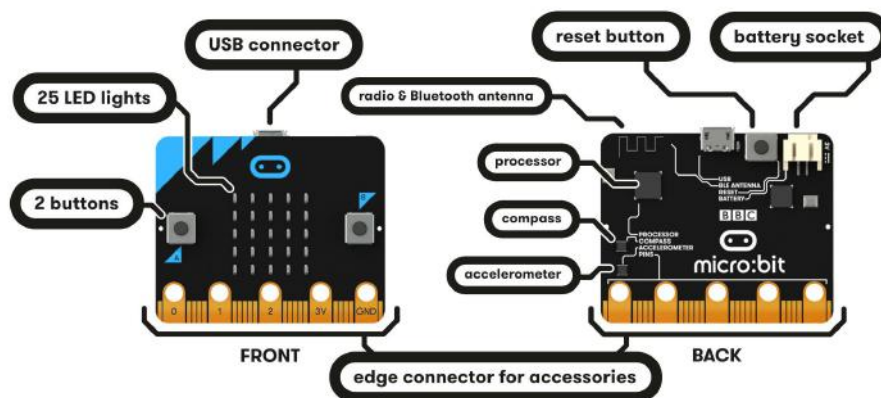
Postoje dvije mogućnosti za izgradnju robota:
Robotski komplet Maqueen Plus + HuskyLens AI kamera (jednostavna i pogodna za početnike)
micro: Maqueen Plus je napredni STEM edukacijski robot za Micro:bit. Ovaj micro:bit robot ima velik broj mogućnosti i optimizirano upravljanje napajanjem i napajanje baterijama velikog kapaciteta i potpuno je kompatibilan s HuskyLens AI kamerom, što ga čini pristupačnim nastavnim alatom. Micro:Maqueen Plus ima veliko i solidno kućište, dobre osobine i više priključaka za proširenja. Nije pogodan samo za nastavu u učionici, već se može koristiti i za vannastavne aktivnosti i razna robotička natjecanja.

Arduino UNO + Upravljač motora + HuskyLens AI kamera (složenija varijanta - samo za iskusne korisnike)



Mozak 1 - micro:bit

"Mozak" u elektroničkom svijetu naziva se procesor ili, u ovom specifičnom slučaju - mikrokontroler. Micro:bit je jednostavan za korištenje, moćan i isplativ mikrokontroler džepne veličine dizajniran za podučavanje djece i početnika kako programirati, omogućujući im da lako unesu svoje ideje u "uradi sam" digitalne igre, interaktivne projekte i robotiku.

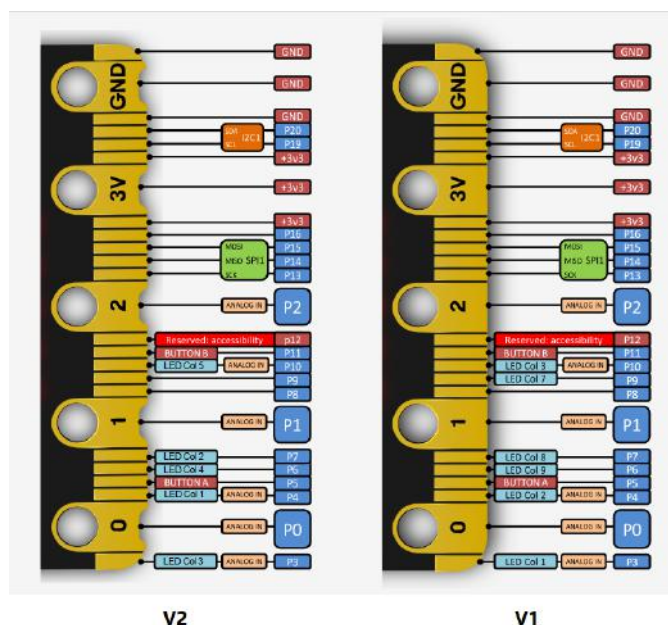


Zahvaljujući vanjskim ulazno/izlaznim priključcima i hardverskoj podršci, Micro:bit je pogodan za različita učenja i razvoj povezana s robotima.

Micro:bit izgled i tehnološke specifikacije

Na tržištu su dostupne dvije verzije micro:bita - pogledajte ovaj članak kako biste saznali koju imate:

<https://kitronik.co.uk/blogs/resources/explore-micro-bit-v1-microbit-v2-differences>





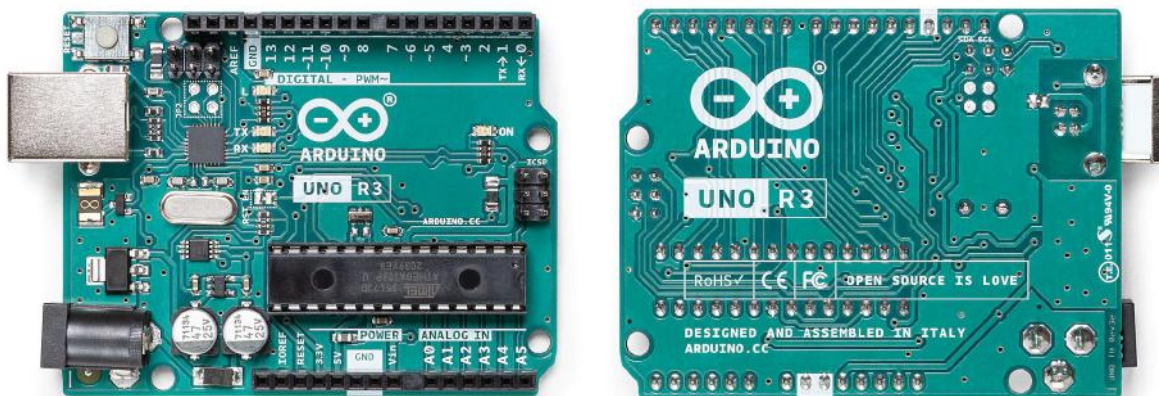
- Mala ploča slične veličine kreditnoj kartici (4 cm x 5 cm)
- Ugrađeni moduli, kao što su akcelerometar, kompas i Bluetooth® Smart modul
- Mikrokontroler veličine džepa
- LED matrica 5x5 (također podržava otkrivanje svjetlosti)
- Senzori svjetla i temperature i drugi uobičajeni senzorski uređaji

Opremljen ARM-ovim M0 procesorom, micro:bit može izvršiti većinu temeljnih funkcija robota.

Uvod u micro:bit https://www.youtube.com/watch?v=POkeI_2NXMo

Mozak 2 - Arduino UNO

Arduino UNO najpopularniji je mikrokontroler na svijetu s ogromnom internetskom zajednicom i mnoštvom projekata dostupnih na internetu. Na tržištu postoji mnogo UNO klonova. Klon znači da je temeljna arhitektura, u električkom smislu, slična Arduino UNO-u, ali napravljene su neke modifikacije kako bi se pružile dodatne značajke ploči. Ove modifikacije su dizajnirane i razvijene posebno kako bi se naučilo programiranje i arhitektura mikrokontrolera. Pogledajmo Arduino UNO.



Svaki od 14 digitalnih pinova na Arduino UNO može se koristiti kao ulaz ili izlaz i radi s naponima do 5 volti. Svaka pin može podnijeti jačinu struje do 20 mA kao preporučeno radno stanje i ima ugrađeni *pull-up* otpornik (odspojen prema zadanim postavkama) od 20-50k ohm. Najviša vrijednost strujnog opterećenja je 40mA i ne smije prekoračiti niti na jednom ulazno-izlaznom pinu kako bi se izbjeglo trajno oštećenje mikrokontrolera.

Osim toga, neki pinovi imaju posebne funkcije:

- Serijski: 0 (RX) i 1 (TX). Koristi se za primanje (RX) i prijenos (TX) TTL serijskih podataka. Ti su pinovi spojeni na odgovarajuće izvode ATmega8U2 USB-to-TTL serijskog čipa.
- Vanjski prekidi (*external interrupts*): pinovi 2 i 3. Ti se pinovi mogu konfigurirati tako da pokrenu prekid pri niskoj vrijednosti, pri rastu ili padu napona ili pri promjeni vrijednosti napona.
- PWM: 3, 5, 6, 9, 10 i 11. Ovi pinovi daju 8-bitni PWM (pulsno-širinski moduliran) izlaz.



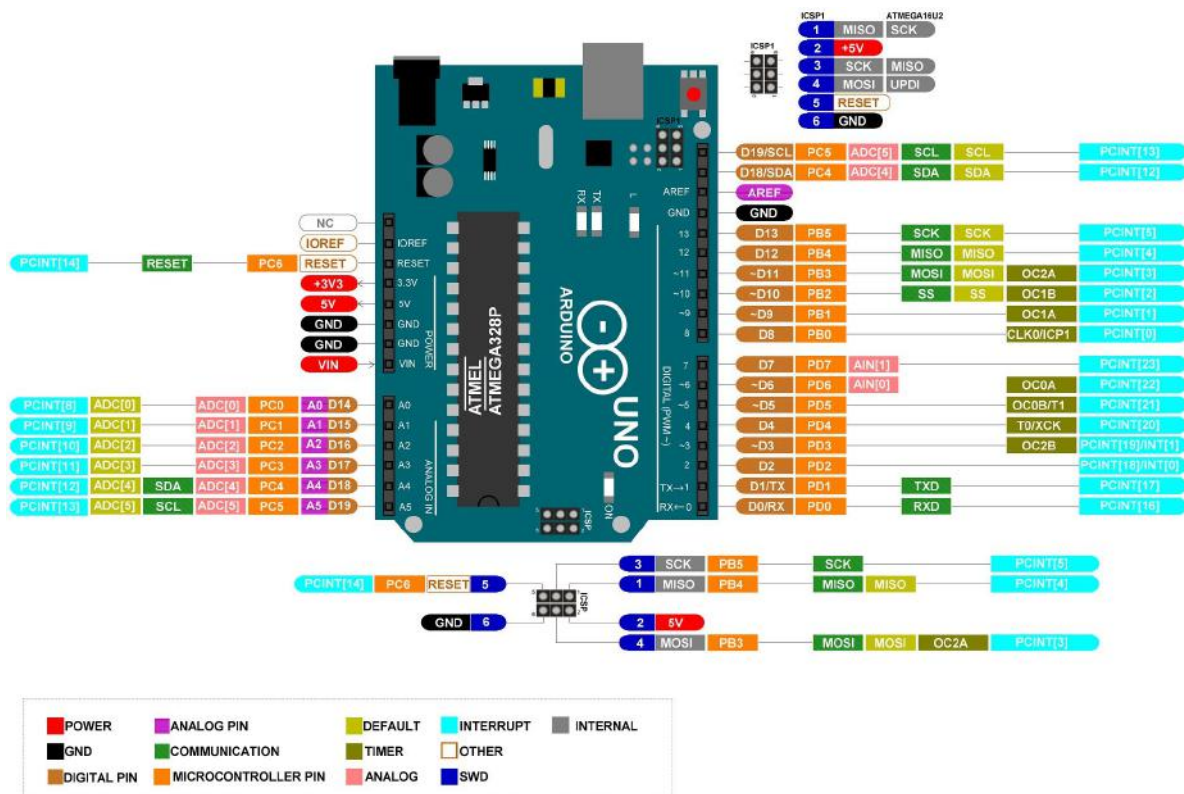
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ovi pinovi podržavaju SPI (serijsku podatkovnu) komunikaciju pomoću SPI biblioteke.
- LED: 13. Postoji ugrađena LED dioda spojena na digitalni pin 13. Kada je pin postavljen u stanje visoke logičke vrijednosti, LED je uključen, kada je u niskom logičkom stanju - LED je isključen.
- TWI/I2C: A4 ili SDA (Data signal) pin i A5 ili SCL (Clock signal) pin. TWI/I2C komunikacija podržana je pomoću biblioteke wire.

UNO ima 6 analognih ulaza, označenih s oznakama od A0 do A5, od kojih svaki pruža 10 bitnu razlučivosti (tj. 1024 različite vrijednosti). Prema zadanim postavkama mjere se od naponi do 5 volti, iako je moguće promijeniti gornji kraj njihovog raspona pomoću AREF pina. Postoji još nekoliko posebnih pinova na ploči:

- AREF. Referentni napon za analogne ulaze.
- RESET. Dovođenje ovog pina u nisko logičko stanje inicira resetiranje mikrokontrolera. Obično se koristi za dodavanje gumba za resetiranje na štitovima u slučaju da blokiraju pristup gumbu za RESET na Arduino UNO ploči.

Arduino UNO izgled i tehničke specifikacije

Arduino UNO ploča s mapom pinova:





MIKROKONTROLER	ATmega328P
RADNI NAPON	5V
ULAZNI NAPON (PREPORUČENO)	7-12V
ULAZNI NAPON (GRANIČNO)	6-20V
DIGITALNI U/I PINOVI	14 (od kojih 6 daje PWM izlaz)
PWM DIGITALNI U/I PINOVI	6
ANALOGNI ULAZNI PINOVI	6
ISTOSMJERNA STRUJA PO ULAZNO-IZLAZNOM PINU	20 mA
ISTOSMJERNA STRUJA ZA PIN OD 3,3 V	50 mA
FLASH MEMORIJA	32 KB (ATmega328P) od čega 0,5 KB koristi bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
FREKVENCIJA TAKTA	16 MHz
UGRAĐENI LED (LED_BUILTIN)	13
DULJINA	68,6 mm
ŠIRINA	53,4 mm
TEŽINA	25 g

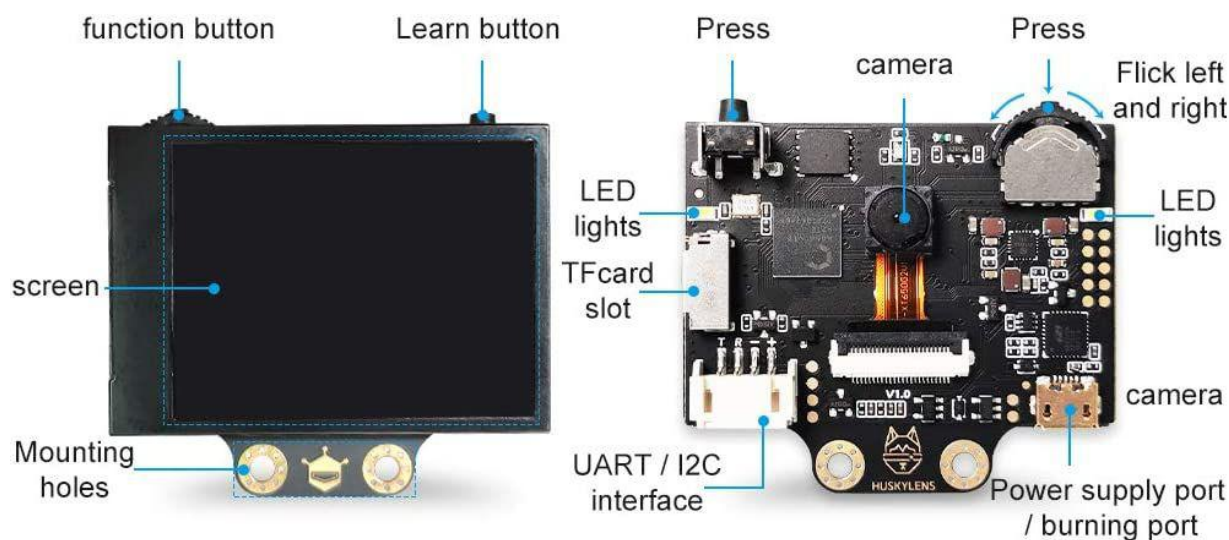
Glavno napajanje za Arduino UNO je putem USB veze, ali se može napajati iz baterija spojenih putem Pinova označenih sa VIN (+) i GND (-).

Arduino UNO uvod: <https://www.youtube.com/watch?v=bniUECtJkeU>

Oči - HuskyLens Kendryte K210 (koristi se u svim varijantama robota)

Naš mobilni robot moći će osjetiti okolinu pomoću ove kamere. HuskyLens kamera je jednostavna za korištenje, sadrži AI senzor računalnog vida sa 7 ugrađenih funkcija: prepoznavanje lica, praćenje objekata, prepoznavanje objekata, prepoznavanje linije, praćenje linija, prepoznavanje boja, prepoznavanje oznaka i klasifikacija objekata.

Može se lako povezati s bilo kojim uređajem kompatibilnim s Arduinoom / Arduinoom i micro:bitom. Sada možete raditi vrlo kreativne projekte čak i bez nekog posebnog znanja složenih algoritama strojnog učenja. Pogledajmo kameru.

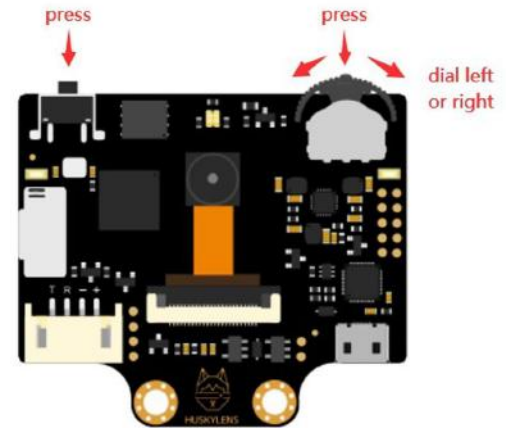




Tipke (gumbi)

Na HuskyLensu postoje dva gumba, funkcijski gumb i gumb za učenje. Osnovne operacije ovih dvaju gumba prikazane su na sljedeći način:

- Potisnite funkcijski gumb (*Function button*) ulijevo ili udesno za prebacivanje između različitih funkcija.
- Kratki pritisak na gumb za učenje (*Learn button*) pokreće "učenje" objekta koji je na lici kamere; dugi pritisak na gumb za učenje omogućuje kontinuirano učenje objekta dok ga pratite kamerom iz različitih kutova i udaljenosti; ako je HuskyLens već naučio objekt, kratkim pritiskom na gumb za učenje možete (nakon potvrde da to zaista želite) "zaboraviti" taj objekt.
- Dugi pritisak nadolje za funkcijski gumb otvara izbornik druge razine (gdje su postavke parametara) za trenutni režim rada. Birajte lijevo, desno ili kratko pritisnite funkcijski gumb da biste postavili odgovarajući parametar.



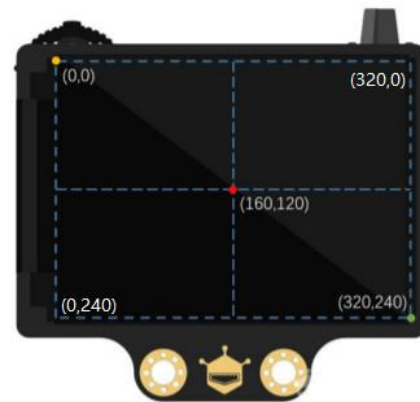
Koordinatni sustav

Kada HuskyLens detektira objekt, on će na zaslonu kamere biti označen okvirom. Koordinate položaja tog okvira su x i y , a dodjeljuju se prema koordinatnom sustavu na slici dolje.

Nakon dobivanja koordinata iz UART / I2C priključka, znamo točan položaj objekta u formatu: x,y

Funkcije:

- 1) Prepoznavanje lica (*Face recognition*)
- 2) Praćenje objekata (*Object tracking*)
- 3) Prepoznavanje objekata (*Object recognition*)
- 4) Praćenje linije (*Line tracking*)
- 5) Prepoznavanje boje (*Color recognition*)
- 6) Prepoznavanje tagova (*Tag recognition*)
- 7) Klasifikacija objekata (*Object classification*)



Upute za boje

Za sve funkcije, boje okvira i simbol "+" u centru zaslona su isti, što vam pomaže da znate trenutni status HuskyLensa.

Boja	Status
Od narančaste do žute, zatim od žute do narančaste	Objekt nije naučen, ali je sve spremno za učenje
Žut	Učenje novog objekta
Plav	Objekt je naučen i prepoznat



RGB LED indikator koristi se za označavanje statusa funkcije prepoznavanja lica. Njegova boja definirana je na sljedeći način.

Boja	Status
Plav	Lice je detektirano, ali nije naučeno
Žut	Učenje novog lica
Zelen	Lice je naučeno i prepoznato

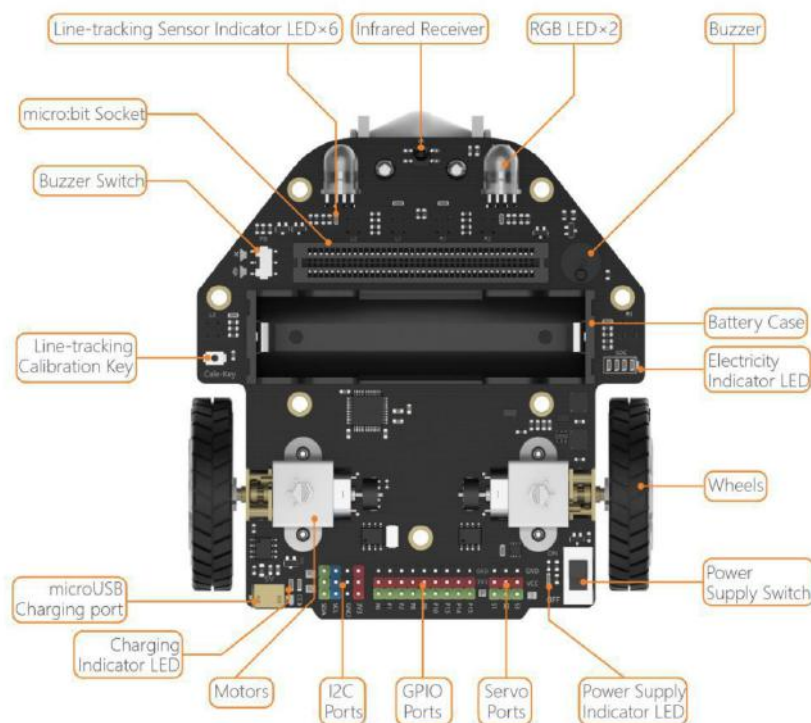
Upute za rad s HuskyLens

kamerom: https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336

Živci/impulsi/mišići 1 - Robotska ploča Maqueen Plus

Maqueen Plus je programabilni edukacijski robot dizajniran za početnike. Može se programirati s programskim platformama Mind+ i MakeCode. Optimizirano je upravljanje napajanjem i dodano napajanje većeg kapaciteta. Idealan je za uporabu s HuskyLens AI kamerom i dolazi s većim i stabilnijim kćištem. Također ima više ugrađenih funkcija i više priključaka za proširenje. Pogodan je za nastavu u učionici te za izvannastavne aktivnosti i natjecanja robota.

Evo glavnih dijelova na ploči Maqueen Plus:





Maqueen plus tehničke specifikacije

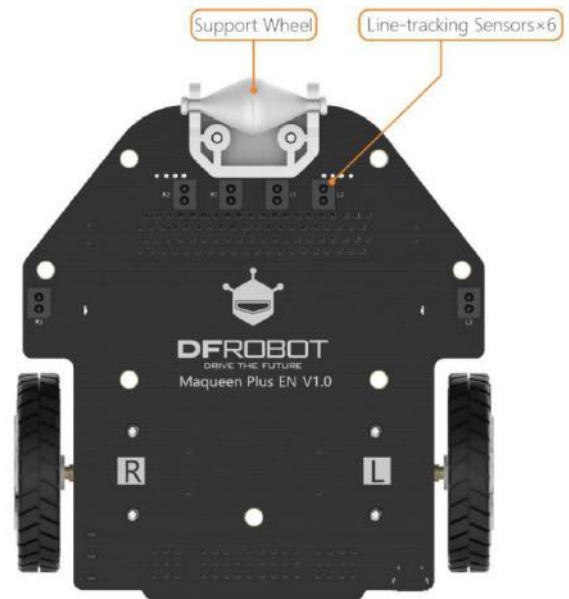
- Napajanje: litijeva baterija od 3,7 V - tip 18650
- Napon punjenja: 5V
- Struja punjenja: 900mA, Vrijeme punjenja: 4h
- Indikator baterije: 4 LED diode
- Pogonski motori: 2 x N20 motor (260 okr/min)
- Zujalica * 1
- RGB-LED * 2
- GPIO priključci za proširenja: P0 P1 P2 P8 P12 P13 P14 P15 P16
- I2C portovi za proširenja * 3
- Servo portovi za proširenja *3
- Senzori za praćenje linije *6
- Izlazni podaci senzora za praćenje linije: analogni + digitalni
- Kalibracija linijskih senzora: podržana
- Infracrveni prijemni senzor *1
- Ultrazvučni senzor: URM10
- Gornja metalna ploča * 1
- Rupe s M3 navojima za spajanje *12
- Veličina podlog za rad s robotom: 50cm x 50cm
- Dimenzija: 107 x 100mm (4,21" x 3,94")

Maqueen Plus robotu pristupamo putem Mind+ programskog sučelja. Mind+ je platforma za blokovsko programiranje temeljena na Scratch 3.0 aplikaciji razvijena u tvrtki DFRobot, koja osim blokova podržava Python, Arduino i druge programske platforme. Trenutno se Mind+ primjenjuje za sve vrste kompatibilnih senzora i modula.

Saznajte više o Maqueen Plus:

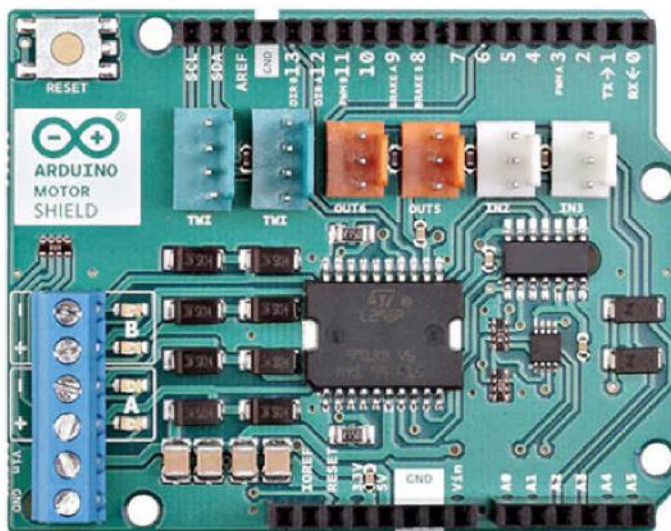
https://github.com/DFRobot/Maqueen_Plus_Basic_Tutorial/blob/master/MBT0021-EN-Maqueen%20Plus%20Basic%20Tutorial.pdf

Kako se ovaj scenarij dovršavao, pojavila se i druga verzija (verzija V2) ovog robota na tržištu.

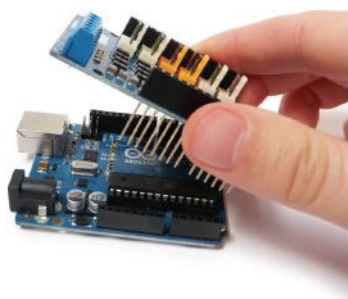


Živci/impulsi/mišići 2 - Arduino Motor Shield Rev 3

Postoji nekoliko načina na koje možemo kontrolirati istosmjerni motor; a najlakši način je direktnim priključenjem na odgovarajući napon. Prva rješenja upravljanja motora radili su na taj način: varirajte veličinu napona i motor će se početi okretati brže ili sporije, mijenjate polaritet priključaka i promijeniti ćete smjer. Ali ako želimo postići više od samog okretanja motora punom brzinom u jednom smjeru, potreban nam je upravljački sklop za motore. Za tu svrhu korisimo dvostruki H-most (L298P čip), koji se nalazi na upravljaču motora (*Motor Shield Rev3*).



Štit za motore se može slagati na Arduino UNO, a to znači da ne morate spajati ožičenje. Samo priključite muške pinove štita na odgovarajuće ženske pinove na mikrokontroleru.



Štit za motore ima 2 kanala, što omogućuje upravljanje s dva istosmjerna motora ili s jednim koračnim motorom (*stepper motor*).

Također ima 6 priključaka za spajanje Tinkerkit ulaza, izlaza i komunikacijskih vodova. Korištenje ovih pinova donekle je ograničeno i stoga nije obuhvaćeno ovim scenarijem.

Uz vanjsko napajanje, motorni štit može osigurati do 12V i 2A po jednom kanalu, odnosno motoru (ili maksimalnih 4A ako se koristi jedan kanal).

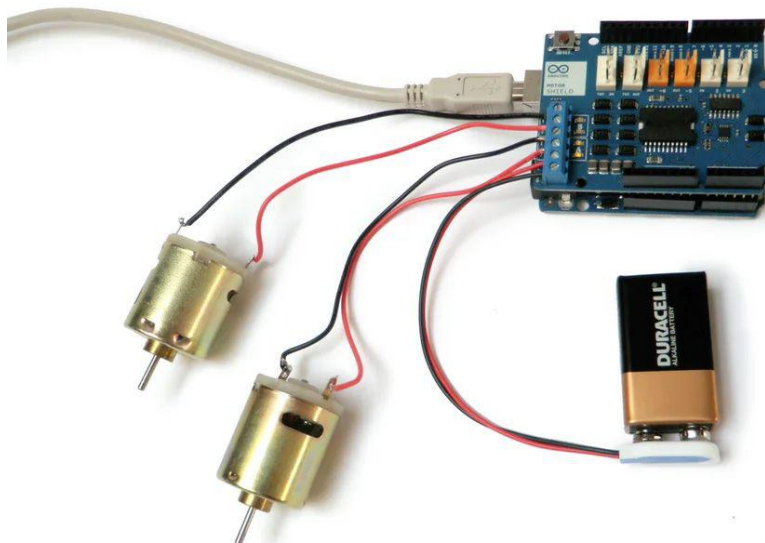
Na Arduino postoje pinovi koje štit za motore fizički koristi i nije ih moguće koristiti u druge svrhe. Dodjeljivanjem vrijednosti ovim pinovima možete birati kojim motorom upravljate, odrediti smjer vrtnje tog motora, postaviti brzinu vrtnje pomoću pulsno-širinskog upravljanja (PWM), zaustaviti i pokrenuti motor te pratiti trenutnu vrijednost struje svakog kanala.

Funkcije pinova su sljedeće:



Funkcija	Kanal A	Kanal B
Smjer	Digitalni pin 12	Digitalni pin 13
Brzina (PWM)	Digitalni pin 3	Digitalni pin 11
Kočnica	Digitalni pin 9	Digitalni pin 8
Senzor struje	Analogni pin 0	Analogni pin 1

I na kraju, ovako bi to trebalo izgledati kad smo sve spojili:



Saznajte više o upravljaču motora na ovoj poveznici (*Instructables*):
<https://www.instructables.com/Arduino-Motor-Shield-Tutorial/>

ZAKLJUČAK

Slično kao i većina živih bića, naš robot ARTIEBot ima mozak (mikrokontroler), oči (AI kamera) i živce / impulse (upravljač motora).

Promisli!



Što sada znam?



Što sam naučio/la?

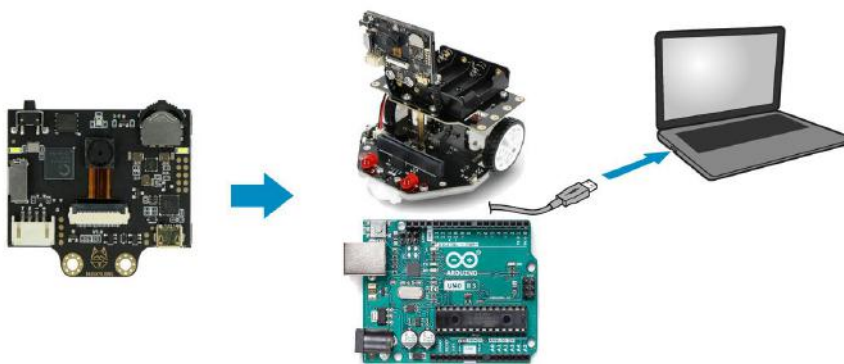


Što još želim znati?



Povezivanje mikrokontrolera s kamerom i računalom, početak rada

Prije nego što počnemo s programiranjem, moramo spojiti kameru na mikrokontroler (micro:bit ili Arduino UNO) i mikrokontroler na računalu



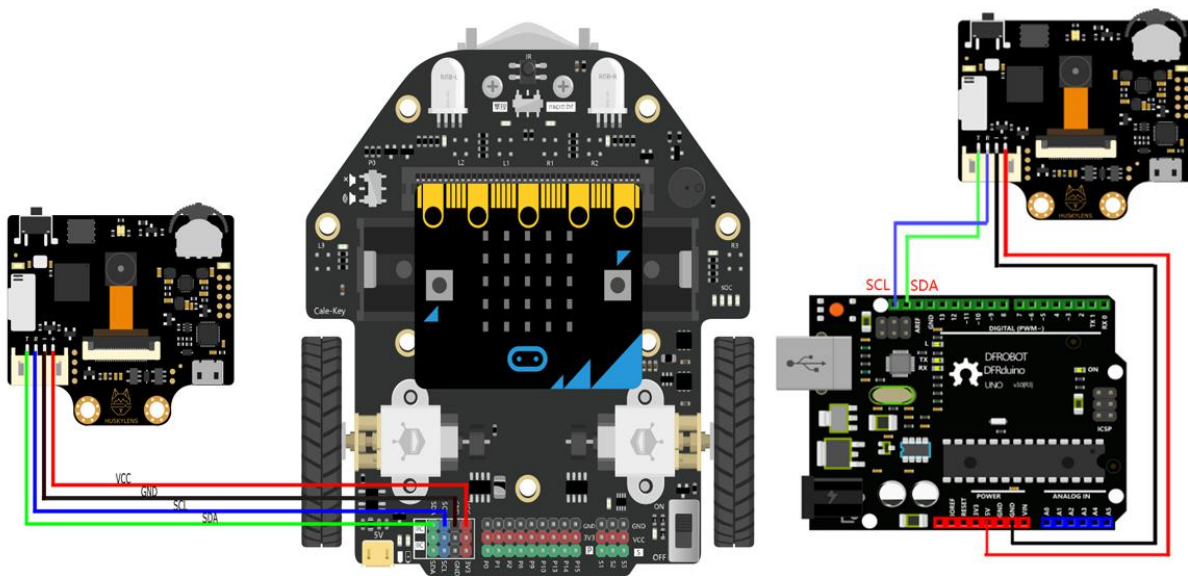
Opcija 1:
Kamera > I²C priključak > **micro:bit/Maqueen plus** > USB
kabel > prijenosno računalo ili računalo

Opcija 2:
Kamera > I²C priključak > **Arduino UNO** > USB kabel >
prijenosno računalo ili računalo

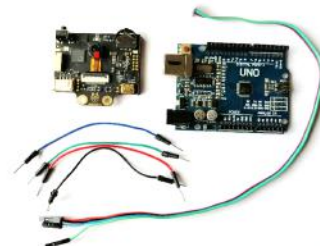
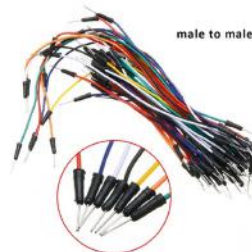
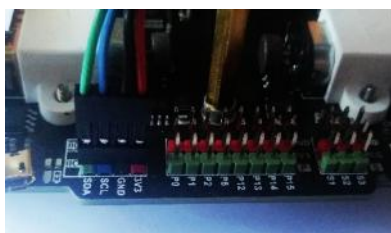


Sufinancirano sredstvima
programa Europske unije
Erasmus+

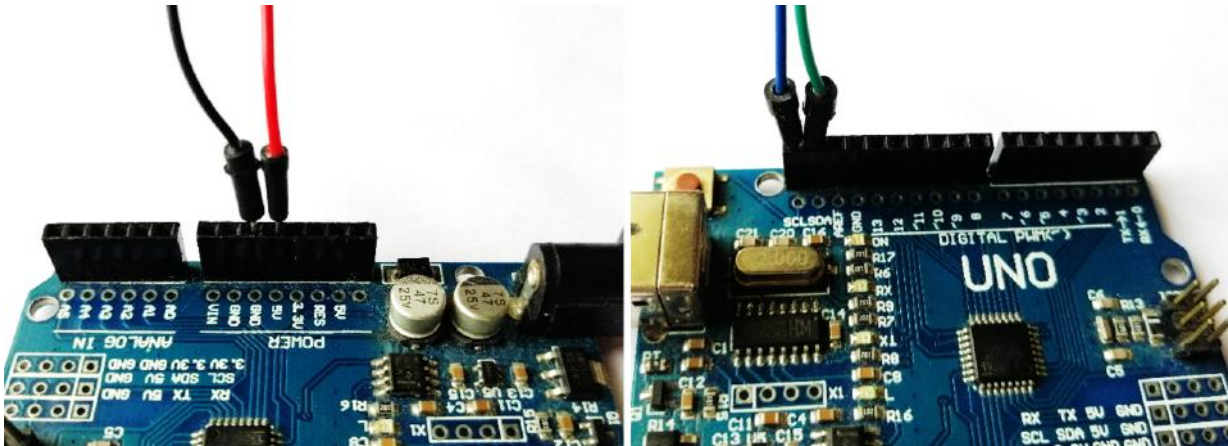
Nema puno novog za reći spajanju preko USB veze, ali vjerojatno niste čuli za I^2C . Sabirnica *Inter-Integrated Circuit* (skraćeno I^2C) dvožično je serijsko sučelje koje je izvorno razvila korporacija Phillips za upotrebu u potrošačkoj elektronici. Slijedi hijerarhiju nadređenog/podređenog (eng. *master/slave*), pri čemu se nadređeni definira kao uređaj koji daje takt sabirnici, komunicira s podređenima te piše ili čita podatke u i iz registara u njima. Podređeni uređaji reagiraju samo kada ih nadređeni traži, preko njihove jedinstvene adrese. Sabirnica I^2C koristi samo dvije dvosmjerne linije, serijsku podatkovnu liniju (SDA) i serijsku liniju takta (SCL).



U opciji 1 - spajanje je vrlo jednostavno i sve što trebate učiniti je uskladiti boje na žicama i priključcima na robotu.

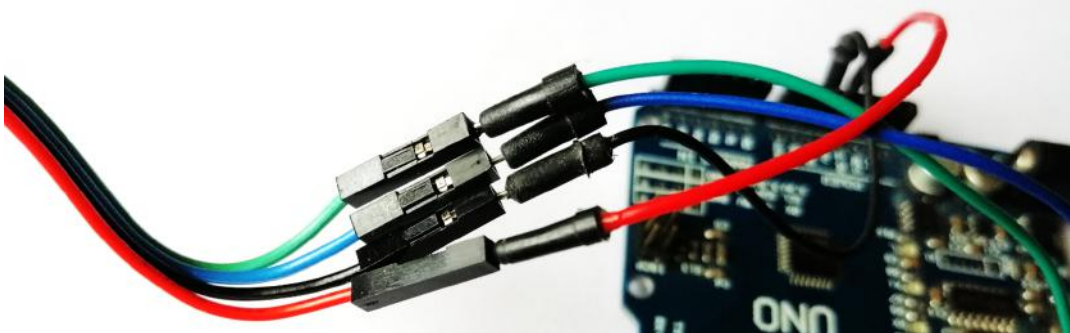


U opciji 2 imamo složeniju situaciju - treba spojiti kablove s kamere na Arduino UNO ali su oba konektora ženskog tipa pa su nam potrebna 4 kratkospojnika (muški-muški tip) koji se mogu naći u gotovo svakoj trgovini elektroničkog materijala koja prodaje komponente za samogradnju i edukaciju.



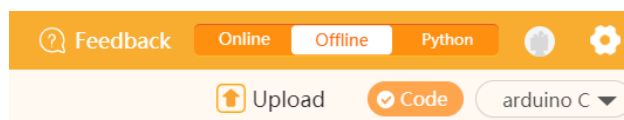
Odaberite iz kompleta 4 kratkospojnika (crvene, crne, zelene i plave boje) kako bi se podudarali s odgovarajućim žicama iz kamere. Prvo priključite crveni kratkospojnik u pin označen s 5V na Arduino UNO-u, zatim crni na GND, plavi na SCL i zeleni na SDA pin kao što je prikazano na slikama iznad.

Jedino što nam preostaje je spojiti drugu stranu ovih kratkospojnika na priključke iz kamere. Vrijedi isto pravilo - uskladite boje kratkospojnika s bojama priključnih vodova iz kamere (crna na crnu, crvena na crvenu itd.).



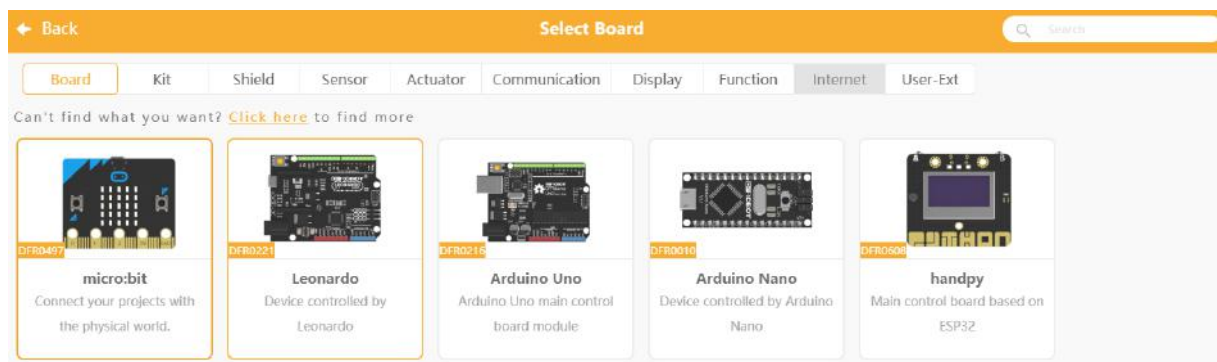
Sada imate kameru spoјenu na mikrokontroler. Priključite USB kabel na Maqueen plus robota ili Arduino UNO ploču i povežite ga s prijenosnim računalom ili računalom.

Idite na: <http://mindplus.cc/download-en.html> i preuzmite verziju za operacijski sustav vašeg računala. Instalirajte i pokrenite Mind+. Prvo prijedite u izvanmrežni način rada (*offline*).

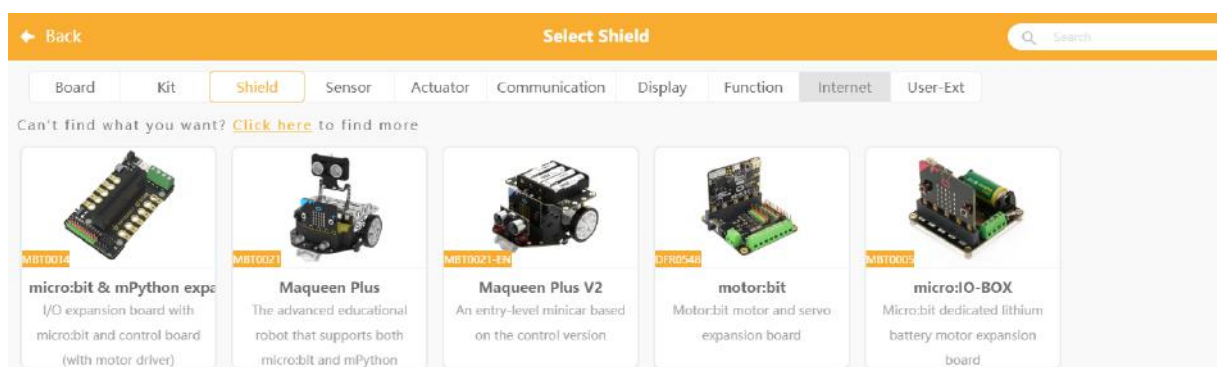




Otvorite proširenja i odaberite za ploču (**Board**): Opcija 1) micro:bit (ako radite s Maqueen plus ili Opcija 2) Arduino UNO

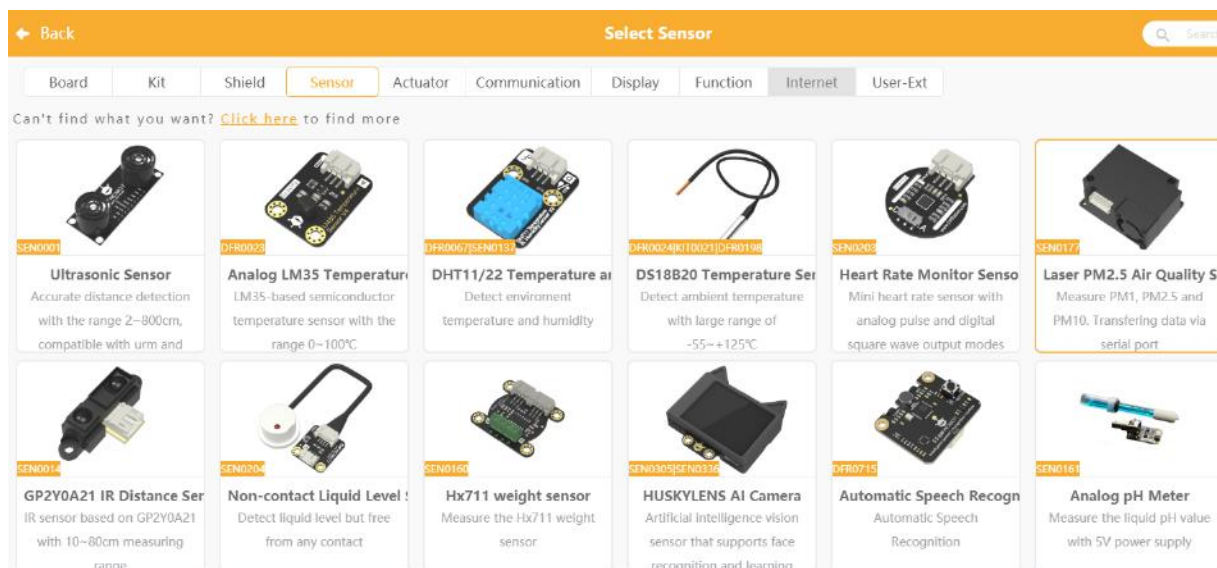


Samo za opciju 1) Prijedite na karticu **Shield** i odaberite Maqueen Plus ili Maqueen Plus V2 (ovisno o vašoj verziji robota)



I opcija 1) i opcija 2)

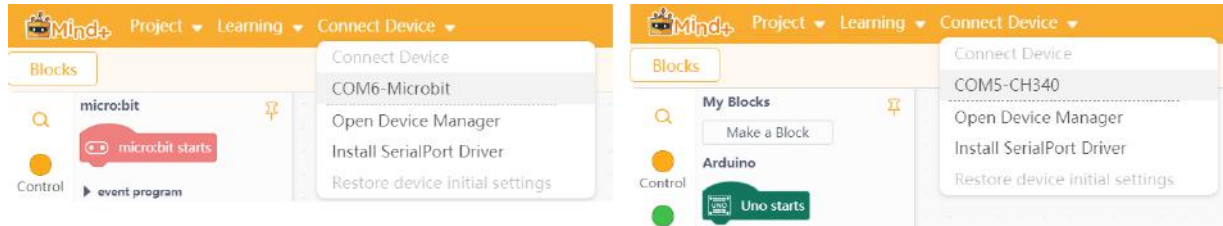
Prebacite se na karticu **Sensor** i odaberite senzor - HUSKYLENS AI camera





Nakon odabira kliknite na **<- Back** (natrag) i spremni ste za korištenje odabrane ploče/robota i kamere. Napravimo test da vidimo radi li sve kako treba.

Prije samog testiranja morate spojiti svoj uređaj. Kliknite na **Connect Device** (poveži uređaj) i odaberite svoj uređaj.

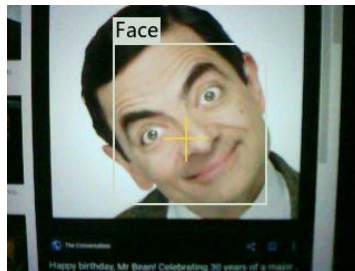


Uzmite kameru i potiskujte funkcijski gumb ulijevo sve dok se na vrhu zaslona ne prikaže riječ **Face Recognition** (prepoznavanje lica).



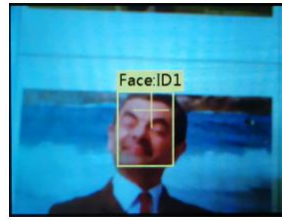
Učenje i detekcija

1. Detekcija lica: Usmjerite HuskyLens na bilo koje lice. Kada se lice detektira, ono će se na zaslonu kamere automatski uokviriti bijelim okvirom na kojem će pisati **Face** (Lice).



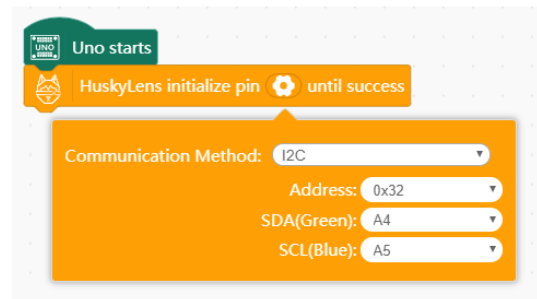
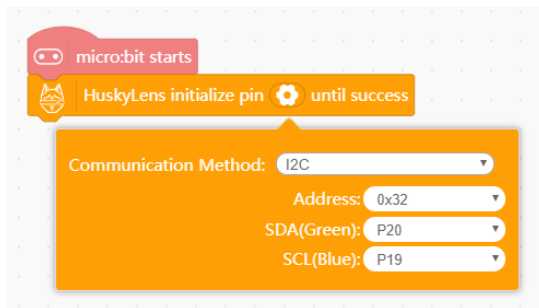
Savjeti: Ako želite da HuskyLens nauči ili prepozna vaše lice, morate okrenuti kameru prema sebi i u tom trenutku ne možete vidjeti zaslona, ali status detekcije možete odrediti prema boji LED indikatora.

2. Učenje lica: Usmjerite simbol + na lice, kratko pritisnite gumb za učenje (**Learn Button**) kako biste naučili lice. Ako HuskyLens otkrije naučeno lice, na zaslonu će se prikazati plavi okvir s riječima **Face: ID0**, što ukazuje na to da je HuskyLens naučio lice i dodijelio mu identifikator (**ID0**) te da ga sada može prepoznati.

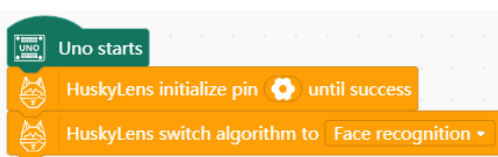
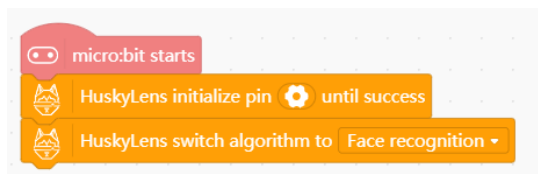


Posjetite: https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336#target_15 za dokumentaciju vezanu uz prepoznavanje lica.

Idite na Mind+ i počnite programirati. Nakon se micro:bit/Uno pokree koristimo HuskyLens blok za inicijaliziranje veze, odnosno pinova i konfiguriramo ga za I²C kao što je prikazano na slici ispod za svaku od opcija.

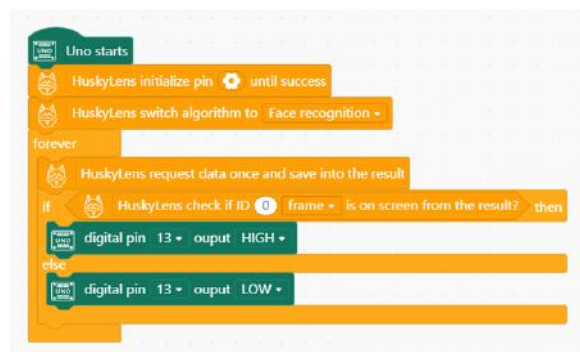


Zatim upotrijebite blok *HuskyLens switch algorithm to Face recognition* za obje opcije.



Slijedi beskonačna koja otkriva je li lice na kameri prepoznato.

Ako se u opciji 1 otkrije lice), vidjet ćete lice na micro:bit zaslonu. Ako nije, vidjet ćete znak X. Ako se u opciji 2 otkrije lice), uključit će se ugrađena LED dioda (pin D13). Ako nije, LED je isključen





Pritisnite **Upload** (prenesi) da biste prenijeli ovaj kôd u micro:bit ili u Arduino UNO.



Nakon prijenosa koda, usmjerite kameru prema "naučenom" licu i LED-u na D13 bi trebao biti uključen.

Ako lice nije na zaslonu kamere, D13 je isključen.

Ako radi kako je opisano – sve je u redu i spremno za upotrebu u našem mobilnom AI robotu.

ZAKLJUČAK

HuskyLens je AI senzor računalnog vida jednostavan za korištenje sa 7 ugrađenih funkcija: prepoznavanje lica, praćenje objekata, prepoznavanje objekata, praćenje linije, prepoznavanje boje, prepoznavanje oznaka i klasifikacija objekata.

Kroz UART/I2C priključak, HuskyLens se može povezati s Arduinoom i micro:bitom kako bi vam pomogao da napravite vrlo kreativne projekte bez suočavanja sa složenim algoritmima.

Promisli!



Što sada znam?



Što sam naučio/la?



Što još želim znati?

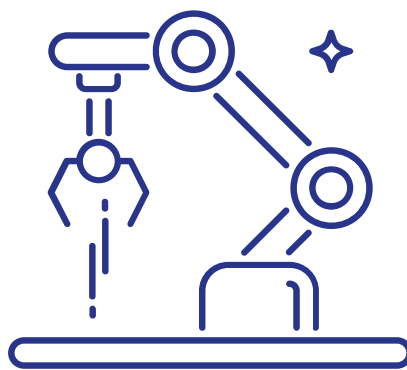




Sastavljanje robota

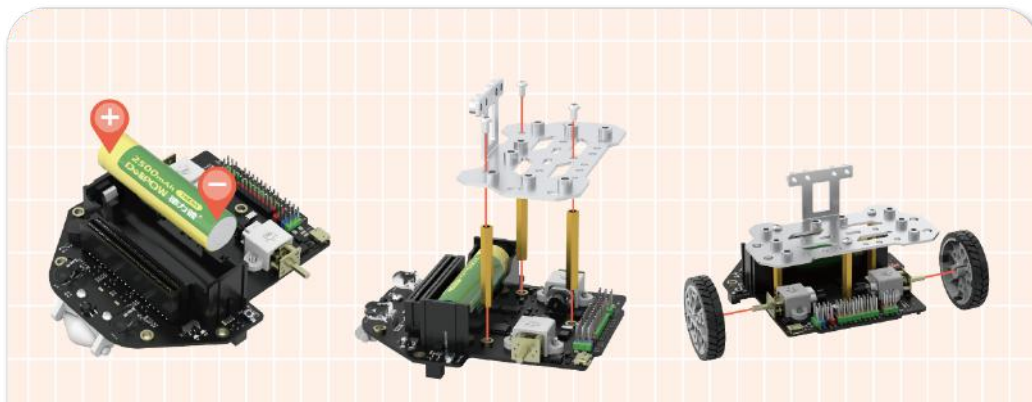
Dio ovog kurikuluma povezan je sa stvarnim fizičkim uređajem s AI mogućnostima. Hrvatska robotički savez razvio je male isplative mobilne robotske platforme za implementaciju programskih znanja iz prethodnih lekcija.

Sastavimo ARTIEBot-a





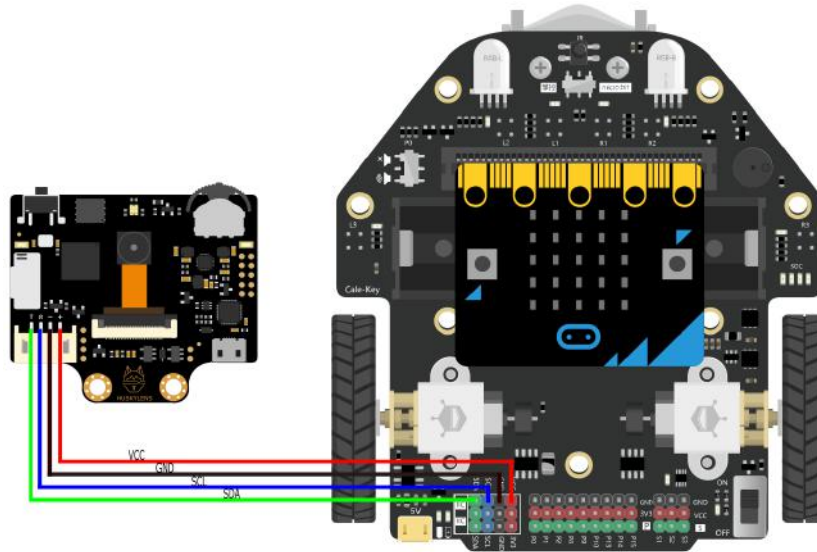
Opcija 1a - Sastavljanje robota V1 Maqueen Plus (preporučuje se početnicima)



1) umetnite bateriju iz 18650. 2) instalirajte nosač 3) pričvrstite kotače



4) Montirajte HuskyLens kameru 5) Postavite micro:bit ploču u utor



6) Povežite Maqueen Plus sučelje s HuskyLens kamerom

Za punjenje baterije tipa 18650 koristite USB mikro priključak (priključak za punjenje) na stražnjoj strani robota:



Battery indicator

Charging port

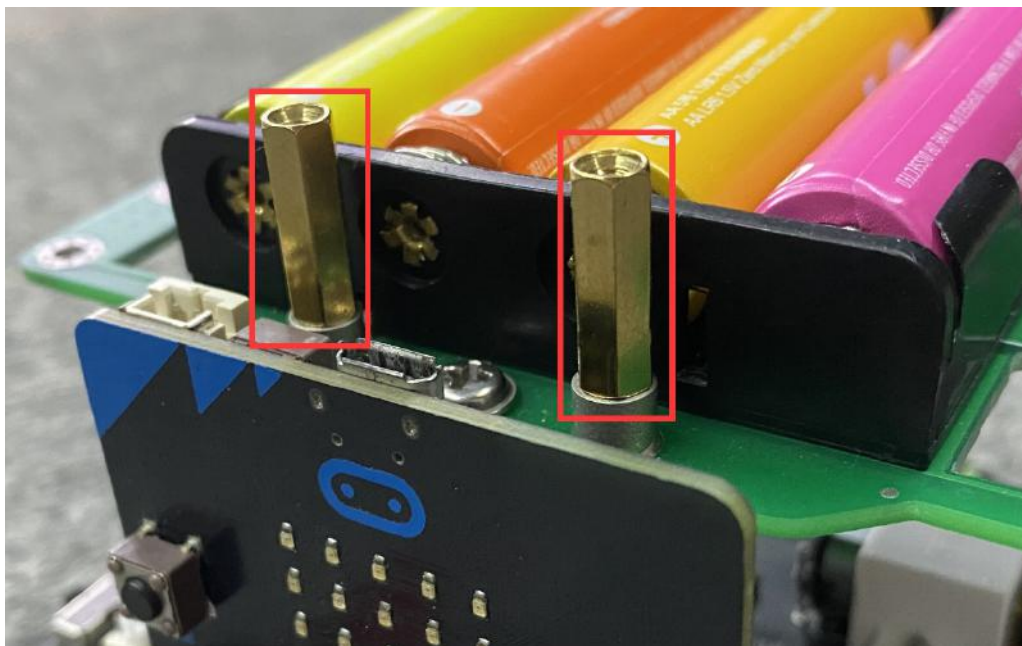
Kada se baterija potpuno napuni, sve LED diode upale se sve LED diode. LED diode će se isključiti jedna po jedna kako se kapacitet baterije postupno smanjuje. Ako se sva svjetla ugase, bateriju treba napuniti. Koristite punjač za pametne telefone i povežite ga s priključkom za punjenje.

Imajte na umu da je priključak za punjenje služi samo za punjenje. Za programiranje robota Maqueen Plus - koristite USB mikro konektor na micro:bitu.

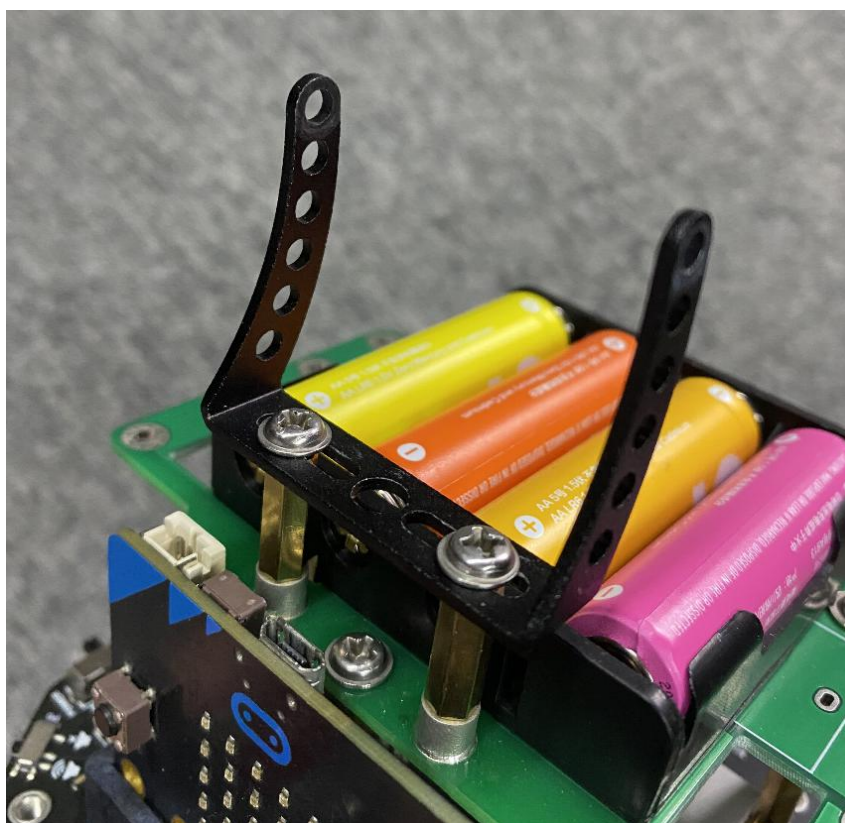
Opcija 1b - Sastavljanje Maqueen Plus robota V2 (također se preporučuje početnicima)

Robot Maqueen plus V2 dolazi unaprijed sastavljen, a na njega jednostavno morate postaviti HuskyLens kameru

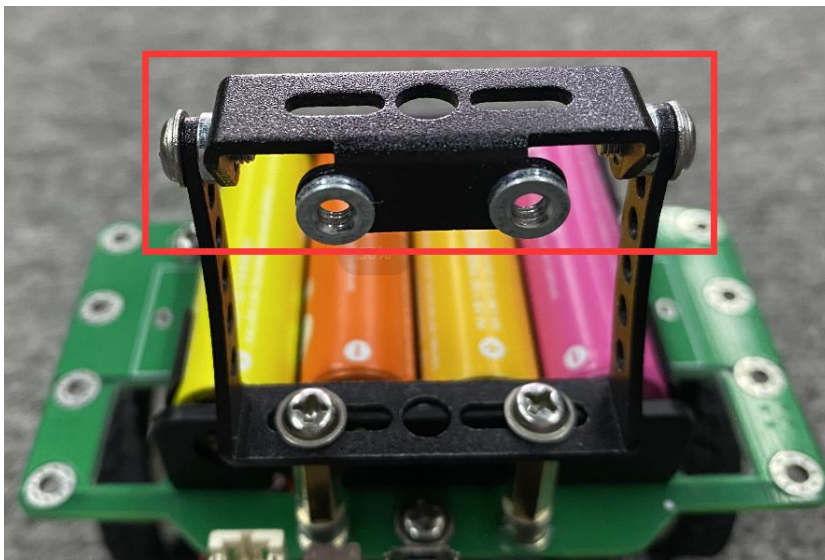
1. Ugradite dva bakrena odstoynika (distance) koji se nalaze u kompletu materijala na mjesta kao što je prikazano na slici.



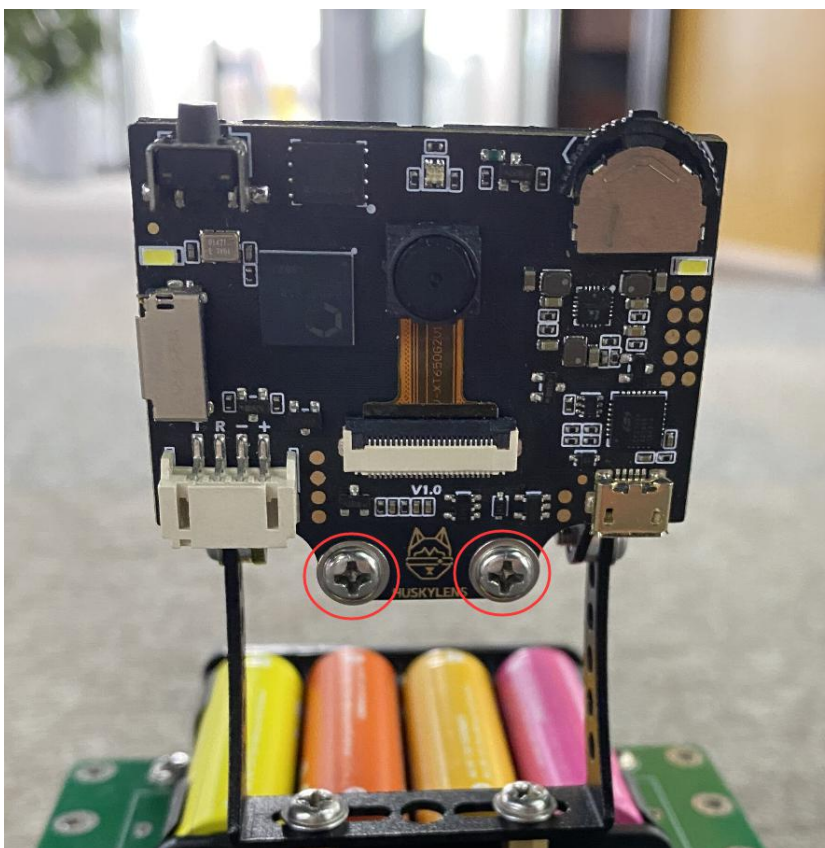
2. Pričvrstite nosač u obliku luka (nosač i montažni vijci opremljeni su proizvodom HuskyLens) na bakreni stup vijcima.



3. Ugradite drugi nosač (nosač i montažni vijci su u kompletu materijala dobivenim uz HuskyLens)

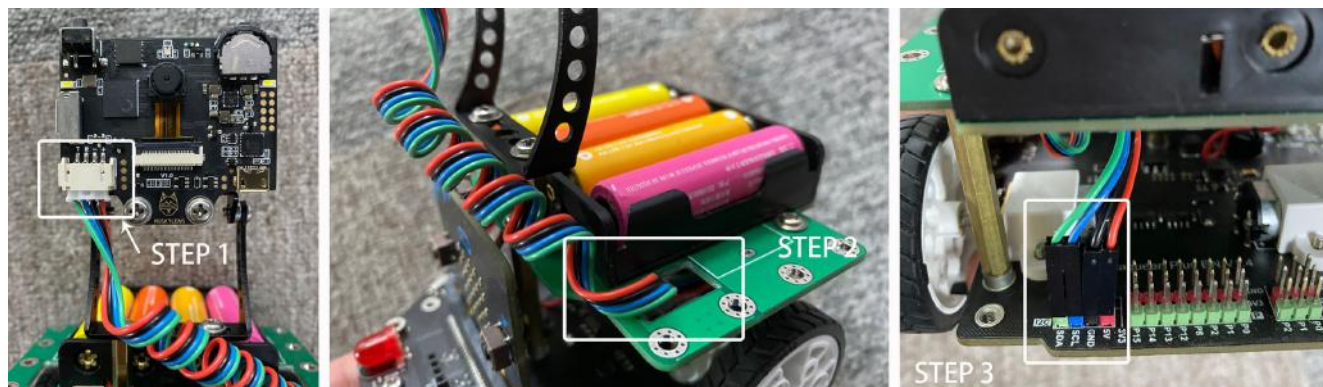


4. Pričvrstite HuskyLens AI kameru

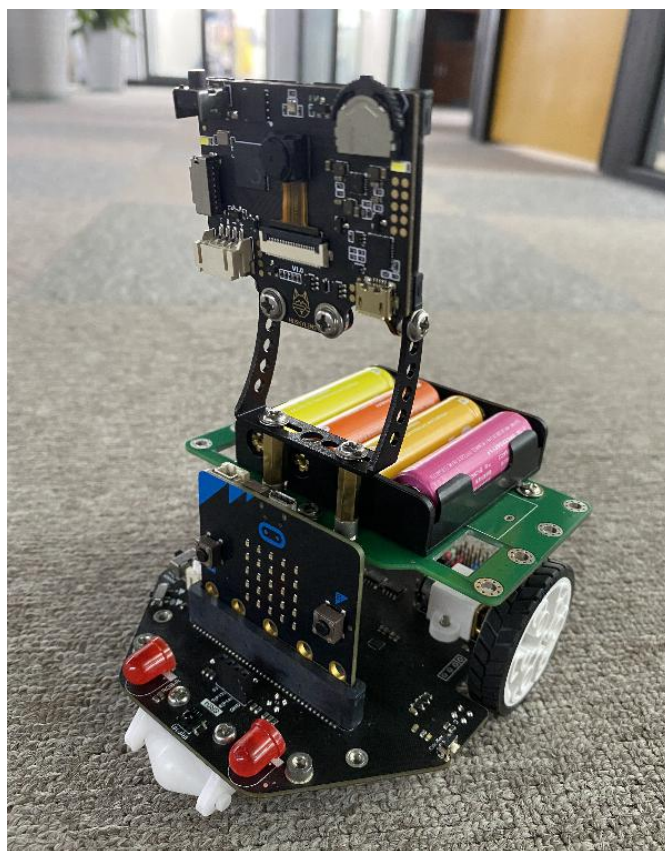




5. Povezivanje AI kamere sa sučeljem robota



6. Instalacija kamere je završena.





Opcija 2 - Sastavljanje Arduino robota (samo za iskusne "uradi sam" korisnike)

Popis materijala (sastavnica)

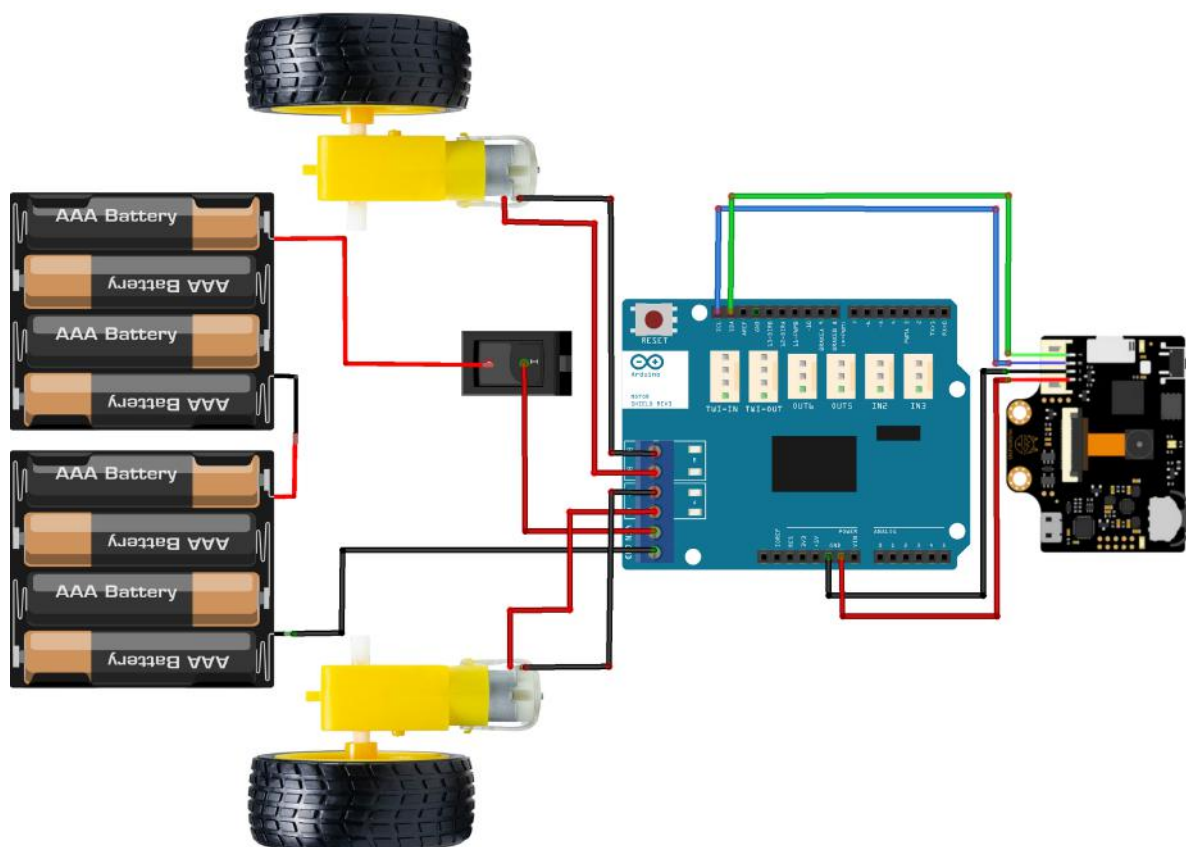
- 1 x Komplet materijala za pametni auto (*smart car*)
<https://botland.store/chassis-for-robots/7283-chassis-rectangle-2wd-2-wheel-robot-chassis-with-dc-motor-drive-5904422335649.html>
- 1 x Set kratkospojnika (*jumper*s)
<https://botland.store/various-wires/1022-connecting-cables-male-male-65pcs.html>
- 1 x Držač baterije (4 x AA)
<https://botland.store/battery-holders/173-battery-holder-4-x-aa-r6-5904422329389.html>
- Razni vijci i matice
<https://botland.store/screws-and-nuts/637-screws-nuts-and-washers-set-330pcs-5410329304478.html>
- 1 x Arduino UNO (original ili klon)
<https://store.arduino.cc/collections/boards/products/arduino-uno-rev3>
<https://botland.store/arduino-compatible-boards-dfrobot/2683-dfduino-uno-v3-compatible-with-arduino.html>
- 1 x Arduino štit za motore R3 (*Arduino motor shield R3*)
<https://store.arduino.cc/collections/shields/products/arduino-motor-shield-rev3>
- 1 x HuskyLens AI kamera
<https://store.arduino.cc/collections/dfrobot/products/gravity-huskylens-ai-machine-vision-sensor>
- 8 x AA baterije (pokušajte koristiti punjive)
- Vezice za kablove (ako želite fiksirati žice i držače baterija)

Potrebni alati

- Odvijači (trebat će vam jedan s ravnim i jedan s križnim vrhom)
- Lemilica
- Izolacijska traka ili toplinski skupljajuće cijevi (*heatshrink tubes*)
- Kliješta za rezanje

Shema

Ovdje je shema ožičenja. Primijetite da ćemo koristiti dvostruki držač baterije u serijskom priključku tako da je svaka od baterija u držaču spojena (+ na - i - na +). 8 AA baterija trebalo bi osigurati veći napon za napajanje potrošača i spriječiti resetiranje Arduina ili HuskyLensa kada motori počnu raditi. Alternativno, u serijskoj vezi možete koristiti i 3 baterije tipa 18650.

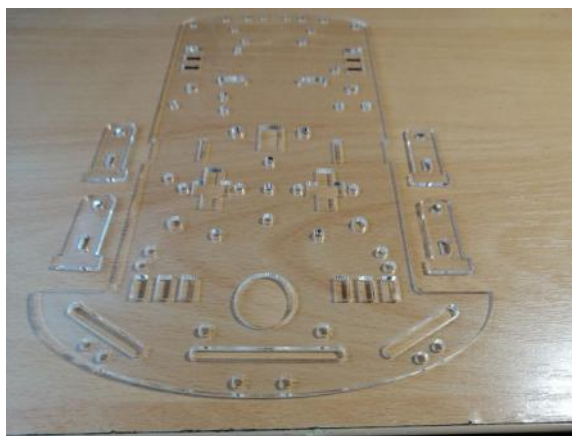


Komplet materijala za pametni auto - otvorite plastičnu vrećicu s dijelovima

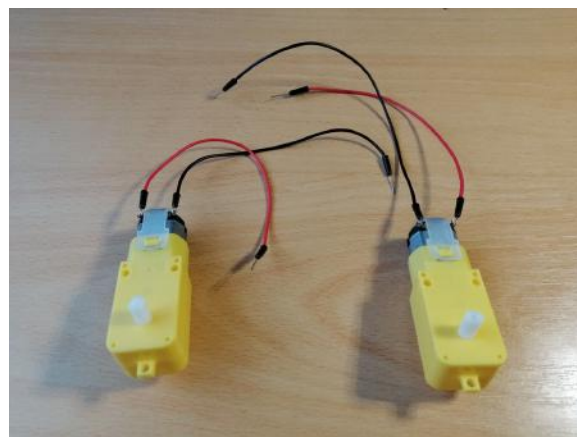
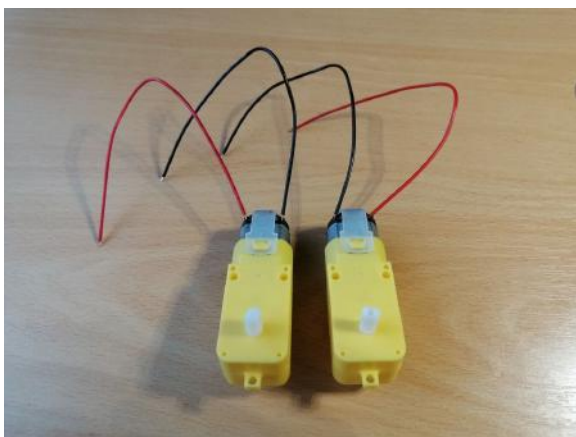




Ogulte zaštitnu foliju s ploče i nosača motora



Zalemite žice (ili krtkospojnike) na motore

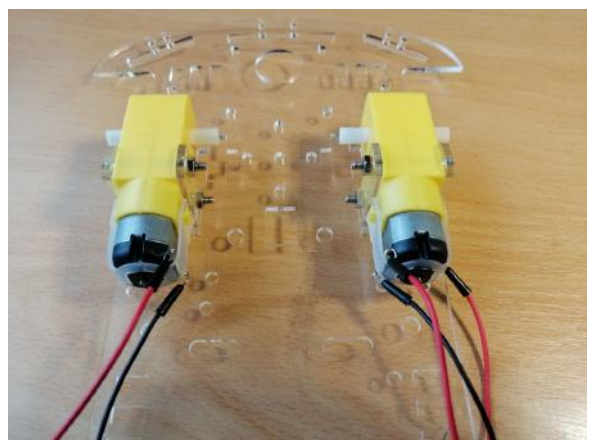
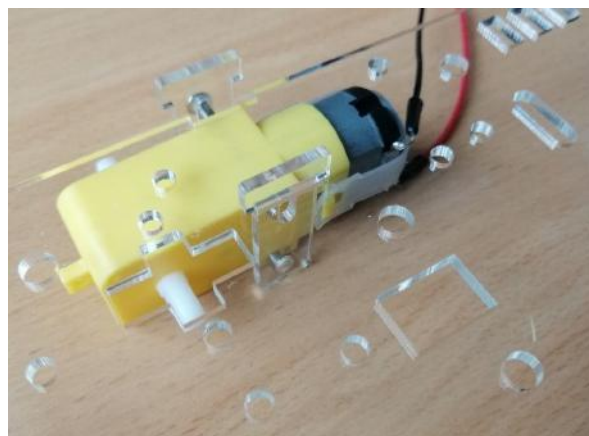
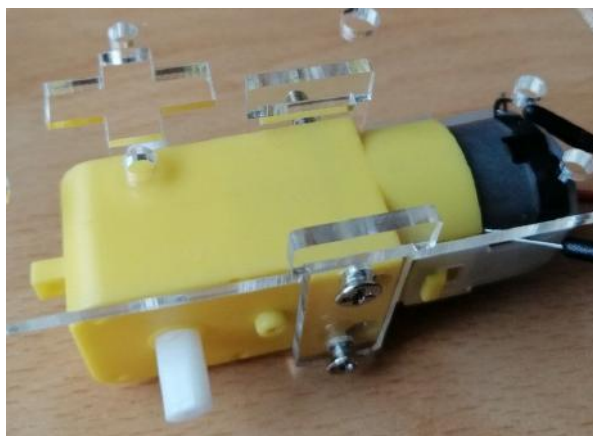


Pogledajte ovaj kratki videozapis da biste vidjeli kako se to radi:

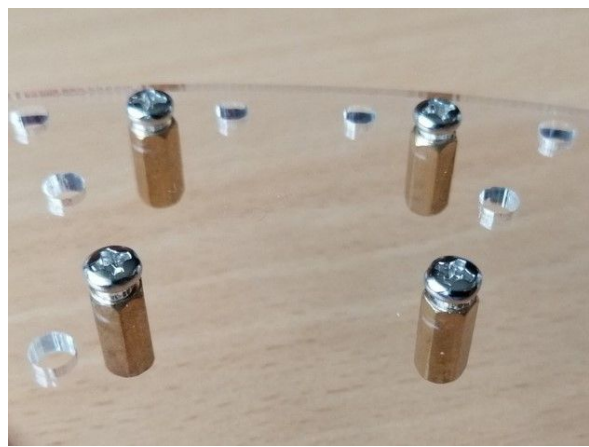
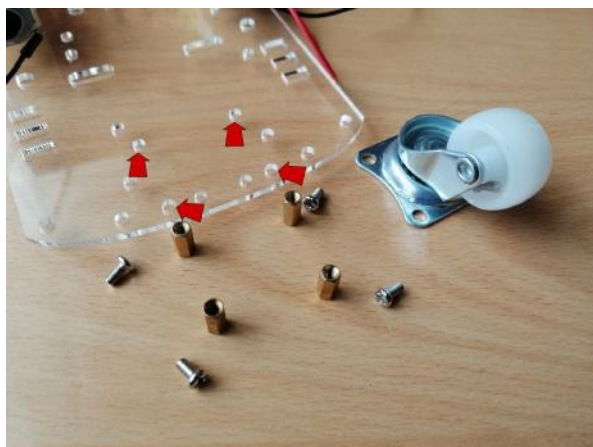
<https://www.youtube.com/watch?v=xSWKnnvGWBs>

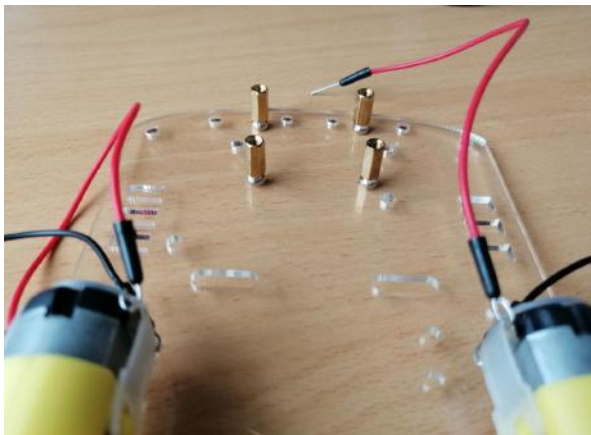
Zamolite svog učitelja ili roditelja da vam pomogne u tome ako nemate prethodno iskustvo ili alate.

Pričvrstite motore na "T nosače" pomoću vijaka i matica

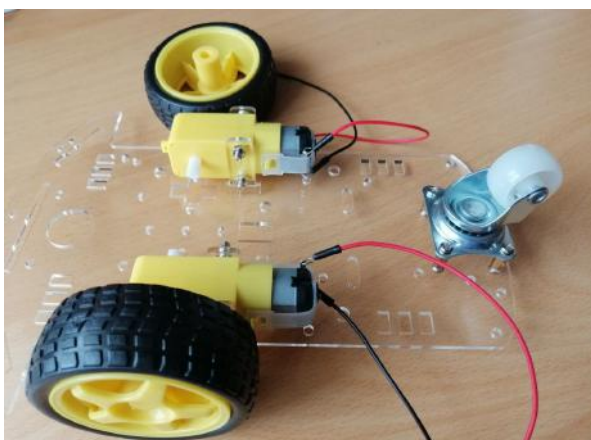


Pričvrstit mali (stražnji) kotač pomoću vijaka i odstoynika (distanci)

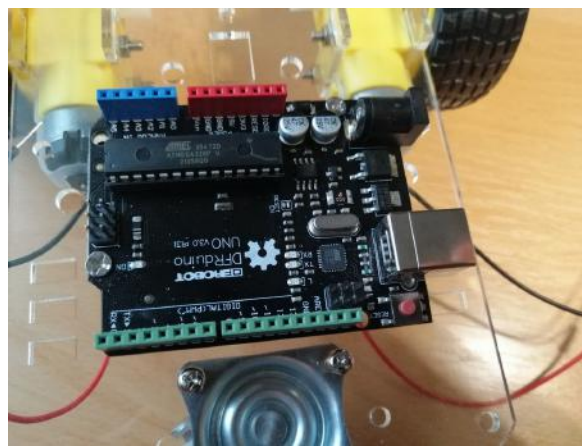
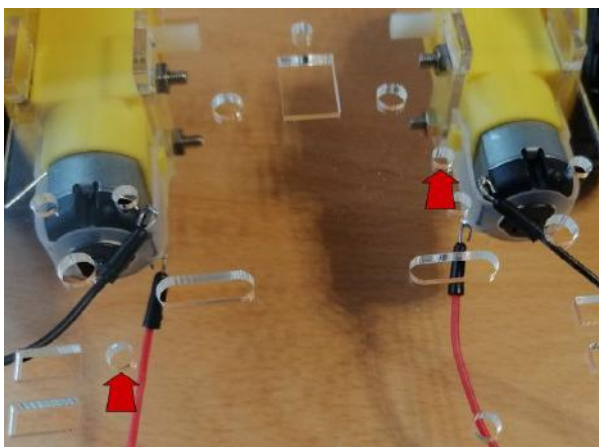


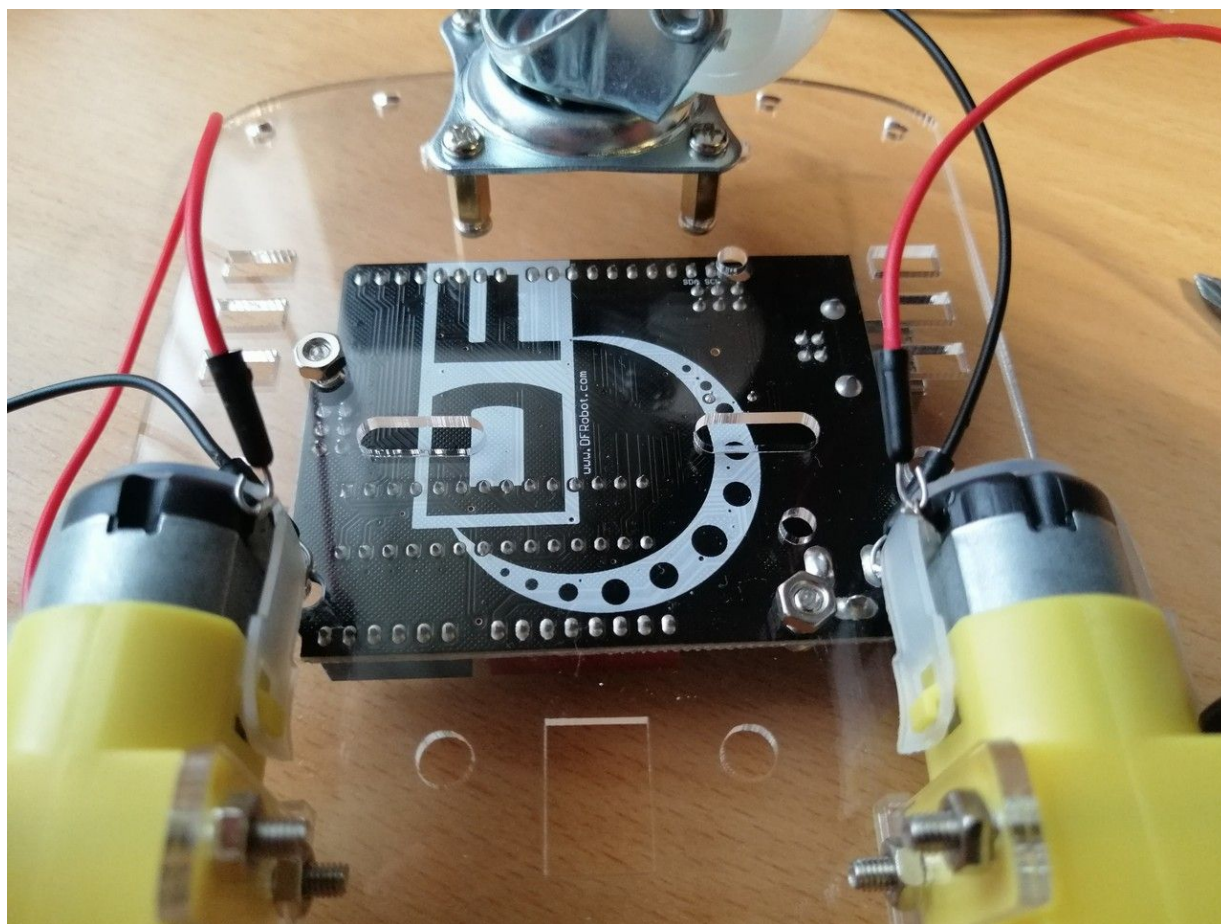


Pričvrstite kotače na osovine motora - pogledajte oblik osovine i oblik otvora kotača



Pričvrstite Arduino UNO ploču u označene rupe

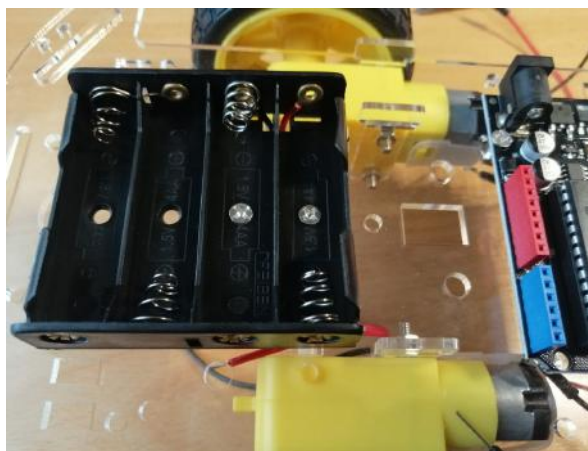
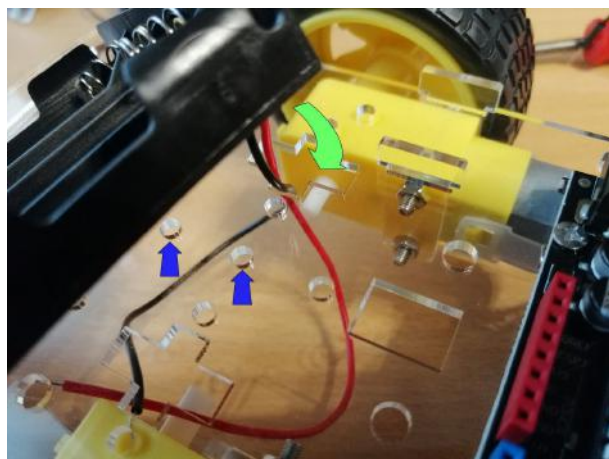


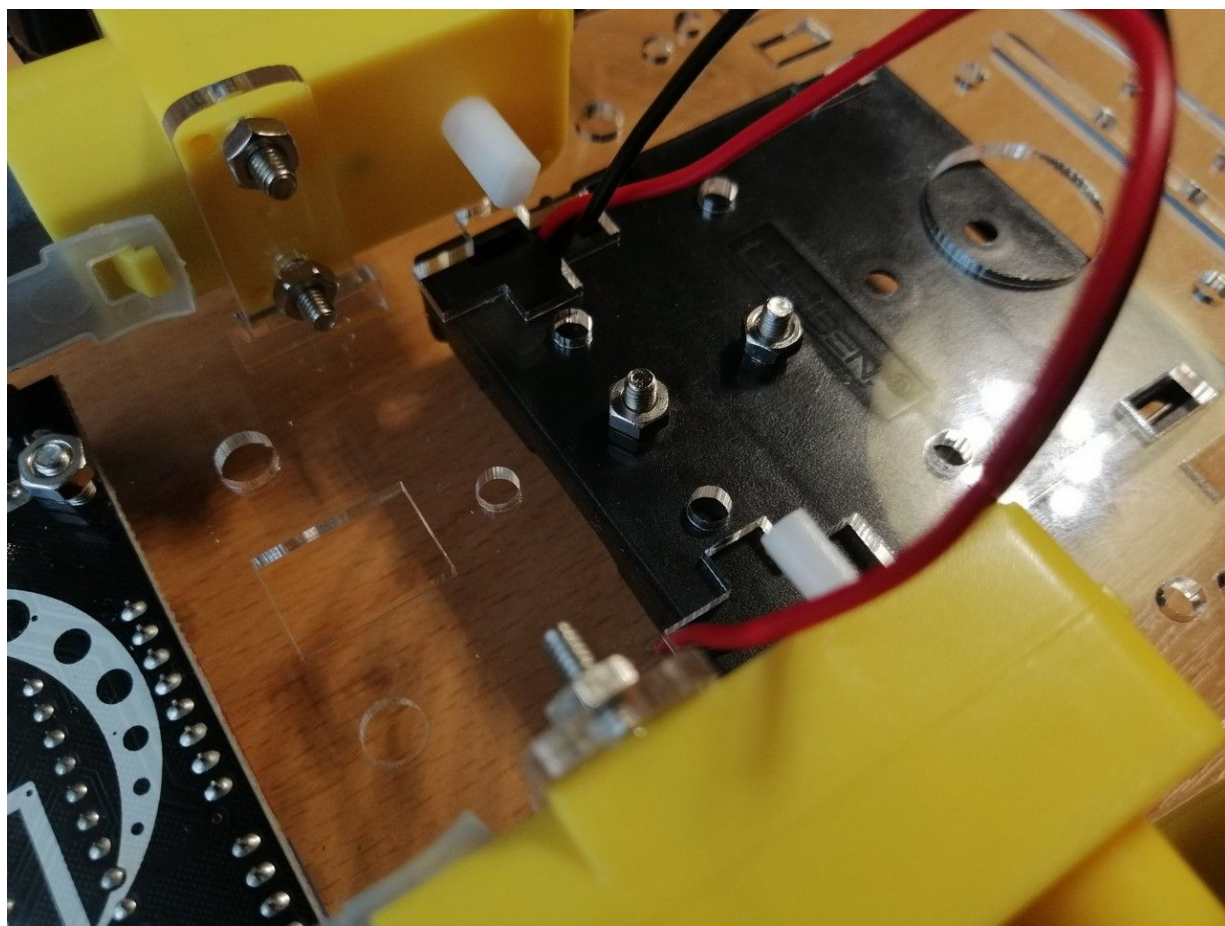


U ovom videozapisu možete vidjeti neke detalje, iako koristimo malo drugačiji pristup:

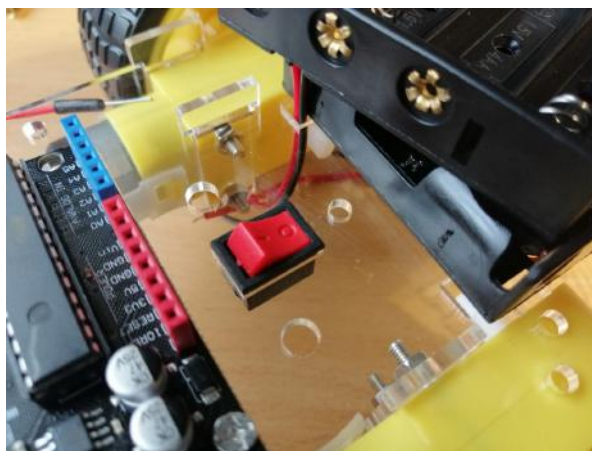
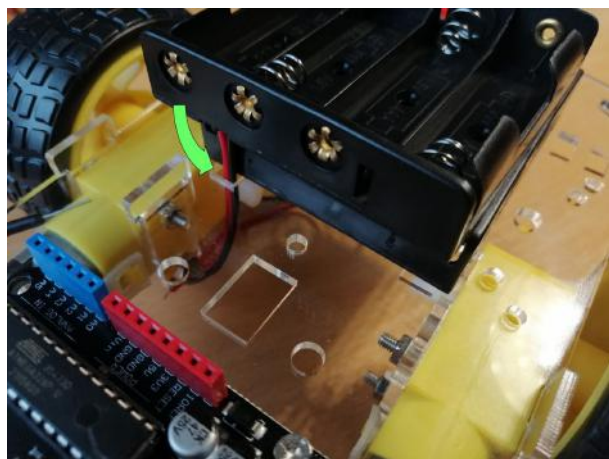
<https://www.youtube.com/watch?v=3a-bE1VlaU8>

Korak 10) Pričvrstite prvi (donji) držač baterije pomoću vijaka u označene rupe (plave strelice)
Prvo provucite kabele držača kroz rupu (označena zelenom strelicom)

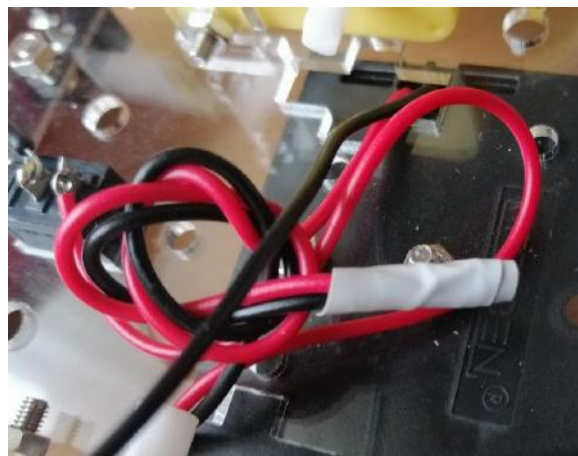
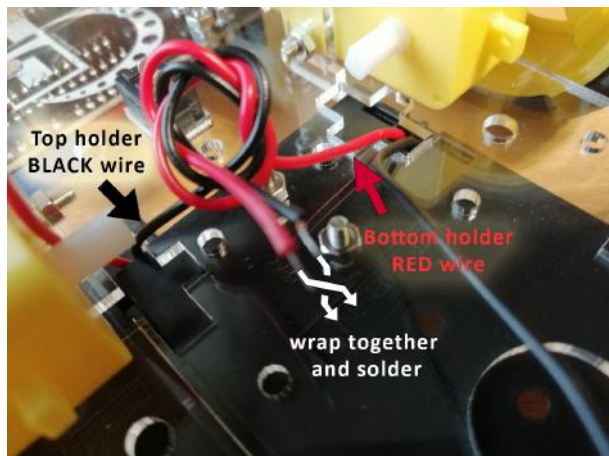




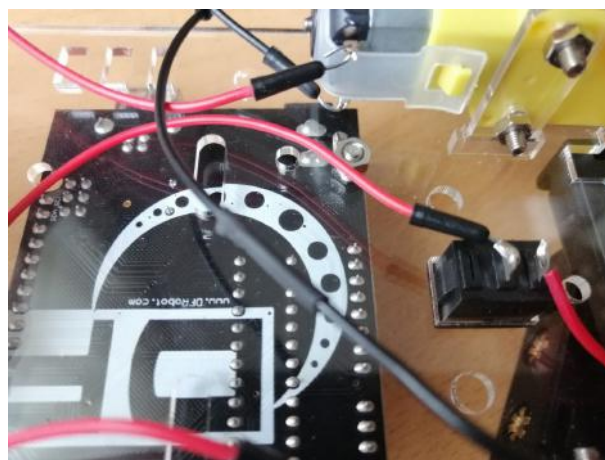
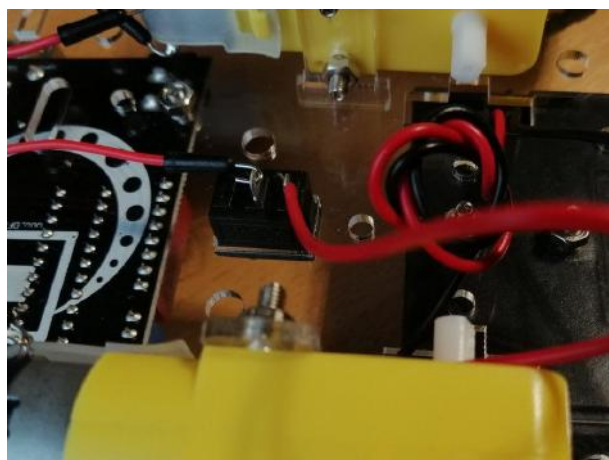
Izvucite kabel drugog (gornjeg) držača baterije kroz označenu rupu (zelena strelica)
Postavite prekidač u rupu pravokutnog oblika koja se nalazi između Arduino UNO-a i držača baterije



Spojite crvenu žicu s donjeg držača baterije s crnom žicom gornjeg držača baterije
Izolirajte spoj toplinski skupivim cijevima ili izolacijskom trakom.

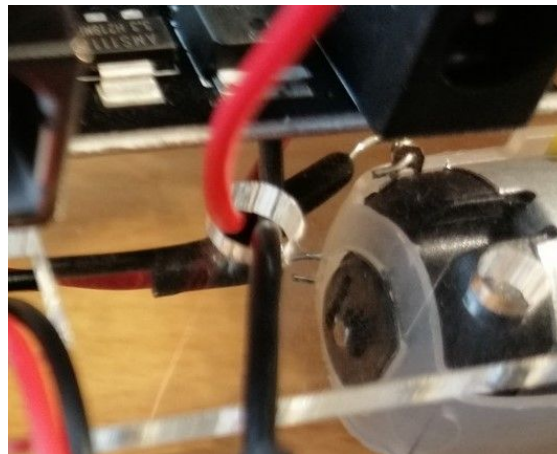
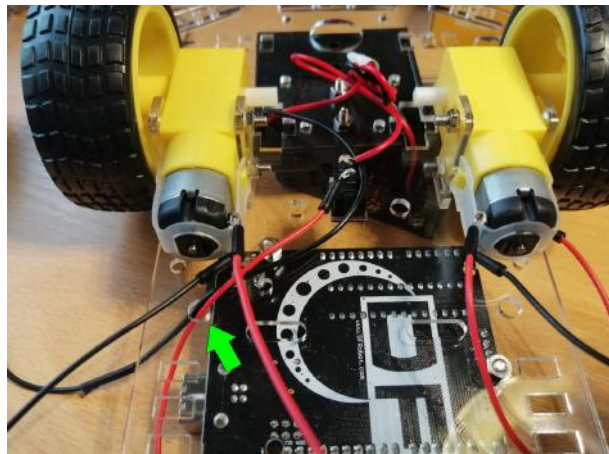


Zalemite crvenu žicu od donjeg držača na prekidač
Nakon toga lemite crveni kratkospojnik (ili žicu) na drugi kontakt prekidača. Lemite crni kratkospojnik na donji držač crne žice. Izolirajte spoj toplinski skupivim cijevima ili izolacijskom trakom.

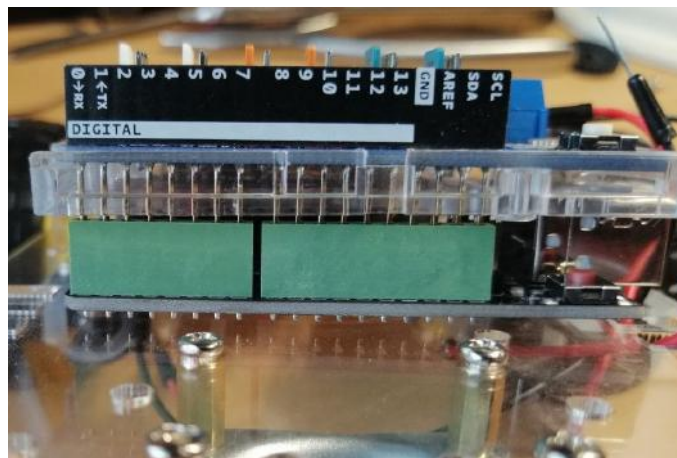
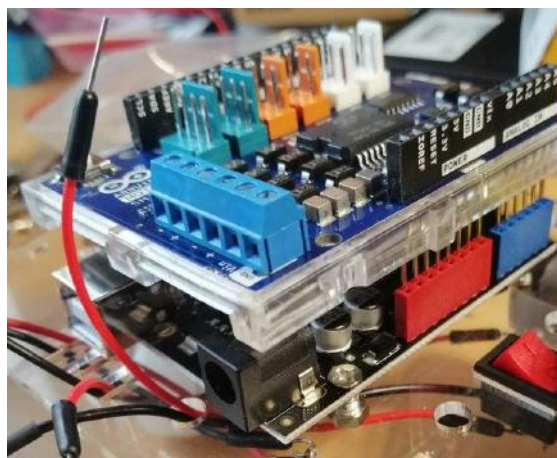




Izvucite crnu i crvenu žicu za napajanje kroz označenu rupu (zelena strelica)



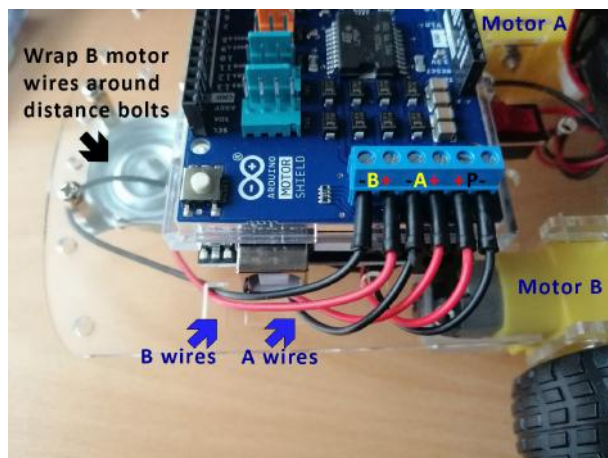
Postavite Arduino štit za motore u Arduino UNO ploču
Poravnajte nožice (pinove) štita i Arduina UNO te potisnite štit prema dolje.





Provucite žice motora kroz označene rupe.

Skratite vrhove kratkospojnika klijestima za rezanje i spojite ih u priključke štita za motore pritežući vijke kako je označeno (- Motor B + - Motor A + + Napajanje P -).



Otvorite kutiju HuskyLens AI kamerom i izvadite komplet materijala



Pričvrstite manji nosač na kameru - koristite vijke iz kompleta materijala





Pričvrstite veći nosač kamere na kućište robota



Spojite međusobno dva nosača vijcima i priključite kabel kamere

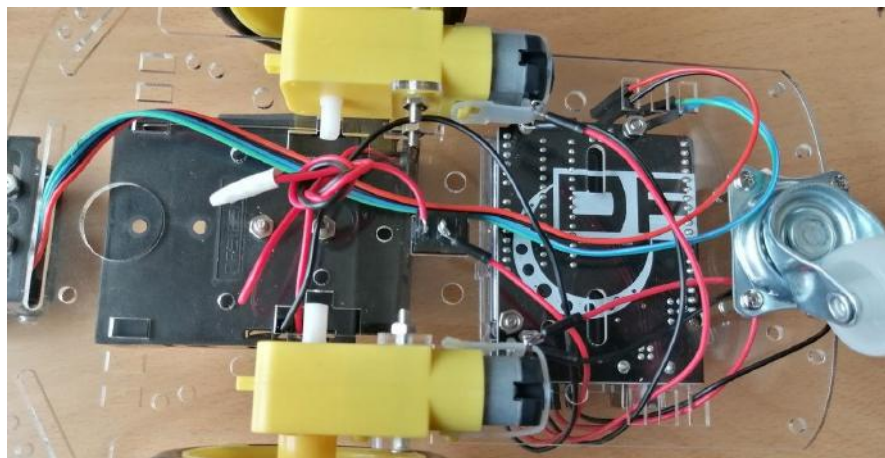


Izvucite kabel kamere kroz rupu

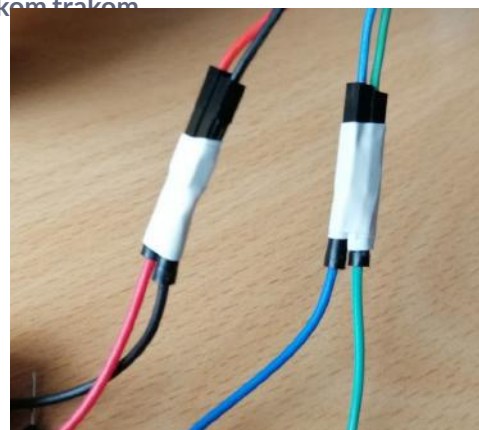
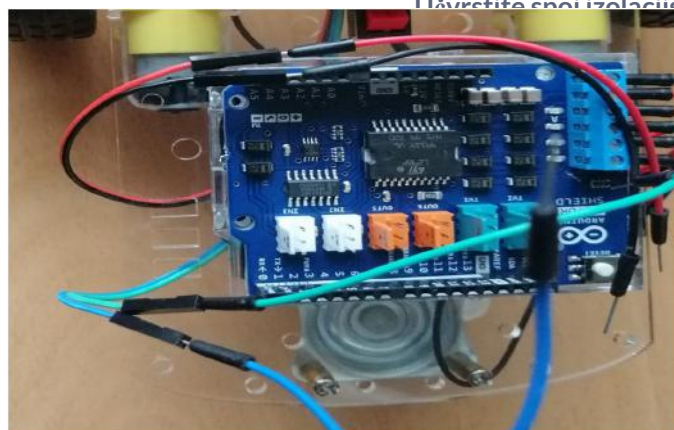




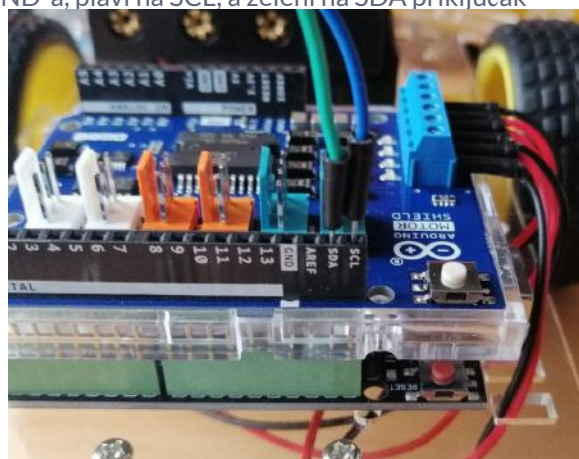
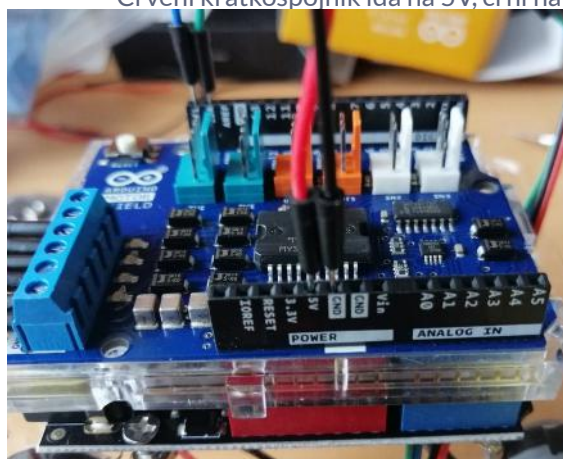
Okrenite robota i provucite kabel kamere kroz rupe kao na slici ispod



Spojite kratkospojnike na kabel kamere - koristite kratkospojnike s odgovarajućim bojama
Izvrstite spojeve izolacijskom trakom



Priključite kratkospojnike napajanja i I2C veze na štít motora
Crveni kratkospojnik ide na 5V, crni na GND-a, plavi na SCL, a zeleni na SDA priključak





Stavite baterije u oba držača

Koristite 2 vezice za kablove kako bi pričvrstili gornji i donji držač baterije (ne prejako zategnuti - trebali biste moći pomicati gornji držač naprijed-natrag)



Uključite prekidač - ako ste dobro spojili sve - uključiti će se Arduino UNO i HuskyLens kamera.



Promisli!



Što sada znam?



Što sam naučio/la?



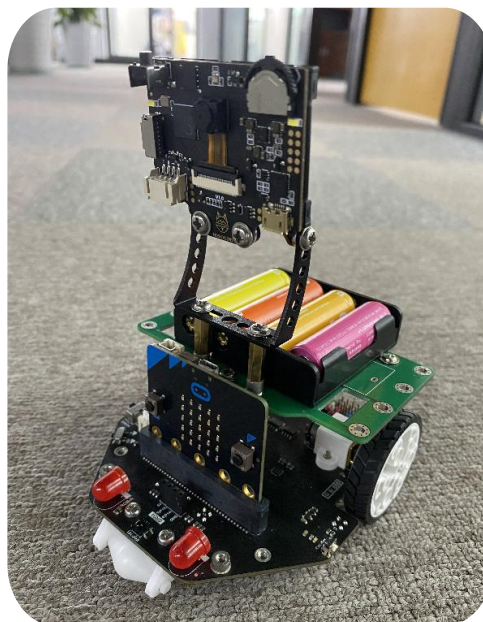
Što još želim znati?



Programiranje kretanja robota

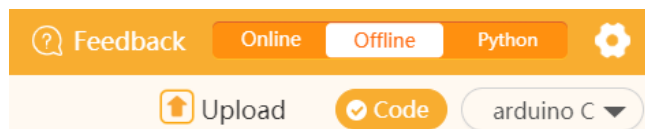
Roboti su cool. U ovoj lekciji prikazujemo korak po korak, lako shvatljive primjere programiranja kretnji autonomnog mobilnog robota.

Napisati ćemo naš prvi program s kojim ćemo pokrenuti robota.

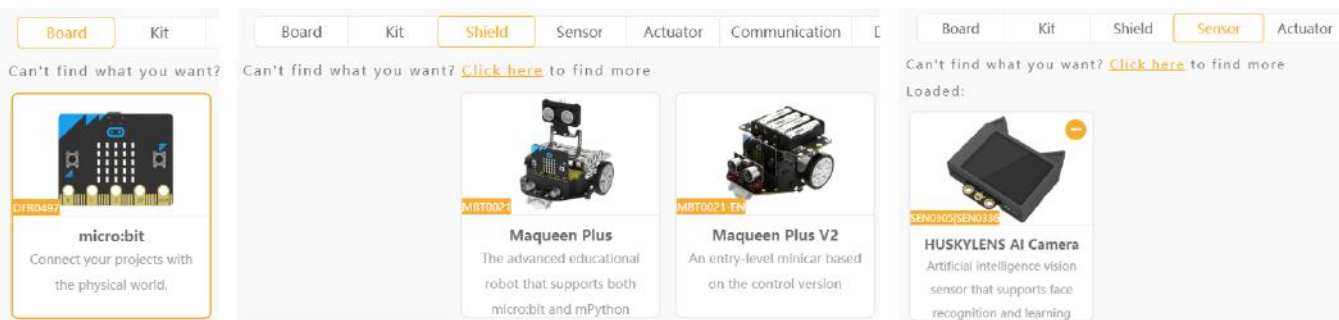




Korak 1 (obje opcije): idite na: <http://mindplus.cc/download-en.html> i preuzmite verziju za operacijski sustav vašeg računala. Instalirajte i pokrenite Mind+. Nakon pokretanja prijedite u izvanmrežni način rada (Offline).



Ako radite s Arduinom preskočite izravno na korak 2.
Maqueen Plus:



Kliknite na **Extensions** i na kartici **Board** odaberite **micro:bit**, na kartici **Shield** odaberite **Maqueen Plus** ili **Maqueen Plus V2**, a na kartici **Sensor** odaberite **HUSKYLENS AI camera**. Kliknite na Back i vaš program je spreman za korištenje odabranih modula.

Povežite micro:bit s računalom putem USB mikro kabela i uključit će se indikator napajanja na pločici:



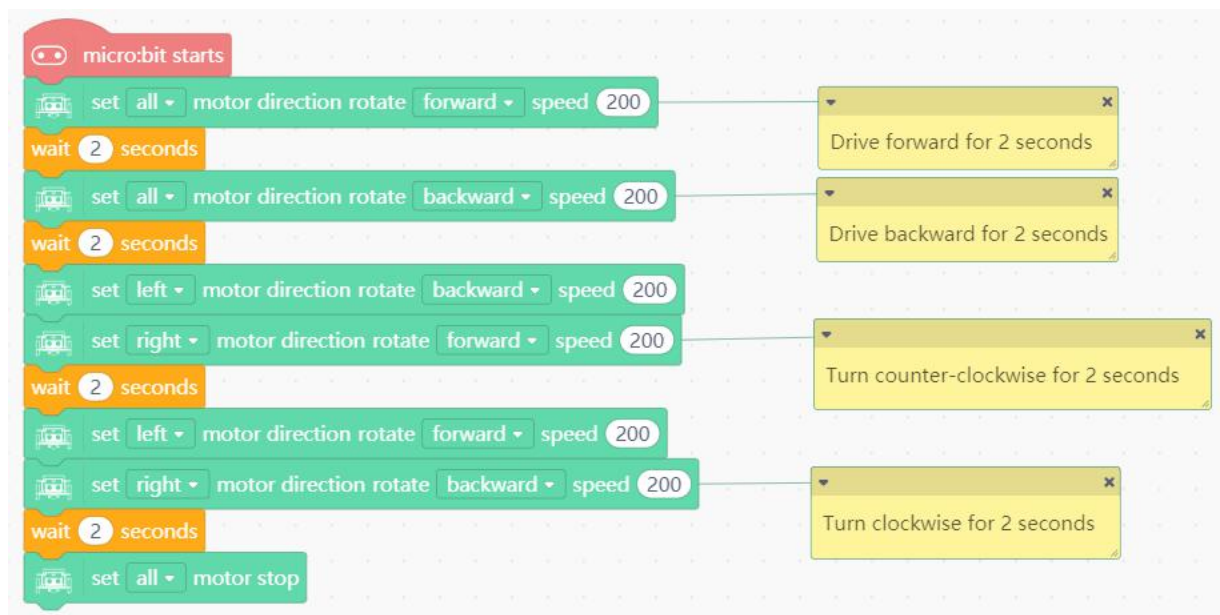


Kliknite na Connect Device i odaberite micro:bit. Po potrebi instalirajte upravljačke programe uređaja.

Testirajte prijenos na micro:bit s ovim blokovima:



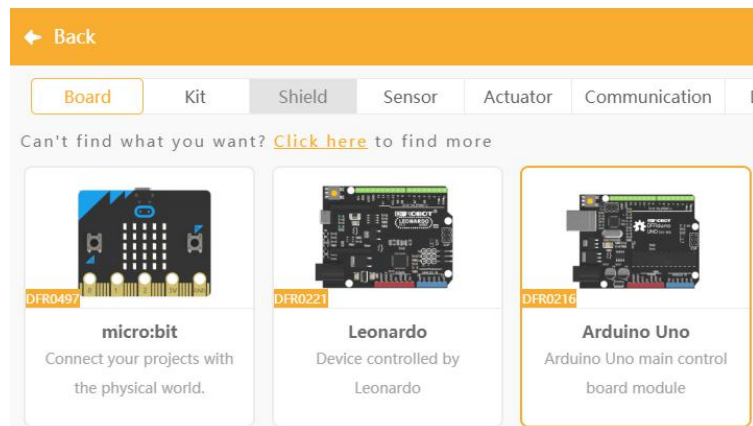
Kliknite na **Upload** (prijenos). Robot bi trebao krenuti naprijed na sekundu i stati. Pokušajte ovaj slijed - robot bi se trebao kretati kako je opisano u komentarima. (naprijed 2 sekunde, nazad 2 sekunde, rotacija u smjeru obrnutom od kazaljke 2 sekunde i rotacija u smjeru kazaljke 2 sekunde nakon čega se robot zaustavi)



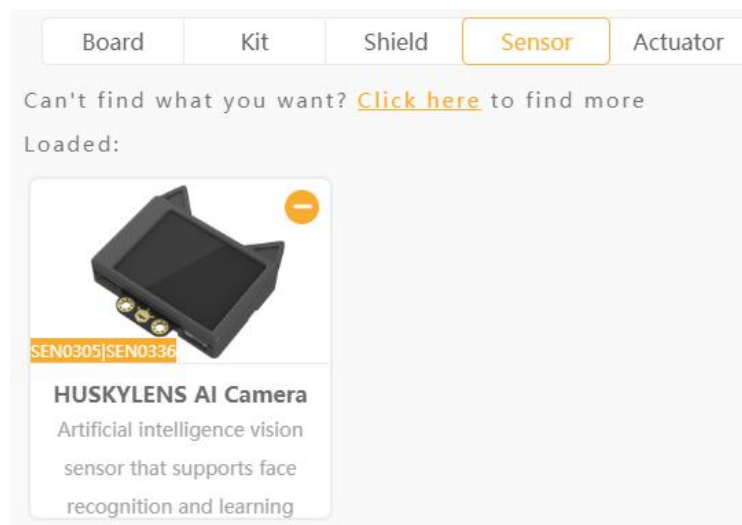
Kliknite na **Upload**. Robot bi se trebao kretati kako je opisano u komentarima (žuti blokovi). Isprobajte različite brzine kako bi vaš robot išao brže ili sporije.



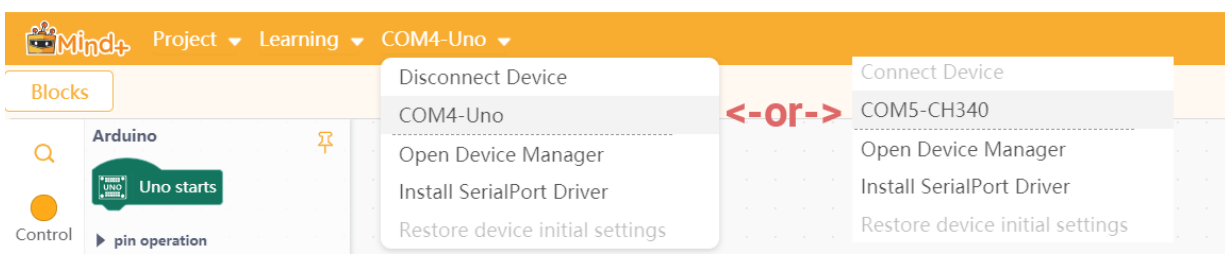
Korak 2: Otvorite proširenja i odaberite **Board** (ploču) - Arduino UNO



Korak 3: Prijeđite na karticu **Sensor** i odaberite senzor - HUSKYLENS AI kamera



Korak 4: Nakon odabira kliknite na **<- Back** (Natrag) i spremni ste za korištenje Arduino i Senzor blokova. Testirajmo ga da vidimo radi li. Prije toga morate spojiti uređaj. Uključite svoj Arduino UNO putem USB kabela i odaberite COM X-Uno (ili CH340) ovisno o proizvođaču Arduinoa.





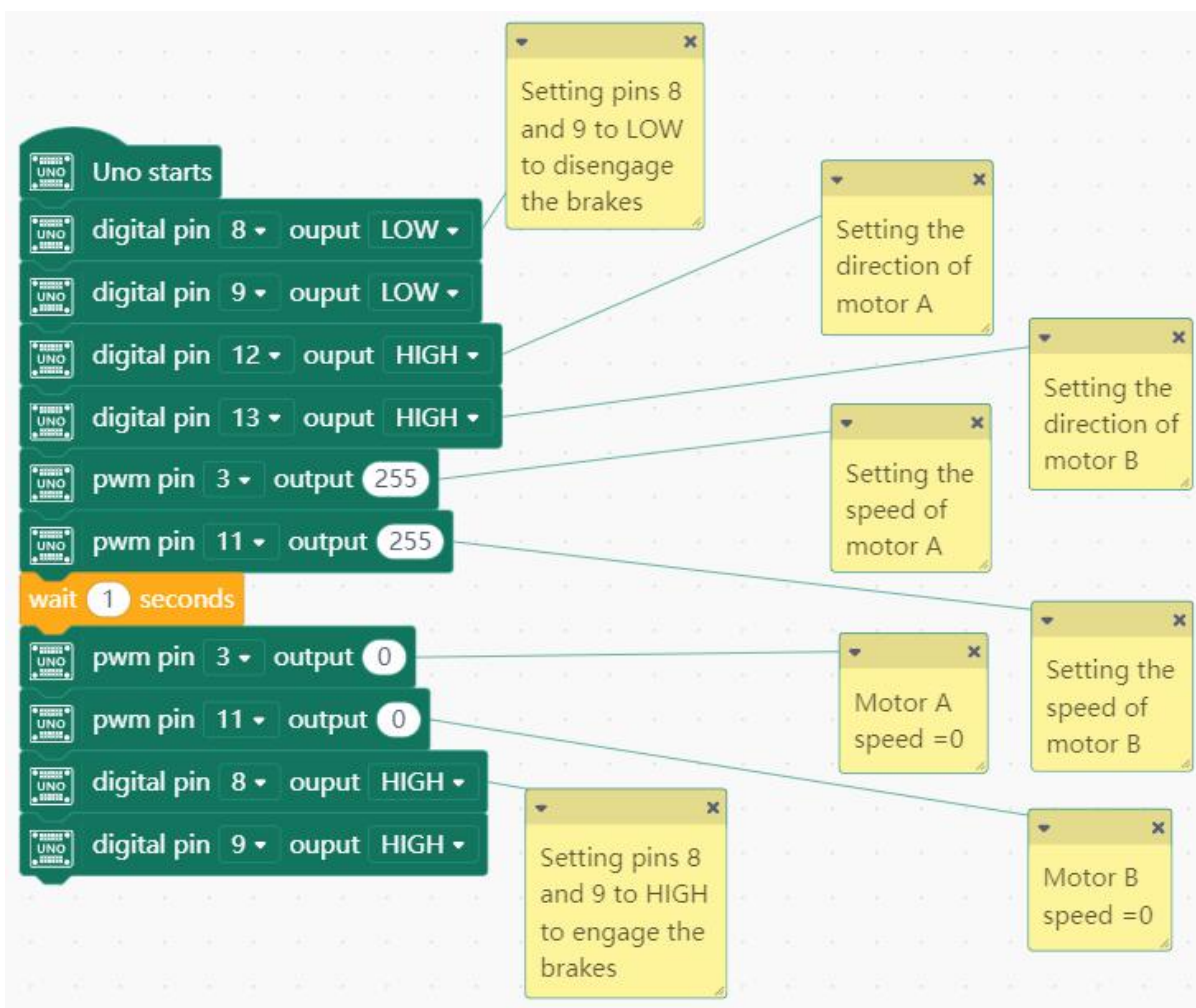
Osnovne kretnje

Sjećate li se ove tablice iz scenarija uvođenja hardvera?

Funkcija	Motor A	Motor B
Smjer	Digitalni pin 12	Digitalni pin 13
Brzina (PWM)	Digitalni pin 3	Digitalni pin 11
Kočnica	Digitalni pin 9	Digitalni pin 8
Trenutno istraživanje	Analogni pin 0	Analogni pin 1

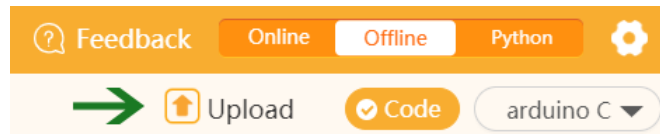
Dva motora (A i B) su lijevi i desni motori. Digitalni pinovi 12 i 13 koriste se za promjenu smjera (vrijednosti HIGH - jedan smjer i LOW - suprotni smjer), a PWM pinovi 3 i 11 koriste se za podešavanje brzine (vrijednosti 0-255). Pinovi 9 i 8 uključuju/isključuju kočenje (HIGH - kočnice uključene, LOW - kočnice isključene).

Ispod možete vidjeti primjer koda s komentarima na desnoj strani koji će vam pomoći da shvatite kako to funkcionira.





Korak 1a: Pritisnite **Upload** za prijenos ovog koda na Arduino UNO.

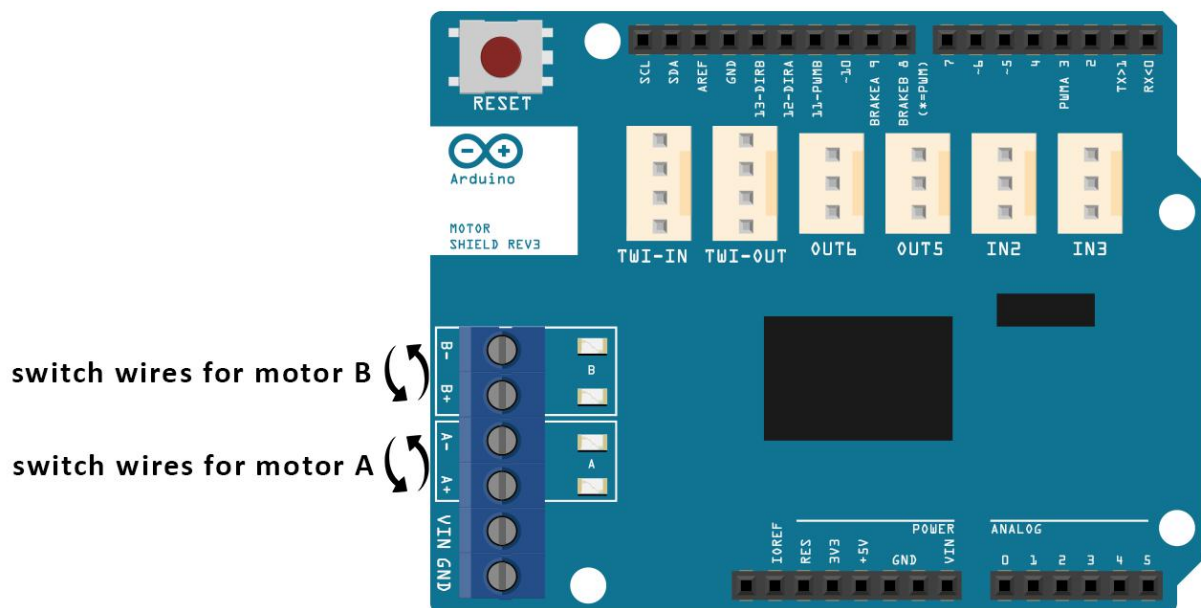


Korak 2a: Budite oprezni, robot će se početi kretati odmah nakon što je prijenos završen!

Robot bi trebao krenuti naprijed na sekundu i stati.

Otklanjanje poteškoća – ako robot :

- vozi unatrag - zamijenite crvene i crne žice u priključcima za oba motora.
- rotira u smjeru kazaljke na satu - zamijenite crvene i crne žice u priključcima za B motor.
- okreće se u smjeru suprotnom od kazaljke na satu - zamijenite crvene i crne žice u priključcima za A motor.

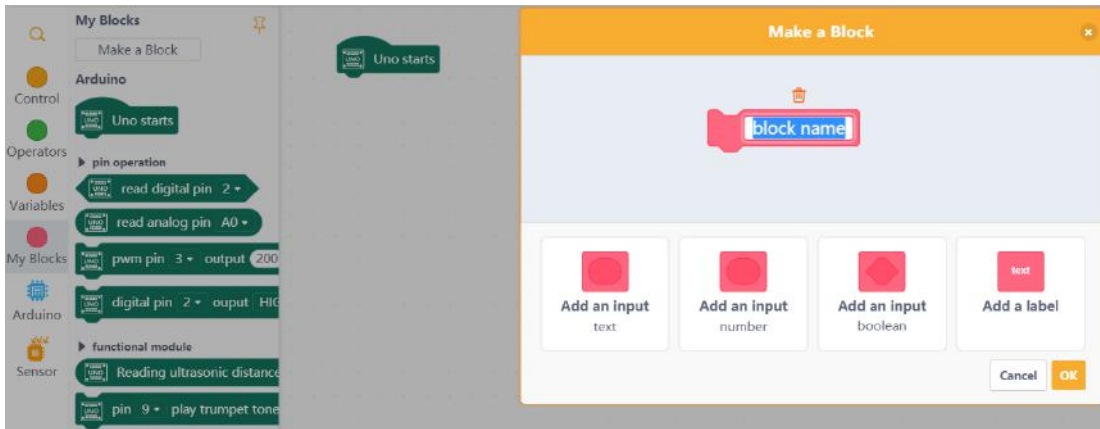


Sada biste trebali imati svog robota koji vozi naprijed s HIGH stanjem na pinovima 12 i 13. Postoji i lakši način programiranja kretanja - koristit ćemo naše prilagođene blokove umjesto ponavljanja cijele grupe blokova za upravljanje pinovima.

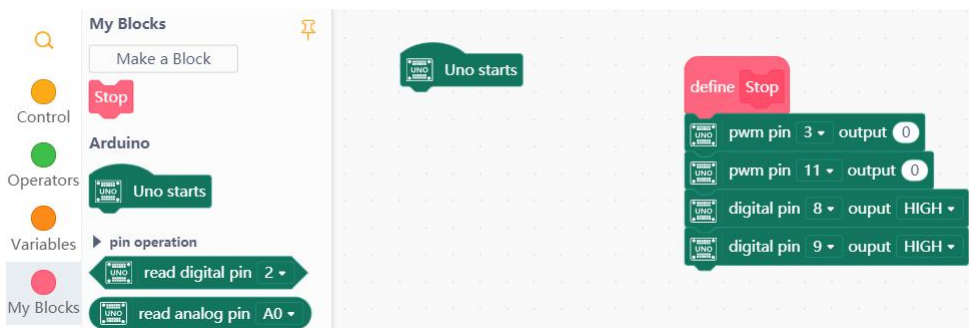
Korak 1b: Kliknite na grupu **My Blocks** (Moji blokovi, crvena boja).

Korak 2b: Kliknite gumb **Make a Block** (Napravi blok) i promijenite "block name" u "Stop".

Korak 3b: Kliknite U redu.

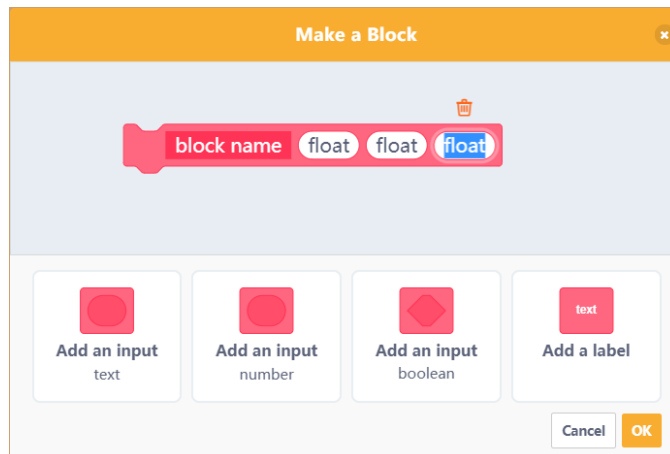


Stvorili ste svoj prvi blok, ali trenutno nije povezan ni sa čime, pa ga prvo moramo definirati.



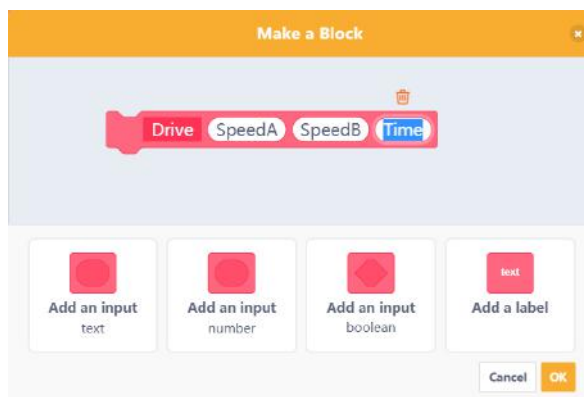
Zaustavni blok sadrži 4 bloka. Prva dva bloka postavljaju brzinu oba motora na 0, a posljednja dva su blokovi koji s visokim (HIGH) stanjem koč motore. Ako želite koristiti da se motori zaustave inercijom bez kočenja (*coasting*), samo postavite stanje digitalnih pinova 8 i 9 na nisko (LOW).

Sada je vrijeme za izradu ulaznog (*input*) bloka s 3 numerička parametra koji će sadržavati brzinu i motora i trajanje. Kliknite ponovno na gumb Napravi blok, a zatim kliknite na **Add an input - number** - 3 puta i trebali biste dobiti ovo:

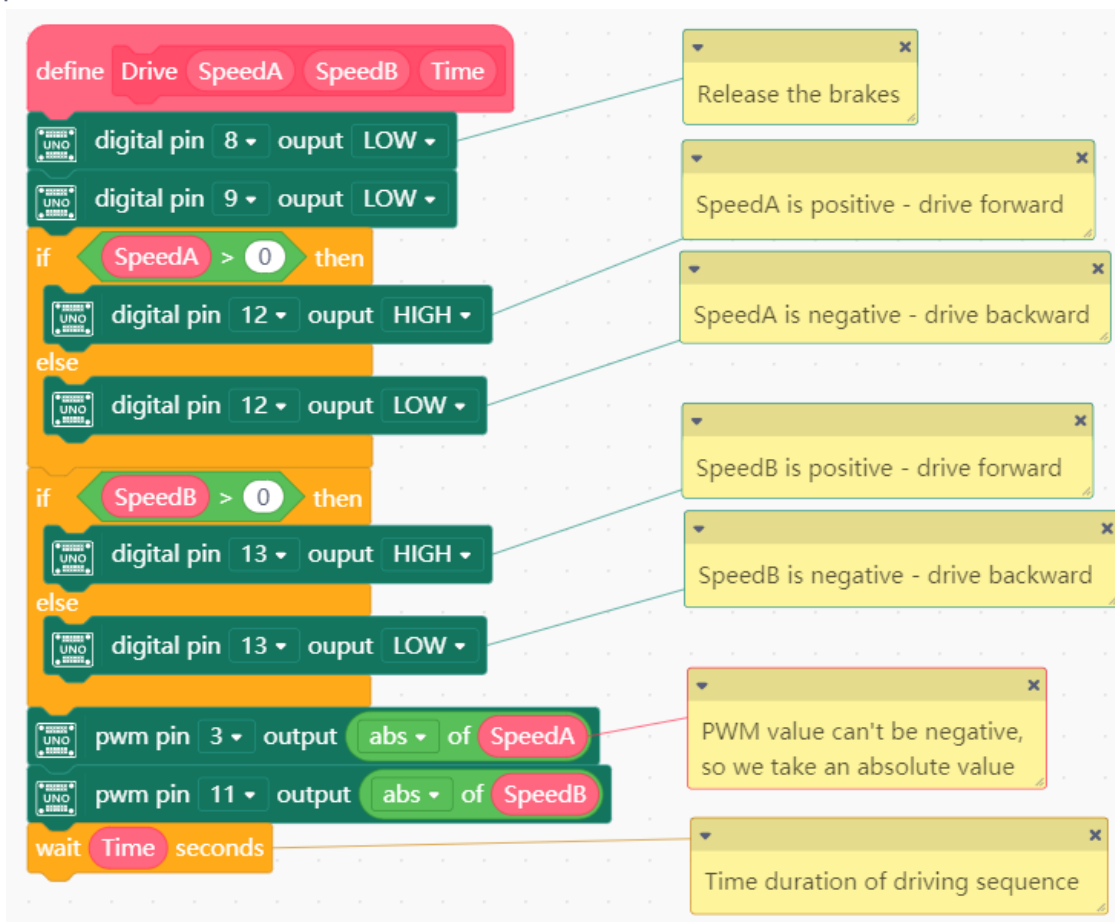




Promijenite naziv bloka u **Drive** (vozi), prvi float promijenite u **SpeedA**, drugi float u **SpeedB**, a treći float u **Time**, a zatim kliknite na gumb **OK**.

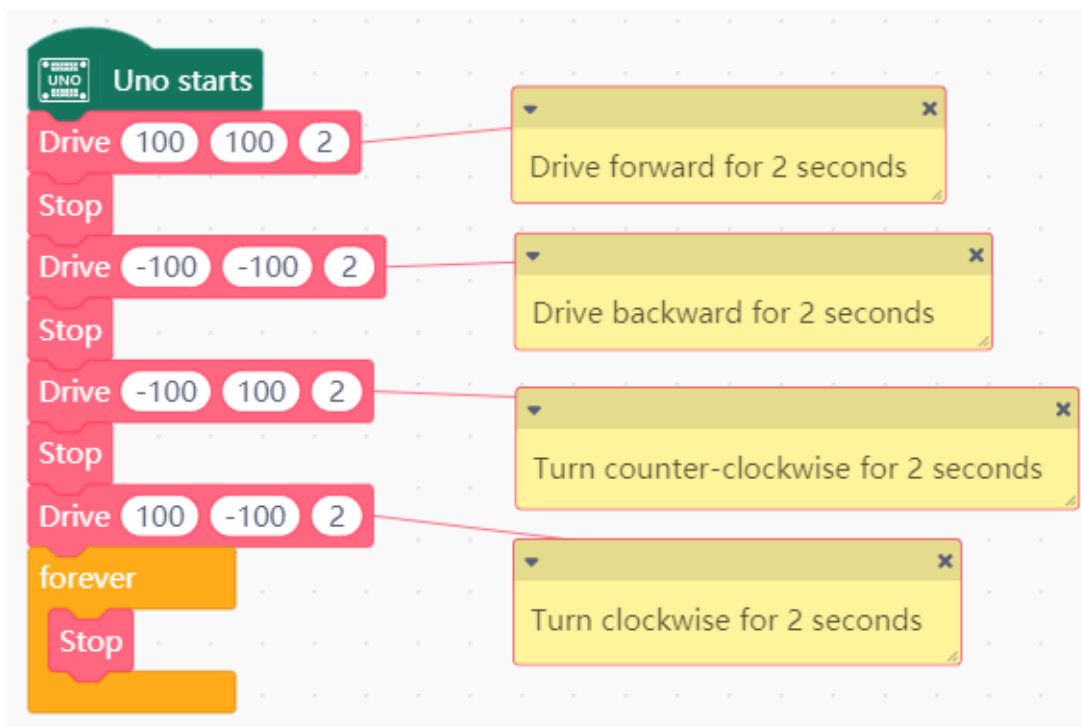


Osnovna ideja je dobiti SpeedA i SpeedB vrijednosti (prihvatljiv raspon: -255 do 255), zatim provjeriti je li koji od brojeva negativan (ili oba) i ako jest - preokrenuti smjer postavljanjem vrijednosti za odgovarajući pin. Za pokretanje motora morate koristiti apsolutnu vrijednost na PWM pinu.





Dakle, pokušajmo pokrenuti našeg robota sljedećim blokovima.



Posljednji stop blok je u beskoačnoj petlji čime se završava kretanje robota.

Promijenite vrijednosti u bloku **Drive**, a zatim prenesite svoj program u Arduino UNO i promatrajte koliko brzo se kreće vaš robot.

Sada ste spremni koristiti svog ARTIEbota u složenijim primjerima, uključujući i one sa HuskyLens kamerom.

ZAKLJUČAK

Budimo iskreni, roboti su cool. Oni će možda upravljati svijetom jednog dana, i nadamo se da će u tom trenutku imati milosti prema svojim slabašnim tvorcima (inženjerima robotike) i pomoći nam izgraditi raj u svemiru. To je bila šala, naravno, ali to zna.

Promisli!



Što sada znam?



Što sam naučio/la?



Što još želim znati?



Programiranje robota - praćenje objekta

Naučili smo kako programirati kretanje robota u našoj prethodnoj lekciji.

Pomaknimo naše robote prema određenom objektu. Prvo, moramo detektirati taj objekt, a zatim ga pratiti. Praćenje pokretnog objekta zahtijeva vizualnu tehnologiju praćenja objekata, kao i ručno upravljanje.

Kako pripremiti robota za praćenje objekata?



Sufinancirano sredstvima
programa Europske unije
Erasmus+



<http://erasmus-artie.eu>



Praćenje objekata važan je zadatak u računalnom vidu. Odnosi se na proces kontinuiranog pronalaženja određenih objekata u video sekvenci. Slika se dobiva jednom kamerom, a podaci iz kamere se prenose u mikrokontroler. Nakon analize i obrade izračunava se relativni položaj pokretnog objekta. U isto vrijeme, kontrolira se kretanje robota koji nosi kameru kako bi pratio objekt u stvarnom vremenu.

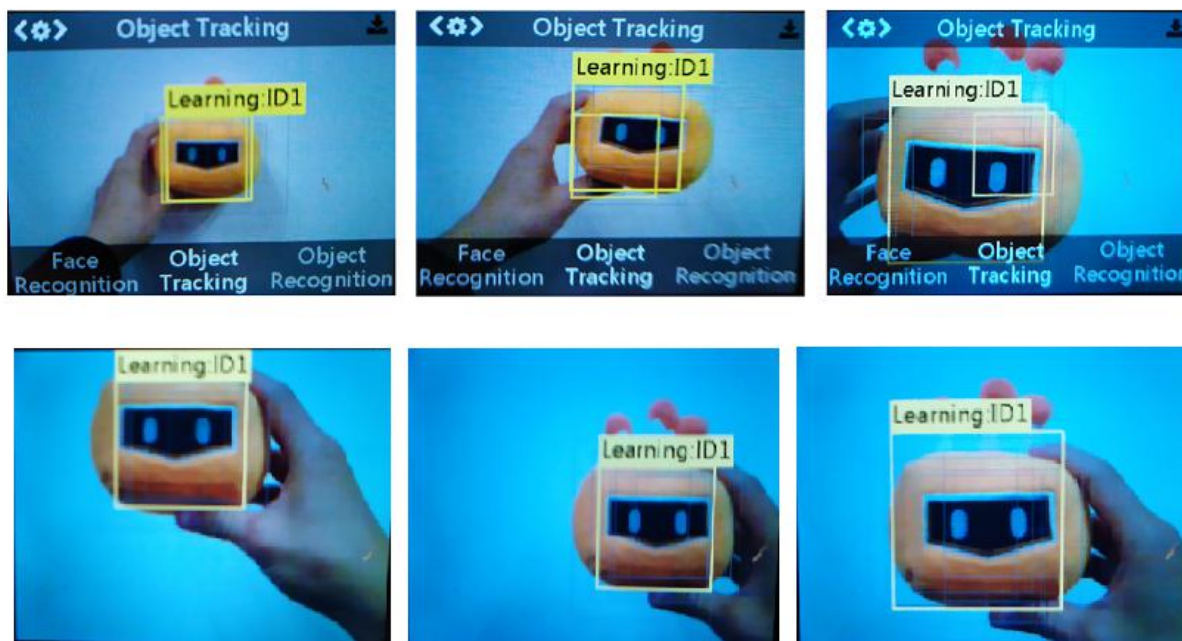
Funkcioniranje sustava za praćenje objekata uglavnom se dijeli na 4 koraka:

- prepoznavanje objekata
- praćenje objekata
- analiza premještanja objekta
- upravljanje robotom (ili bilo kojim drugim sustavom) kamerom

Prepoznavanje objekata - učenje

Povežite micro:bit ili Arduino UNO s HuskyLens kamerom i sa prijenosnim ili stolnim računalom. Usmjerite HuskyLens na ciljni objekt, prilagođavajući udaljenost sve dok se objekt na zaslonu kamere ne nađe u narančastom graničnom okviru. Također je prihvatljivo da je samo dio objekta unutar okvira, ali taj dio mora imati neke specifičnosti u izgledu.

Zatim dugo pritisnite gumb za učenje (**Learn Button**) kako biste naučili objekt iz različitih kutova i udaljenosti. Tijekom procesa učenja na zaslonu će se prikazati narančasta kutija s riječima "Učenje: ID1".



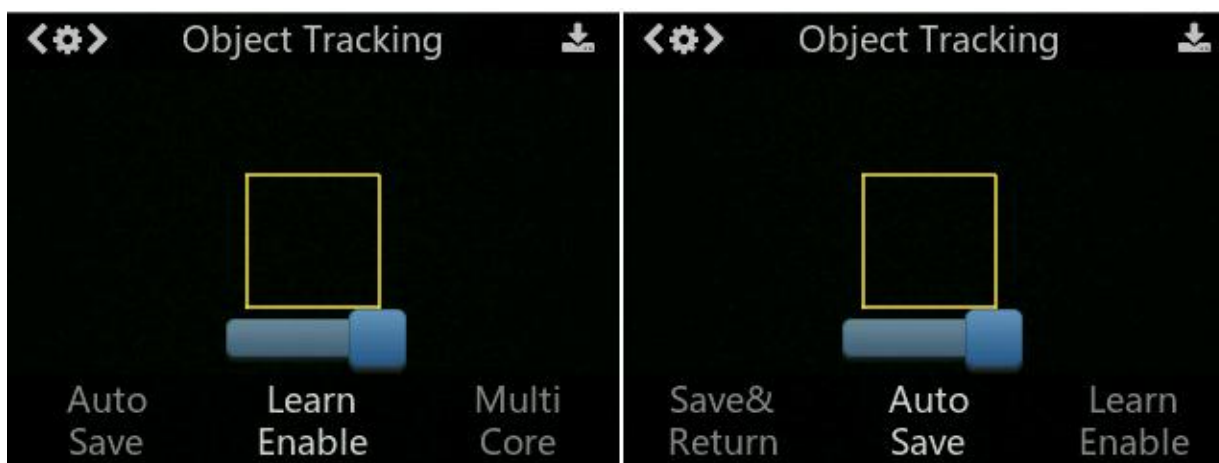


Kada primijetite da HuskyLens može pratiti objekt pod različitim kutovima i udaljenostima, otpustite gumb za učenje kako biste završili taj proces. Ako na zaslonu nestane početni narančasti okvir, to znači da je HuskyLens već naučio objekt. Ako želite pratiti drugi objekt - odaberite **Forget Learned Object** (Zaboravi naučeni objekt) i ponovno pokrenite proces učenja.

Unutar funkcije praćenja objekta, HuskyLens može nastaviti učiti, to jest, sve dok kamera vidi naučeni objekt, nastaviti će učiti trenutno stanje objekta, što dobro kod učenja na objektima koji se kreću. Za izbor tog način rada dugo pritisnite funkcijsku tipku (prema dolje) da biste ušli u podizbornik gdje možete promijeniti postavku parametara za funkciju praćenja objekta.



Potisnite funkcijsku tipku udesno da biste došli do parametra **Learn Enable**, zatim kratko pritisnite funkcijsku tipku za njen izbor, a zatim ju potisnite udesno da biste uključili Learn Enable, odnosno klizač na traci je u desnom položaju. Zatim opet kratko pritisnite funkcijsku tipku da biste potvrdili ovaj parametar. Izadite iz podizbornika kratkim pritiskom na stavci **Save&Return** kao bi ste pohranili izmjene.



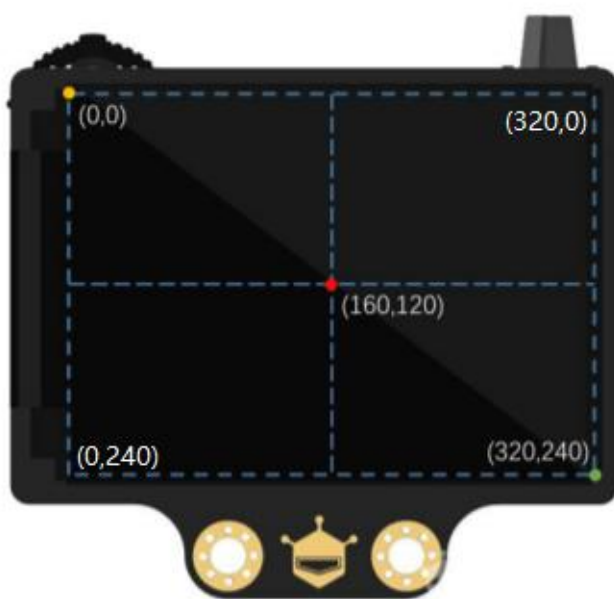


Prilikom ponovnog pokretanja HuskyLensa, posljednji naučeni objekt nije spremljen prema zadanim postavkama i možete na prethodno objašnjen način uključiti prekidač za automatsko spremanje modela (**Auto Save**).

Način rada: isto kao gore, nakon unosa postavke parametra prebacite klizač u stavci **Auto Save** u desni položaj. Na taj način objekt trebate naučiti samo jednom. Ponovno pokretanjem kamere spremit će se objekt koji ste zadnji put naučili.

Praćenje objekata

Razlučivost zaslona senzora HuskyLens kamere je 320*240, kao što je prikazano na sljedećoj slici.



Koordinate središta objekta dobivene kroz program također su unutar ovog raspona. Na primjer, ako su dobivene koordinatne vrijednosti (160, 120), objekt koji se prati nalazi se u središtu zaslona. "Koordinate X" i "Y koordinate" odnose se na položaj središnje točke okvira u koordinatnom sustavu zaslona. "Širina objekta" i "Visina objekta" odnose se na veličinu okvira. Pod funkcijom praćenja objekta okvir je kvadratnog oblika, tako da su širina i visina jednake.

Testirajte praćenje objekta - Opcija 1 (Maqueen Plus/HuskyLens)

Otvorite svoj Mind+ i učitajte proširenja za rad s Maqueen Plus i HuskyLens kamerom.

Koristi slijedeći kod:



```
micro:bit starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Object tracking
set motor All stop

forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 frame is on screen from the result? then
      serial output join "X: " HuskyLens get X center of ID 1 frame from the result in string , No-Wrap
      serial output join ", Y: " HuskyLens get Y center of ID 1 frame from the result in string , No-Wrap
      serial output join ", W: " HuskyLens get width of ID 1 frame from the result in string , No-Wrap
      serial output join ", H: " HuskyLens get height of ID 1 frame from the result in string , Wrap
    wait 1 seconds
```



Preskoči na provjeru rezultata

Testirajte praćenje objekta - Opcija 2 (Arduino UNO/HuskyLens)

Otvorite svoj Mind+ i učitajte proširenja za rad s Arduino UNO i HuskyLens kamerom.

Koristi ovaj kôd za Arduino/HuskyLens:

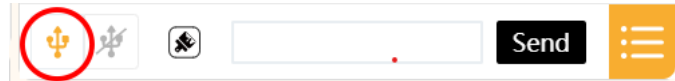
```
Uno starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Object tracking

forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 frame is on screen from the result? then
      serial output join "X: " HuskyLens get X center of the ID 1 No. 1 frame from the result in string , No-Wrap
      serial output join ", Y: " HuskyLens get Y center of the ID 1 No. 1 frame from the result in string , No-Wrap
      serial output join ", W: " HuskyLens get width of the ID 1 No. 1 frame from the result in string , No-Wrap
      serial output join ", H: " HuskyLens get height of the ID 1 No. 1 frame from the result in string , Wrap
    wait 0.5 seconds
```

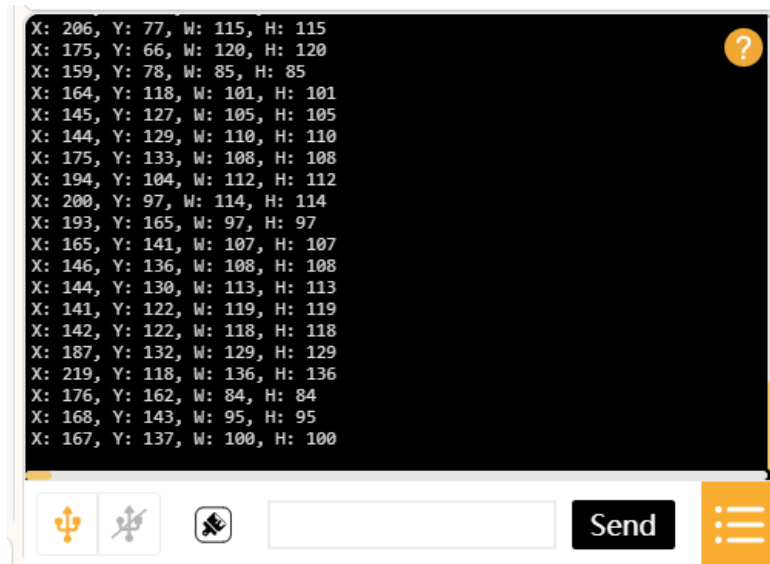


Provjera rezultata na serijskom monitoru (obje opcije)

Serijski monitor otvorite klikom na USB ikonu u donjem desnom dijelu zaslona Mind+.

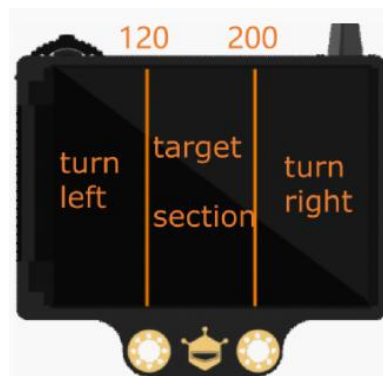


Pokušajte pomaknuti objekt ulijevo i udesno kako biste promatrali numeričku promjenu X središta. Premjestite objekt gore-dolje da biste promatrali numeričku promjenu Y centra. Pomaknite objekt naprijed-natrag da biste promatrali numeričku promjenu širine i visine.



Analiza gibanja objekata

Kao što je prikazano na sljedećoj slici, zaslon je podijeljen u 3 odjeljka prema osi X koordinatnog sustava zaslona kamere, a srednji dio je naš ciljni dio.

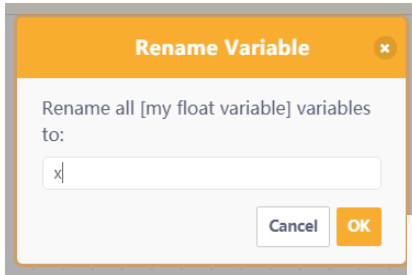




Kada kamera kontinuirano detektira stanje ciljnog objekta na slici, njezin X centar je 120-200, što znači da je cilj u središtu vidnog polja, a robot ne mora prilagoditi svoj položaj; njegov X centar je 0-120, naš robot se mora prilagoditi skretanjem udesno; njegov X centar je 200-320, ARTIEbot treba skrenuti lijevo kako bi se prilagodio.

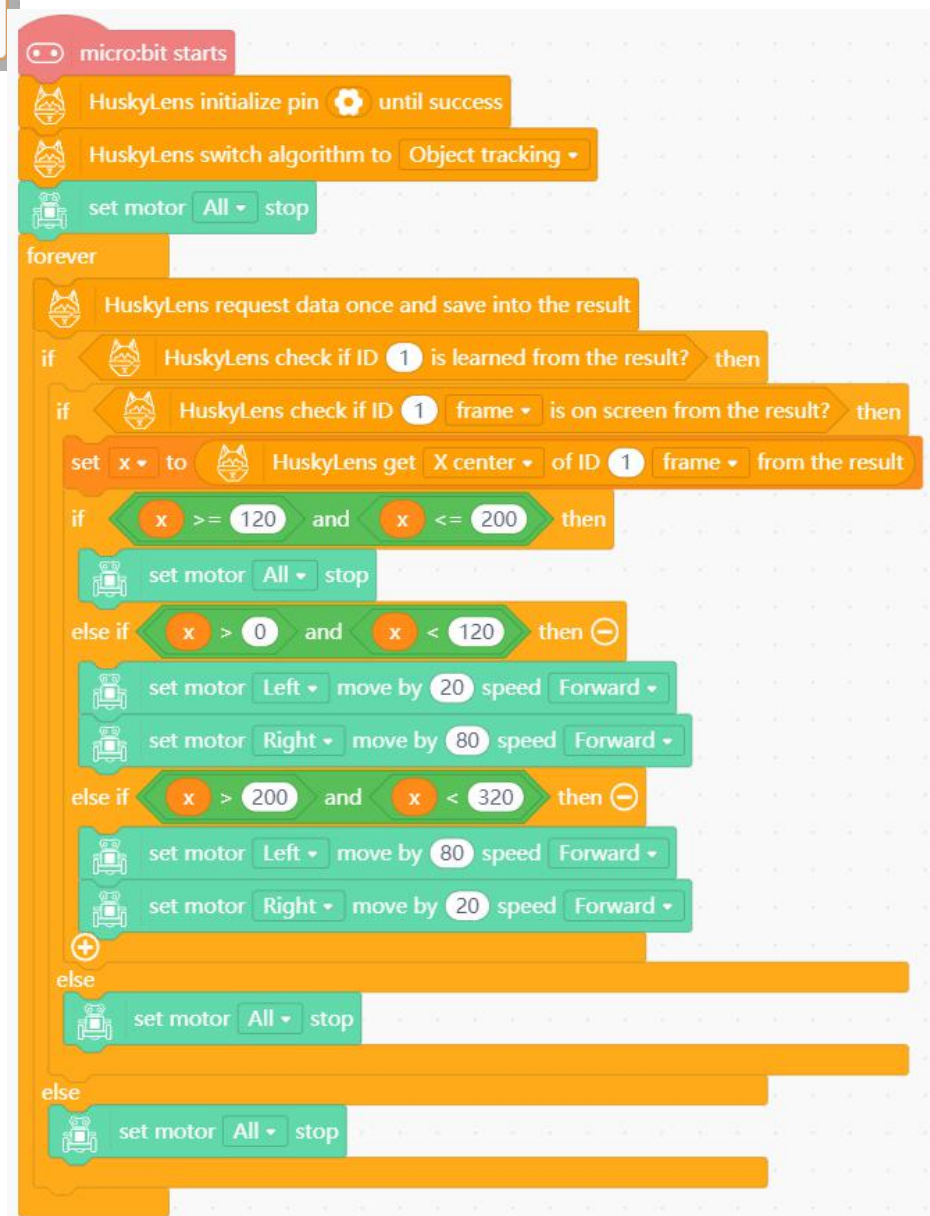
Sada je vrijeme da napišete glavni dio koda kako biste okrenuli robota prema objektu.

Objek opcije - preimenujte **my float variable** u x. Desnom tipkom miša kliknite na varijablu i birajte **Rename numeric variable**.



Opcija 1 - pratite objekt pomoću Maqueen Plus robota

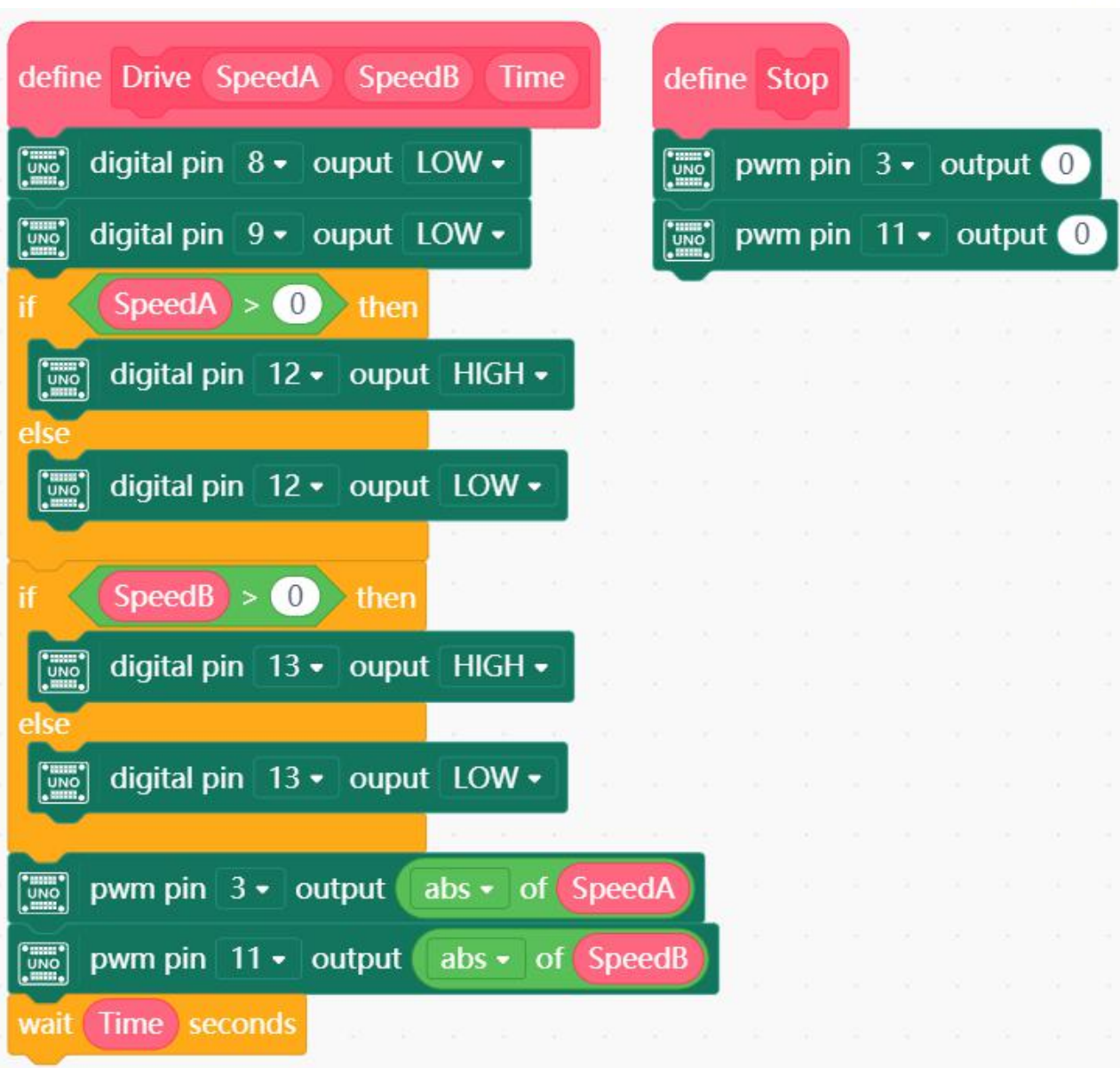
Koristite i konfigurirajte
blokove kao na slici:





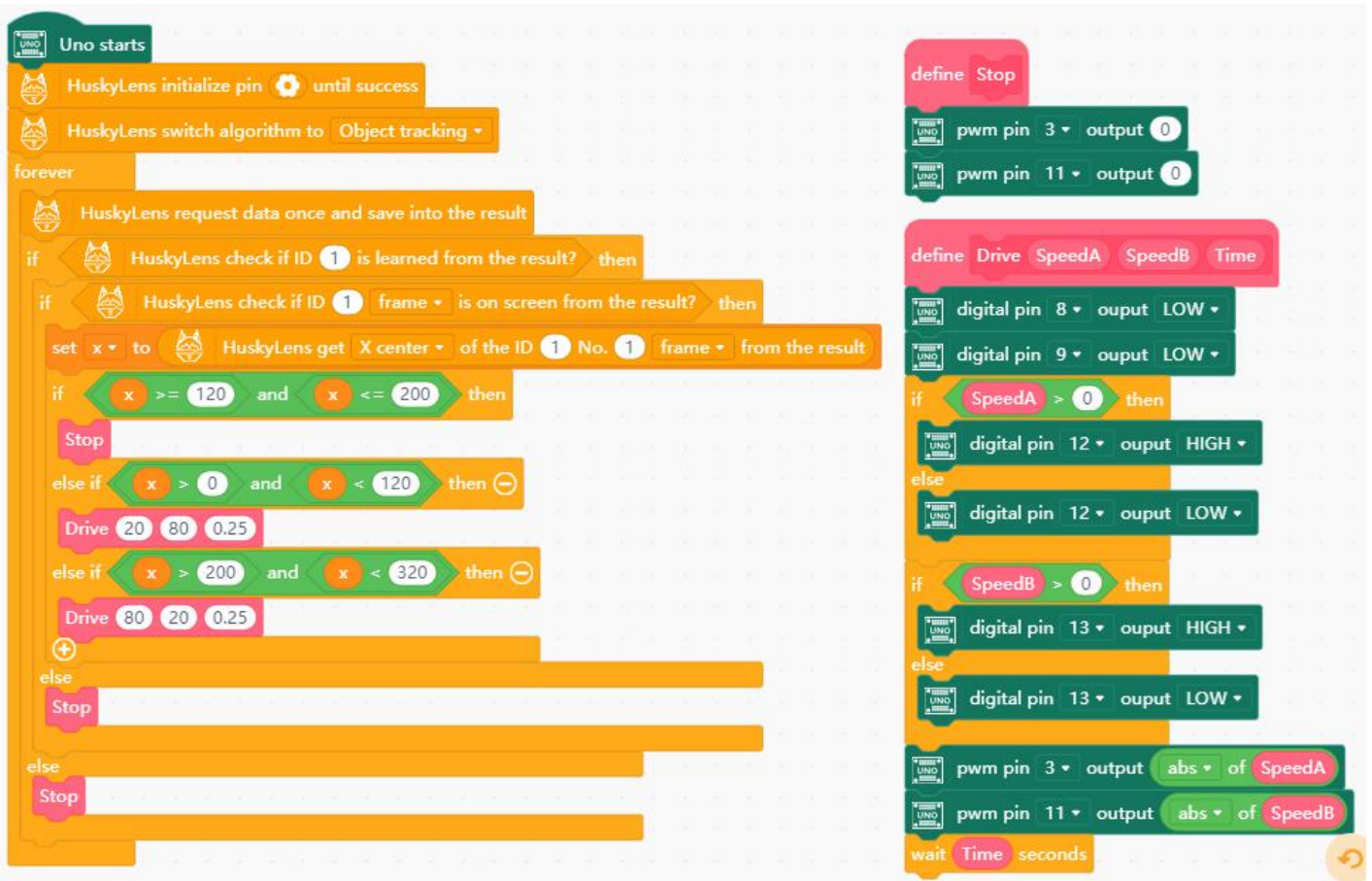
Opcija 2 - pratite objekt pomoću ArtieBota

Prvo definirajte **Drive** blokove i zaustavljanje (**Stop**) kako je opisano u prethodnoj lekciji (Programiranje kretanja robota)





Koristite i konfigurirajte blokove kao na slici ispod:

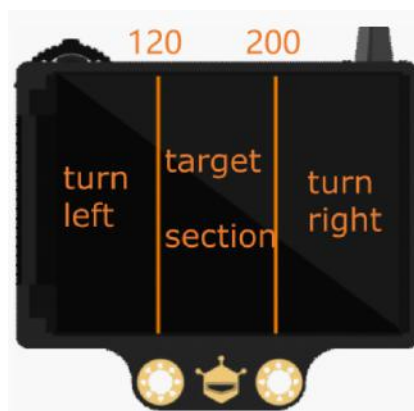


Prenesite program u mikrokontroler svog robota.

Ako je potrebno, prilagodite brzine za MotorA i za MotorB.

Kada je okvir detektiranog objekta u sredini zaslona, robot se zaustavlja.

Kada se okvir nalazi na lijevoj ili desnoj strani zaslona, robot automatski prilagođava svoj položaj kretanjem ulijevo (**turn left**) ili udesno (**turn right**) sve dok se okvir ne nađe u središnjem dijelu zaslona (**target section**).





Praćenje objekta

Roboti se okreću prema objektu, ali ga još uvijek ne prate. Da bismo to postigli, morat ćemo otkriti veličinu objekta kako bismo saznali je li velik (nalazi se blizu kamere) ili mali (udaljen je od kamere).

Napravite novu varijablu (numeričku) i nazovite je **h**. Ona će sadržavati visinu objekta koji pratimo. Ako je visina objekta između 60 i 100, robot će zadržati trenutni položaj. Ako je **h** manji od 60, objekt je daleko i moramo pokrenuti robota naprijed prema objektu. Ako je **h** veći od 100, objekt je preblizu i robot se treba pokrenuti unatrag i odmaknuti od objekta.

Opcija 1 - Kod za Maqueen Plus





Opcija 2 - Kod za Arduino (ArtieBot):

```

Uno starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Object tracking

forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 frame is on screen from the result? then
      set x to HuskyLens get X center of the ID 1 No. 1 frame from the result
      set h to HuskyLens get height of the ID 1 No. 1 frame from the result
      if x >= 120 and x <= 200 and h >= 60 and h <= 100 then
        Stop
      else if x > 0 and x < 120 then
        Drive 20 80 0.25
      else if x > 200 and x < 320 then
        Drive 80 20 0.25
      else if h > 0 and h < 60 then
        Drive 80 80 0.25
      else if h > 100 and h < 240 then
        Drive -80 -80 0.25
      else
        Stop
    else
      Stop
  else
    Stop

```

define Drive SpeedA SpeedB Time

```

digital pin 8 output LOW
digital pin 9 output LOW
if SpeedA > 0 then
  digital pin 12 output HIGH
else
  digital pin 12 output LOW
if SpeedB > 0 then
  digital pin 13 output HIGH
else
  digital pin 13 output LOW
pwm pin 3 output abs of SpeedA
pwm pin 11 output abs of SpeedB
wait Time seconds

```

define Stop

```

pwm pin 3 output 0
pwm pin 11 output 0

```

Obje opcije - Provjerite kako to funkcionira

Prenesite program na micro:bit/Arduino UNO kako biste provjerili kako funkcionira.

Napravite korekcije kako bi kretanje bilo glatko podešavanjem brzine i vremena vožnje motora.

Nakon što HuskyLens završi s učenjem objekta, roboti će automatski slijediti objekt i kretati se naprijed, natrag, lijevo i desno, držeći okvir objekta u središtu zaslona i na odgovarajućoj udaljenosti.

Kada se robot koristi kao robot za praćenje, može se programirati za lociranje bilo kojeg objekta s HuskyLens kamerom. To znači da ovim programom možete pretvoriti vašeg robota u robota koji prati osobe.



Praćenje objekata zadatak je dobivanja skupa vrijednosti pri detekciji objekata, stvaranja jedinstvenog identifikatora za svaki od detektiranih objekata, a zatim praćenja svakog od objekata dok se kreću te označavanjem okvirom u videozapisu, poštujući dodijeljene identifikatore. Najsuvremenije metode uključuju spajanje podataka s RGB i drugih kamera kako bi se dobilo pouzdanije praćenje objekata.

Sada razumijemo osnovne principe praćenja objekata i naučili smo funkciju praćenja objekata HuskyLens.

Također znamo kako koristiti HuskyLens kako bi naš robot mogao pratiti objekt.

ZAKLJUČAK

Za uspješno praćenje objekta, prvo moramo detektirati taj objekt, a zatim ga pratiti.

Praćenje pokretnog objekta zahtijeva vizualnu tehnologiju praćenja objekata, kao i ručno upravljanje.

Promisli!



Što sada znam?



Što sam naučio/la?



Što još želim znati?

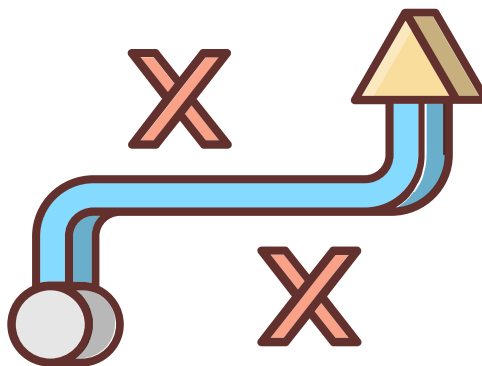




Programiranje robota - praćenje linije

Naučili smo kako programirati robote da bi mogli pratiti objekte u prethodnoj lekciji.
Pobrinite se da vaši učenici razumiju sve iz prethodne lekcije i da su spremni za sljedeći korak.
Naučimo sada kako naći i slijediti liniju.

Kako pripremiti robota za praćenje linije?





Praćenje linije odnosi se na proces kretanja objekta duž određene rute. Potpuno funkcionalan sustav za praćenje linija koristi mobilnog robota s kamerom, infracrvenim termalnim uređajem ili drugom opremom za detekciju, koji pribavlja skup različitih informacija kao što su podaci dobiveni od računalnog vida, informacije o elektromagnetskom polju, GPS i GIS informacije iz navigacijskog sustava za autonomno kretanje i praćenje te na kraju koristi ugrađeno računalo kao softversku i hardversku razvojnu platformu za upravljanje tim sustavom.

Linijski senzori za praćenje

Tip Stavke za usporedbu	Senzor praćenja infracrvene linije	Vizualni senzor
Cijena	Niska	Visoka
Domet senzora	Senzor ima mali domet; mora biti blizu podloge	Domet senzora je veći, a očitavanja prilikom kretanja mogu se unaprijed prilagoditi promjenama.
Prilagodba okolini	Kada se promijeni okruženje, potrebno je prilagoditi osjetljivost senzora, a postupak podešavanja je kompliciran.	Kada se promijeni okruženje, potrebno je samo „naučiti“ senzor na novu liniju, a ta operacija je jednostavna.
Prilagodba mape za kretanje	Općenito, prikladno samo za jednostavne mape s jasno razlučivom pozadinom, crno-bijelim linijama ili punim linijama	Pogodno za mape s jasnim pozadinskim linijama, raznobojnim linijama, punim linijama, točkastim linijama i drugim složenijim situacijama.

Algoritam praćenja linije HuskyLens

Huky Lens funkcija praćenja linije temelji se na Pixyju, projektu otvorenog koda Sveučilišta Carnegie Mellon. Pixy algoritam može prepoznati boju na slikama. Njegova osnovna ideja je iskoristiti algoritme za rad s bojama za uklanjanje pozadine koja nam nije potrebna i izdvojiti informacije iz prednjeg plana (poput linija).



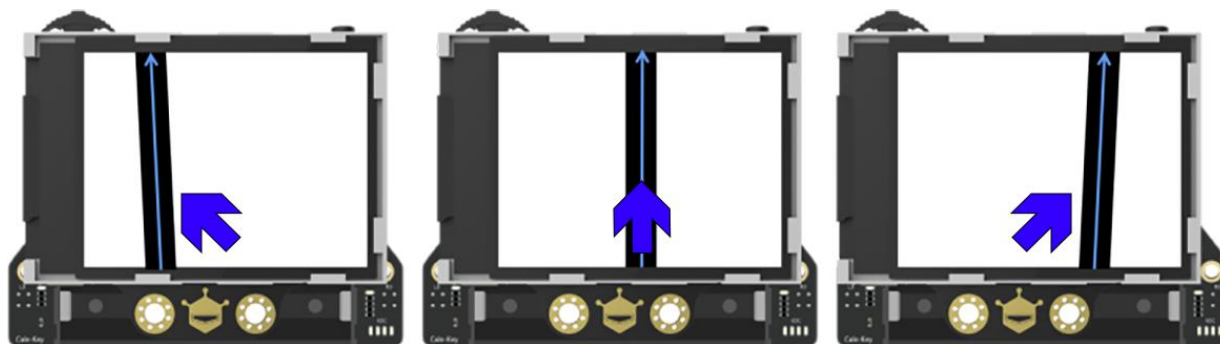
Kako robot može slijediti crne linije na bijeloj podlozi?

Zapravo, samo trebamo znati relativni položaj našeg robota prema crnoj liniji.



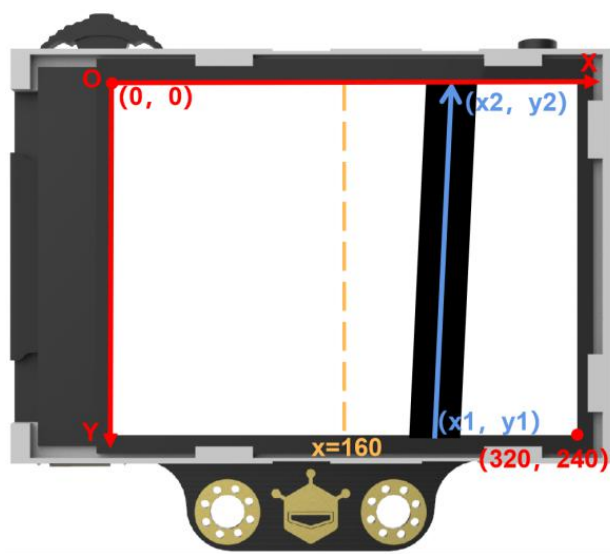
U osnovi, imamo sljedeće tri situacije:

1. Kada je robot s desne strane crne linije, treba skrenuti ulijevo
2. Kada je ARTIEbot u sredini, poravnat s crnom linijom, treba ići ravno
3. Kada je ARTIEbot na lijevoj strani crne linije, treba skrenuti udesno



Implementacija

Razlučivost zaslona HuskyLens je 320×240 . 0-ta točka u gornjem lijevom kutu zaslona je ishodište koordinatnog sustava zaslona (0, 0), vodoravno slijeva nadesno je pozitivni smjer osi X, a vertikalno odozgo prema dolje je pozitivan smjer osi Y, tako da su koordinate u donjem desnom kutu zaslona (320, 240). Točkasta narančasta linija na slici je središnja os zaslona, a x vrijednost ove linije je 160. Crna linija na zaslonu ispod je linija koju je "vidjela" HuskyLens kamera. Plava strelica je smjer linije koji je izračunala HuskyLens kamera. Koordinate početne točke plave strelice su (x_1, y_1) , a koordinate završne točke su (x_2, y_2) .





Da bismo to pojednostavili - trebamo znati samo početnu točku (x1) plave strelice u odnosu na središnju os ($x = 160$) za implementaciju praćenja linije.

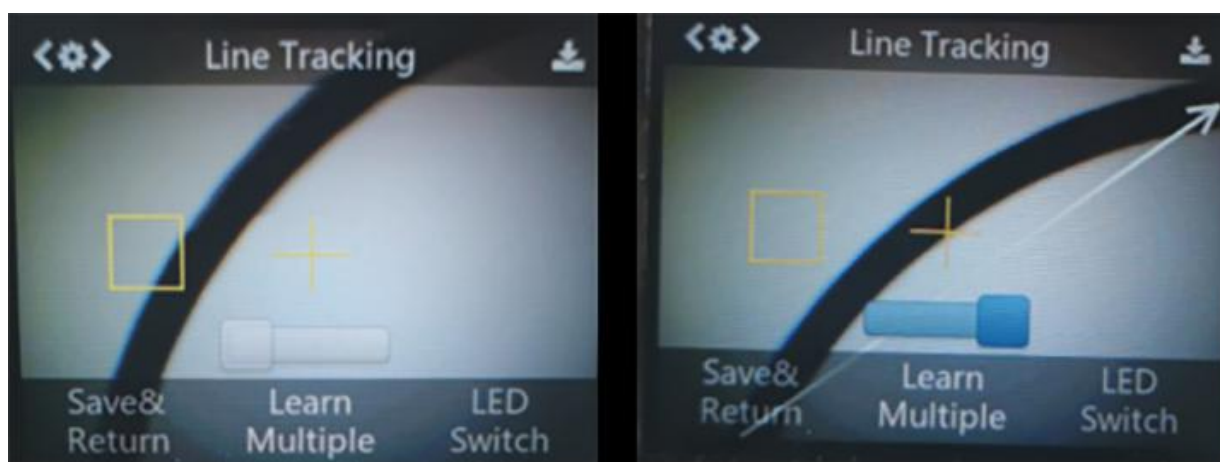
Ova funkcija može pratiti linije određenih boja i predvidjeti putanju. Zadana postavka je praćenje linija samo jedne boje, a ovaj će projekt koristiti **praćenje linije u jednoj boji**.

Postavke kamere

Korak 1: Birajte funkcijsku tipku lijevo ili desno dok se na vrhu zaslona ne prikaže riječ **Line Tracking** (Praćenje crte).

Korak 2: Dugo pritisnite funkcijsku tipku prema dolje da biste pristupili podizborniku s postavkama funkcije za praćenje linije.

Korak 3: Pritišćite funkcijsku tipku desno ili lijevo dok ne dođete do stavke **Learn Multiple**, zatim kratko pritisnite funkcijsku tipku i pritisnite ju ulijevo da biste isključili prekidač **Learn Multiple** (Nauči više), odnosno klizač na traci treba biti u lijevom položaju. Zatim kratko pritisnite funkcijsku tipku da biste izašli iz ove stavke.

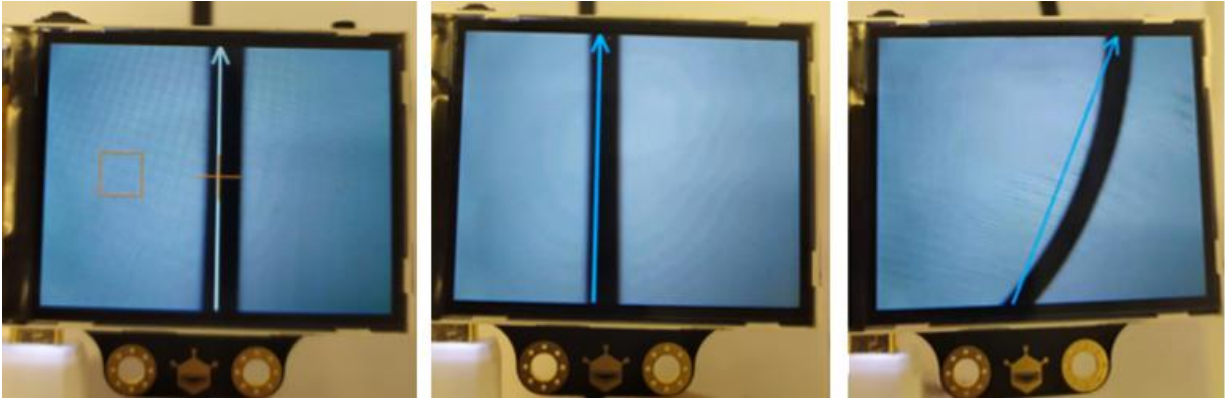


Korak 4: LED diodu možete uključiti kako bi osvijetlila prostor ispred kamere u postavci **LED Switch**. To je vrlo korisno u tamnom okruženju. Na prethodno prikazani način uključite LED prekidač.

Korak 5: Potiskujte funkcijsku tipku ulijevo dok ne odaberete **Save&Return** (Spremi i vrati se) i zatim kratko pritisnite funkcijsku tipku prema dolje da biste spremili parametre i vratili se u glavni izbornik.

Učenje i praćenje

Učenje crte: Usmjerite simbol + na crtu, a zatim usmjerite narančasti okvir na područje pozadine (izvan crte). Preporučuje se da na zaslonu nema više od jedne linije. Pokušajte zadržati HuskyLens paralelno s ciljanom linijom; HuskyLens će automatski otkriti liniju i pojaviti će se bijela strelica. Zatim kratko pritisnite **Learn Button** (gumb za učenje) i bijela strelica se pretvara u plavu strelicu.



- Kada učimo liniju, moramo prilagoditi položaj HuskyLensa kako bi bio paralelan s linijom.
- HuskyLens može naučiti liniju u bilo kojoj boji koja ima očigledan kontrast u odnosu na pozadinu, ali ta linija mora biti jednobojna kako bi proces praćenja linije bio stabilan.
- HuskyLens može naučiti i pratiti više linija s različitih boja, ali svaka od tih linija mora biti jednobojna i vidljivog kontrasta u odnosu na pozadinu. U ovom primjeru koristit ćemo crnu liniju (crna izolacijska traka na bijeloj pozadini koja može biti papir ili bijela MDF ploča).
- Uočljivost linije uvelike ovisi o ambijentalnom svjetlu. Kada slijedite liniju, pokušajte održavati ambijentalno svjetlo što stabilnijim i po potrebi koristite HuskyLens LED.



Otvorite Mind+ i učitajte proširenja.

Kada koristite micro:Maqueen plus robot, provjerite jeste li odabrali pravu verziju (V1 ili V2).

Preimenujte **my float variable** u x. Desnom tipkom miša kliknite na **variable** -> **Rename numeric variable**.

Algoritam

- Pročitajte x1 (ili **X beginning**) iz **Line tracking** funkcije - to je početna točka plave strelice
- Ako je crna linija na lijevoj strani zaslona ($x1 < 150$) - robot bi trebao skrenuti ulijevo.
- Ako je crna linija na desnoj strani zaslona ($x1 > 170$) - robot bi trebao skrenuti udesno.
- Ako je crna linija u sredini zaslona ($150 \leq x1 \leq 170$) - robot bi trebao ići ravno.



Opcija 1 - Program za Maqueen Plus

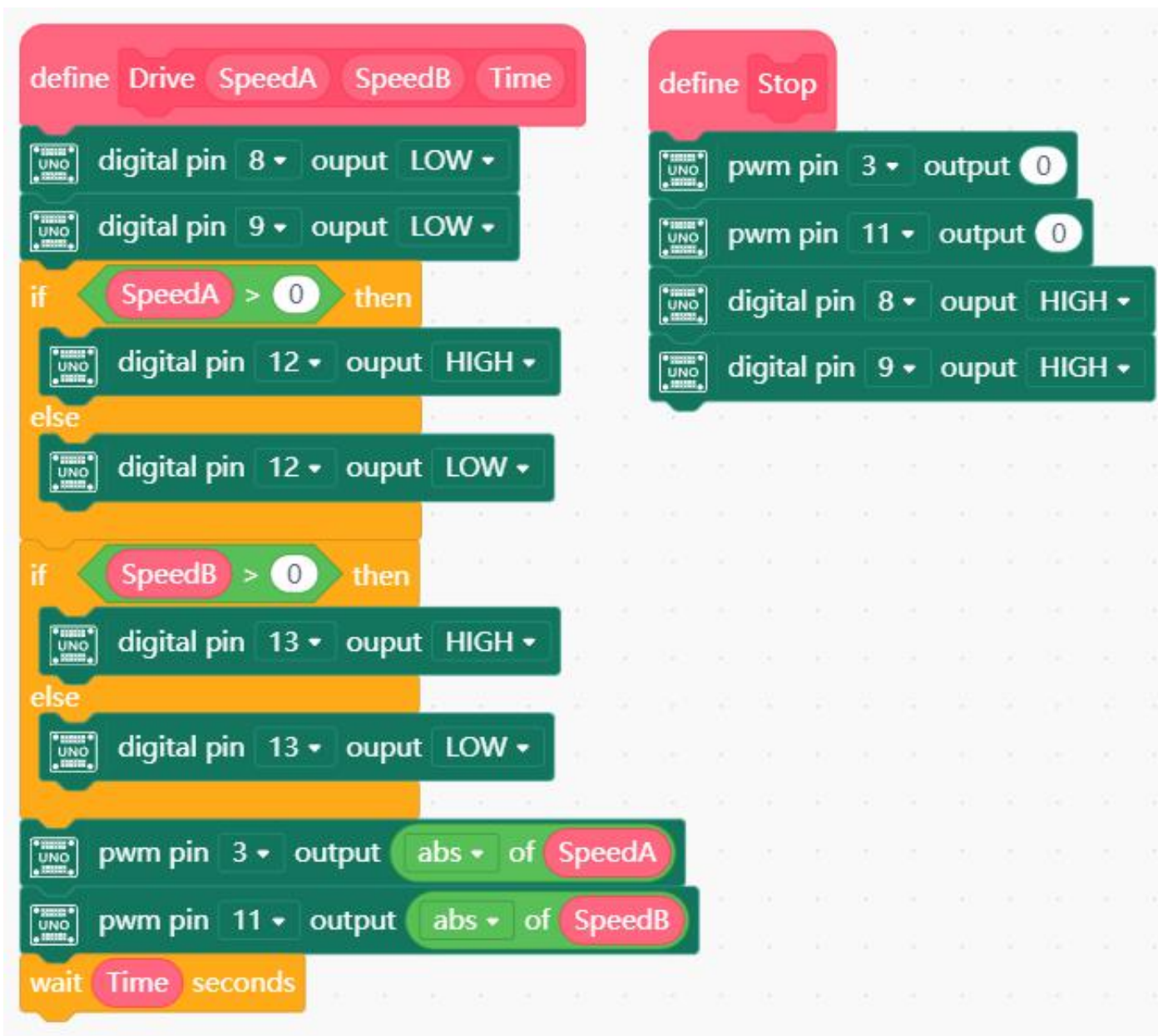
Koristi ovaj kod:

```
micro:bit starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Line tracking
forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 arrow is on screen from the result? then
      set my float variable to HuskyLens get X beginning of ID 1 arrow from the result
      if my float variable >= 150 and my float variable <= 170 then
        set motor All move by 60 speed Forward
      else if my float variable < 150 then -
        set motor Right move by 90 speed Forward
        set motor Left move by 30 speed Forward
      else if my float variable > 170 then -
        set motor Left move by 90 speed Forward
        set motor Right move by 30 speed Forward
      +
    else
      set motor All stop
```




Opcija 2 - Linija koja slijedi s Arduinoom (ArtieBot)

Prvo definirajte blokove **Drive** i **Stop** kako je opisano u lekciji Programiranje kretanja robota. Dobro je staviti često korištene skripte u ruksak (**backpack**) i lako ih dijeliti između projekata.





```
Uno starts
HuskyLens initialize pin until success
HuskyLens switch algorithm to Line tracking
define Stop
  pwm pin 3 output 0
  pwm pin 11 output 0
forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result? then
    if HuskyLens check if ID 1 arrow is on screen from the result? then
      set x to HuskyLens get X beginning of ID 1 arrow from the result
      if x >= 150 and x <= 170 then
        Drive 90 90 0.25
      else if x < 150 then -
        Drive 40 90 0.25
      else if x > 170 then -
        Drive 90 40 0.25
      +
    else
      Stop
```

f

```
define Drive SpeedA SpeedB Time
  digital pin 8 output LOW
  digital pin 9 output LOW
  if SpeedA > 0 then
    digital pin 12 output HIGH
  else
    digital pin 12 output LOW
  if SpeedB > 0 then
    digital pin 13 output HIGH
  else
    digital pin 13 output LOW
  pwm pin 3 output abs of SpeedA
  pwm pin 11 output abs of SpeedB
  wait Time seconds
```



Obje opcije – testiranje algoritma

Pripremite bijelu površinu (papir ili MDF) i pomoću izolacijske trake napravite liniju (vidi primjer u nastavku ili napravite nešto slično).



Prenesite program na svog robota, stavite robota negdje na liniju i pogledajte kako se kreće. Slijedi li liniju?

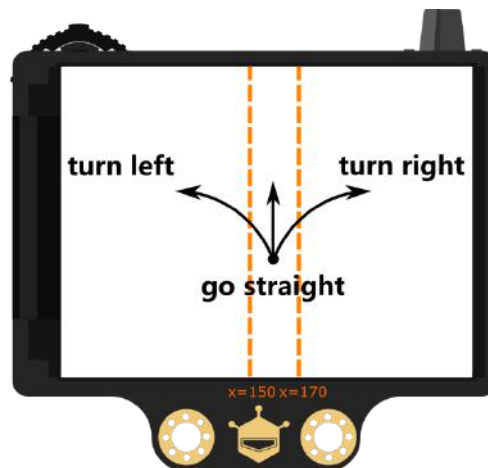
Je li linija izgubljena? Postoji li način da se to riješi? Sljedeći savjeti će vam pomoći:

- Pokušajte promijeniti kut kamere
- Pokušajte koristiti **X endpoint** ID 1 strelice (**arrow**) umjesto **X beginning** točke

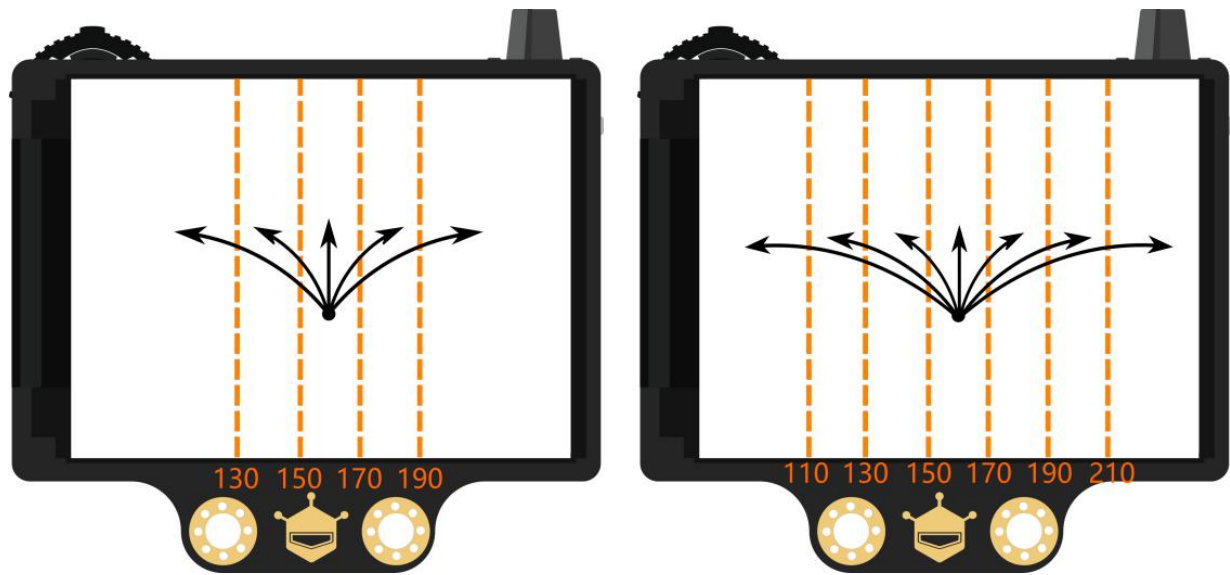
Dodajte kôd za postupanje kod gubitka linije pomoću sljedećeg savjeta:

- Ako je crta prisutna i otkrivena je strelica, prijedite na kod za rukovanje strelicama i spremite sadržaj varijable **x** u novu varijablu koja se zove **lastx**
- Ako se linija izgubi i strelica se ne otkrije, upotrijebite **lastx** umjesto **x** da biste ponovno došli do crte.

I na kraju najvažnije - primijenite korekcije kako biste poboljšali kretanje podešavanjem brzine i vremena vožnje motora. Ovaj algoritam analizira položaj **x**; ima 3 slučaja kao na donjoj slici.



Možete li optimizirati ovaj algoritam za 5 slučajeva ili čak 7 kako bi kretanje bilo glatkije? Pogledajte slike u nastavku da biste dobili ideju kako to učiniti.



Sažetak naučenog

1. Razumijevanje glavnih načela praćenja linije
2. Naučena primjena HuskyLens funkcije za praćenje linije
3. Naučen i optimiziran algoritam koji prati liniju

ZAKLJUČAK

Naučili smo glavna načela praćenja linije primjenom HuskyLens funkcije i optimizirali smo algoritam za praćenje linije.

Promisli!



Što sada znam?



Što sam naučio/la?



Što još želim znati?