

**O‘ZBEKISTON RESPUBLIKASI RAQAMLI TEXNOLOGIYALAR
VAZIRLIGI
MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT AXBOROT
TEXNOLOGIYALARI UNIVERSITETI FARG‘ONA FILIALI**

“SIRTQI BO‘LIM” fakulteti

Kompyuter injiniringi yo‘nalishi

**721-21– guruh talabasi
Sobirjonov Sardorbek ning**

“Ma’lumotlarning Intellektual tahlili ”

**fanidan tayyorlagan
1-Amaliy ishi**

Topshirdi:

Sobirjonov S

Qabul qildi:

AZIMJONOV U

1-Amaliyot

Mavzu: Google teachable machine dasturi yordamida birinchi suniy intellektni yaratish.

Ishdan maqsad: Mavzu yordamida talabalarga sun'iy intellekt haqida umumiy tushuncha berishva kod yozmasdan dastlabki sun'iy intellectual tizim yaratishni amaliy holatda o'rgatish

Nazariy qism:

Teachable Machine - bu Google tomonidan ishlab chiqilgan veb-asoslangan vosita bo'lib, foydalanuvchilarga hech qanday kodlash tajribasisiz o'zlarining mashina o'rganish modellarini o'rgatish imkonini beradi. Tasvirlar yoki videolarni to'plash uchun veb-kameradan foydalanadi va keyin ushbu tasvirlardan mashinani o'rganish modelini o'rgatish uchun foydalanadi. Keyin foydalanuvchi yangi tasvirlar yoki videolarni tasniflash uchun modeldan foydalanishi mumkin. Teachable Machine bilan mashinani o'rganish modelini yaratish jarayoni nisbatan sodda va tushunarli. U quyidagi bosqichlarni o'z ichiga oladi: Ma'lumotlarni to'plash: Tasniflanmoqchi bo'lgan obyektlarning tasvirlari yoki videolarini to'plash uchun veb-kameradan foydalaning.

Modelni o'rgatish: Mashina o'rganish modelini o'rgatish uchun to'plangan rasm yoki videolardan foydalanish. Teachable Machine tasvirlarni belgilash va modelni o'rgatish uchun oddiy interfeysni taqdim etadi. Modelni sinab ko'rish: Yangi rasm yoki videolarni tasniflash va model qanchalik to'g'ri ekanligini ko'rish uchun o'qitilgan modeldan foydalanish.

Modelni sozlash: Agar model yetarlicha aniq bo'lmasa, orqaga qaytib, ko'proq ma'lumot to'plash yoki o'quv parametrlarini sozlash orqali uni aniq sozlashingiz mumkin.

Teachable Machine - bu jismoniy shaxslar, o'qituvchilar va talabalar uchun mashinani o'rganish va kompyuterni ko'rish asoslarini o'rganish uchun ajoyib vosita. U tasvirni tasniflash, ob'ektni aniqlash va imo-ishoralarni aniqlash kabi keng ko'lamli ilovalar uchun oddiy modellarni o'rgatish uchun ishlatilishi mumkin.

Mashinani o'rganish va sun'iy intellekt ilovalar darajasini oshirdi. Ko'pgina tashkilotlar jamiyatga ta'sir o'tkazish uchun sun'iy intellekt ustida ishlamoqda. Mashinani o'rganish sun'iy intellektning asosidir. Ammo hamma ham mashinani o'rganish qanday ishlashini va sun'iy intellektda qo'llanilishi mumkin bo'lgan modellarni qanday yaratishni bilmaydi. Mashinani o'rganish fonida bo'lmagan kodlovchilar yoki koderlar uchun mashinani o'rganish modelini yaratish va uni ilovaga birlashtirish juda oson. Ushbu amaliyot darsda biz bitta qatorni kodlamasdan mashinani o'rganish modelini qurmoqchimiz.

Biz oziq-ovqat mahsulotlarini tasniflash uchun model yaratmoqchimiz. Shunday qilib, biz Kaggle'dan tort, salat, kartoshka va boshqalar kabi turli xil oziq-ovqat mahsulotlaridan iborat oziq-ovqat ma'lumotlar to'plamidan foydalanamiz.

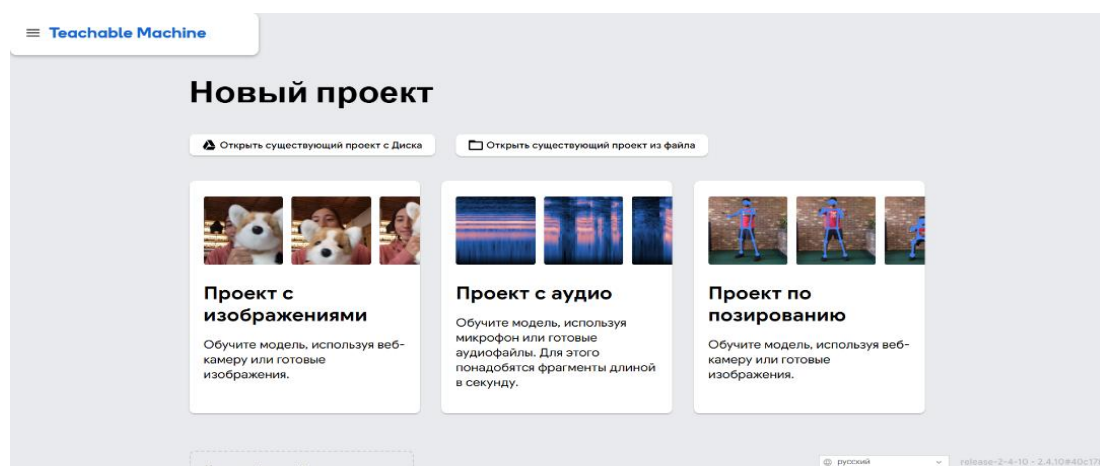
Ma'lumotlar to'plamini <https://www.kaggle.com/cristeaioan/ffml-dataset> manzilidan yuklab olishingiz mumkin.

Ha, biz buni o'rgatiladigan mashina yordamida qilamiz. O'rgatiladigan mashina - bu modellarni osongina va tez yaratadigan veb-ilova. U uchta maqsadga xizmat qiladi - tasvir, tovush va pozani aniqlash. Yaxshi tomoni shundaki, u moslashuvchan. U modelni tasvirlar yoki jonli veb-kamera orqali tasvirlar yoki pozalarni tasniflashni o'rgatishi mumkin. Bu bepul va talabalar uchun eng yaxshisidir. Teachable Machine orqali yaratilgan model haqiqiy Tensorflow modeli bo'lib, uni veb-ilova, Android ilovasi yoki istalgan platforma bilan birlashtirish mumkin. Ikkalasi ham hisob yaratishni talab qilmaydi.

Amaliy qism:

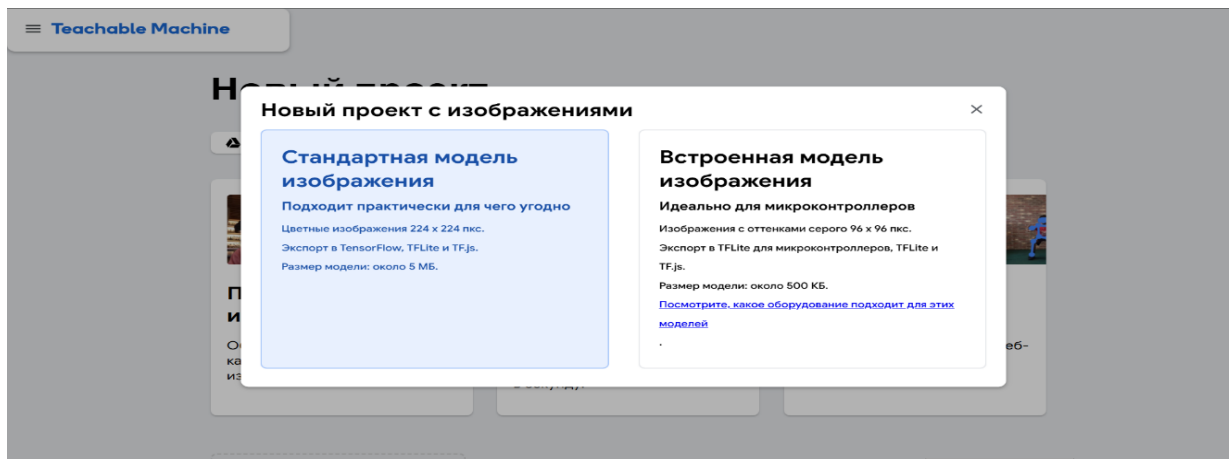
Modelni yarataylik

1-qadam: Teachable Machine (<https://teachablemachine.withgoogle.com/train>) ga o'ting. Siz uchta variantdan iborat bo'lgan quyida ko'rsatilgan ekranga yo'naltirilasiz - Tasvir, Audio va Pose.



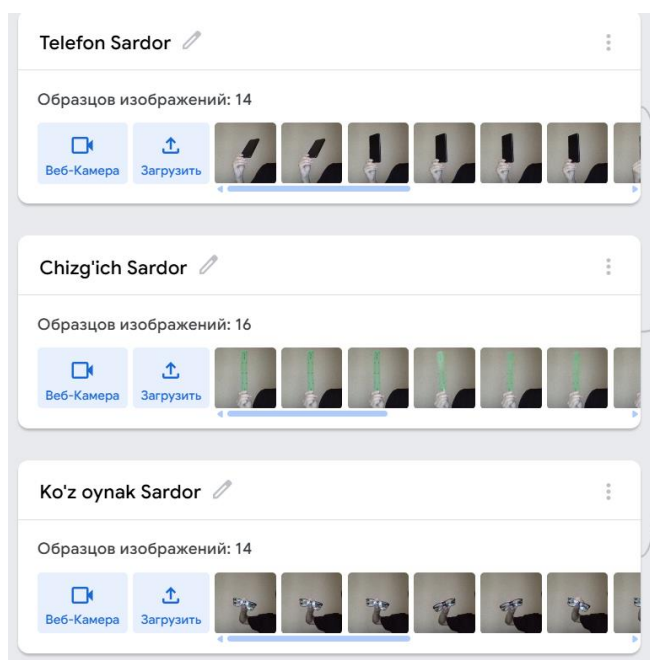
1-rasm

2-bosqich: Rasm loyihasini tanlang. Siz yana ikkita variantni ko'rasiz, standart va o'rnatilgan. Standartni tanlang, chunki biz uni mikro-kontrollerlar uchun yaratmayapmiz. Xo'sh, agar shunday bo'lsangiz, u holda o'rnatilgan rasm modelini tanlang. Agar siz "O'rnatilgan" ni tanlasangiz ham, butun jarayon bir xil bo'ladi, faqat model boshqacha bo'ladi.



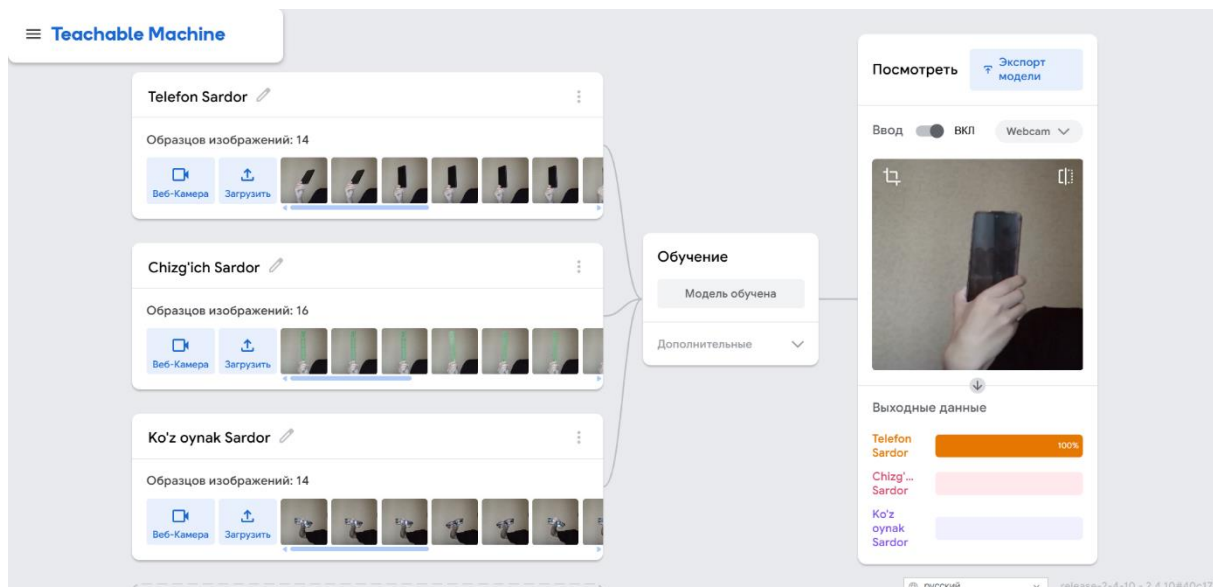
2-rasm

Standart tasvir loyihasini bosgandan so'ng, siz quyida ko'rsatilgan ekranga yo'naltirilasis, u yerda biz model yordamida tasniflamoqchi bo'lgan sinflarni qo'shamiz. Ikkita variant mavjud - siz ma'lumotlar to'plamidan rasmlarni yuklaysiz yoki tasvirlarni olish uchun jonli kameradan foydalanasis.



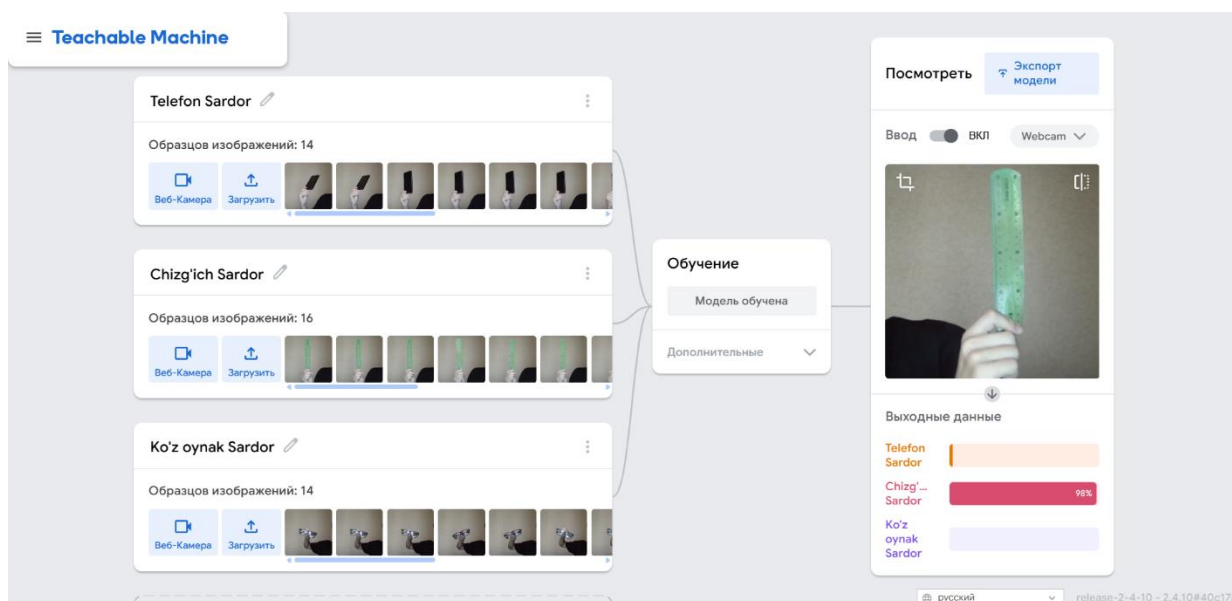
3-rasm

3-qadam: Endi men kabi sinflar yarating va rasmlarni yuklashni boshlang. Oddiylilik uchun men faqat uchta sinf yarataman – Telefon Sardor, Chizgich Sardor va Koz oynak Sardor. Shunday qilib, men 1-sinfni Telefon Sardor, 2-sinf nomini Chizgich Sardor va 3-sinf nomini Koz oynak Sardor bilan almashtiraman. Istaganingizcha yaratishingiz mumkin.



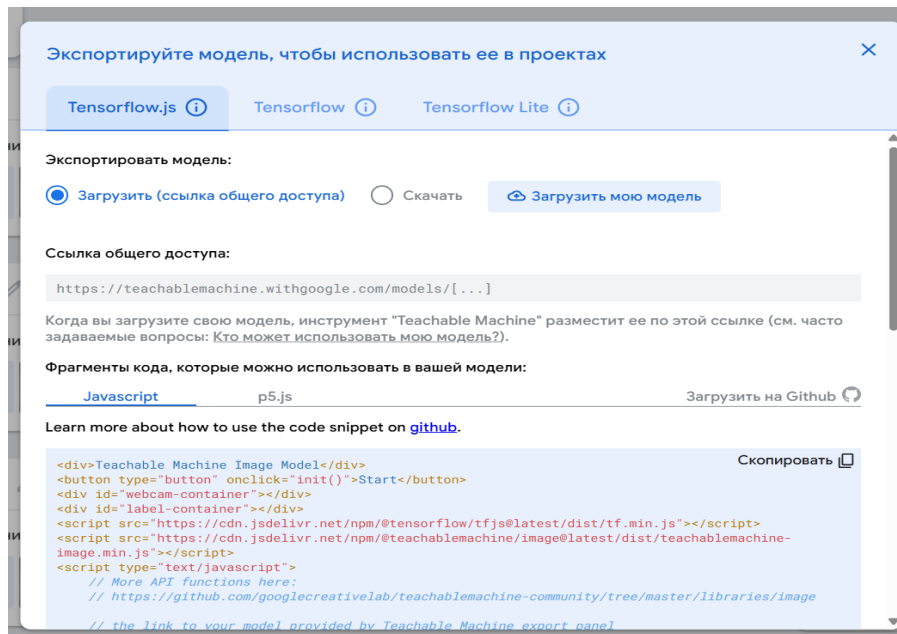
4-rasm

Rasmlarni yuklaganingizdan so'ng, "Training modeli" ni bosing. Siz turli xil variantlarni ko'rasiz - davrlar, partiya hajmi va o'rganish tezligi. Modelni yanada samaraliroq qilish uchun ular bilan o'ynash va qaysi qiymatlarda model eng yuqori aniqlikni berishini aniqlash muhimdir. Albatta, agar model etarlicha aniq bo'lmasa, unda hech qanday ma'no yo'q. Shunday qilib, siz ularning qadriyatlarini o'zgartirishingiz va eng yaxshisini topishingiz mumkin. Bu yerda men standart qiymatlar bilan ketyapman.



5-rasm

4-qadam: Model o'qitilgandan so'ng, modelni eksport qilish vaqti keldi.



6-rasm

```
<div>Teachable Machine Image Model</div>
<button type="button" onclick="init()">Start</button>
<div id="webcam-container"></div>
<div id="label-container"></div>
<script
src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest/dist/tf.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@teachablemachine/image@latest/dist/teachable-
machine-image.min.js"></script>
<script type="text/javascript">
  // More API functions here:
  // https://github.com/googlecreativelab/teachablemachine-
community/tree/master/libraries/image
// the link to your model provided by Teachable Machine export panel
const URL = "./my_model/";
let model, webcam, labelContainer, maxPredictions;
// Load the image model and setup the webcam
async function init() {
  const modelURL = URL + "model.json";
  const metadataURL = URL + "metadata.json";
// load the model and metadata
  // Refer to tmImage.loadFromFiles() in the API to support files from a file
picker
```

```

// or files from your local hard drive
// Note: the pose library adds "tmImage" object to your window
(window.tmImage)
model = await tmImage.load(modelURL, metadataURL);
maxPredictions = model.getTotalClasses();
// Convenience function to setup a webcam
const flip = true; // whether to flip the webcam
webcam = new tmImage.Webcam(200, 200, flip); // width, height, flip
await webcam.setup(); // request access to the webcam
await webcam.play();
window.requestAnimationFrame(loop);
// append elements to the DOM
document.getElementById("webcam-
container").appendChild(webcam.canvas);
labelContainer = document.getElementById("label-container");
for (let i = 0; i < maxPredictions; i++) { // and class labels
  labelContainer.appendChild(document.createElement("div"));
}
}
async function loop() {
  webcam.update(); // update the webcam frame
  await predict();
  window.requestAnimationFrame(loop);
}
// run the webcam image through the image model
async function predict() {
  // predict can take in an image, video or canvas html element
  const prediction = await model.predict(webcam.canvas);
  for (let i = 0; i < maxPredictions; i++) {
    const classPrediction =
      prediction[i].className + ": " + prediction[i].probability.toFixed(2);
    labelContainer.childNodes[i].innerHTML = classPrediction
  }
}
</script>

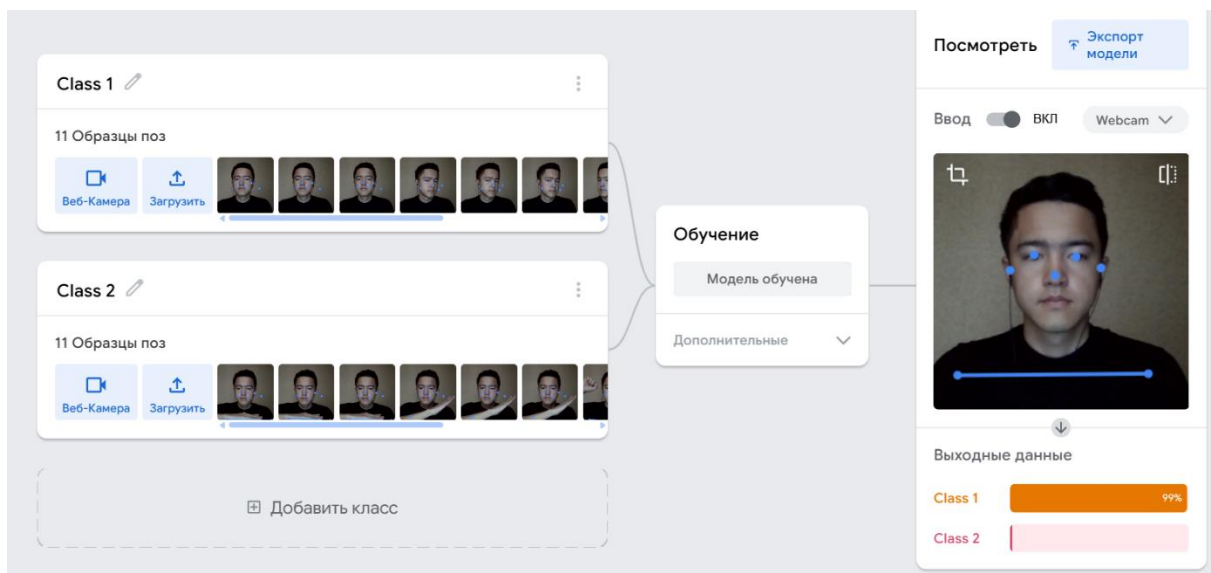
```

Modelni eksport qilish tugmasini yana bir marta bosganingizda, siz bir nechta variantni olasiz. Modelni ilovangiz bilan birlashtirishga yordam berish uchun hatto kod qatorlari ham berilgan. Tensorflow.js modellari har bir JavaScript kutubxonasi yoki ramkasi bilan ishlaydi. Ba'zi ramkalar faqat ma'lum turdagi modellarni qo'llab-quvvatlaydi. Shunday qilib, ramka yoki kutubxonangiz qaysilarini qo'llab-quvvatlashini tekshiring.

Modelni yuklab olish biroz vaqt olishi mumkin. Rasm pozalariga o'xshab, biz ham audio va pozalar uchun modellar yaratishimiz mumkin. Keling, buni qanday qilishimiz mumkinligini ko'rib chiqaylik.

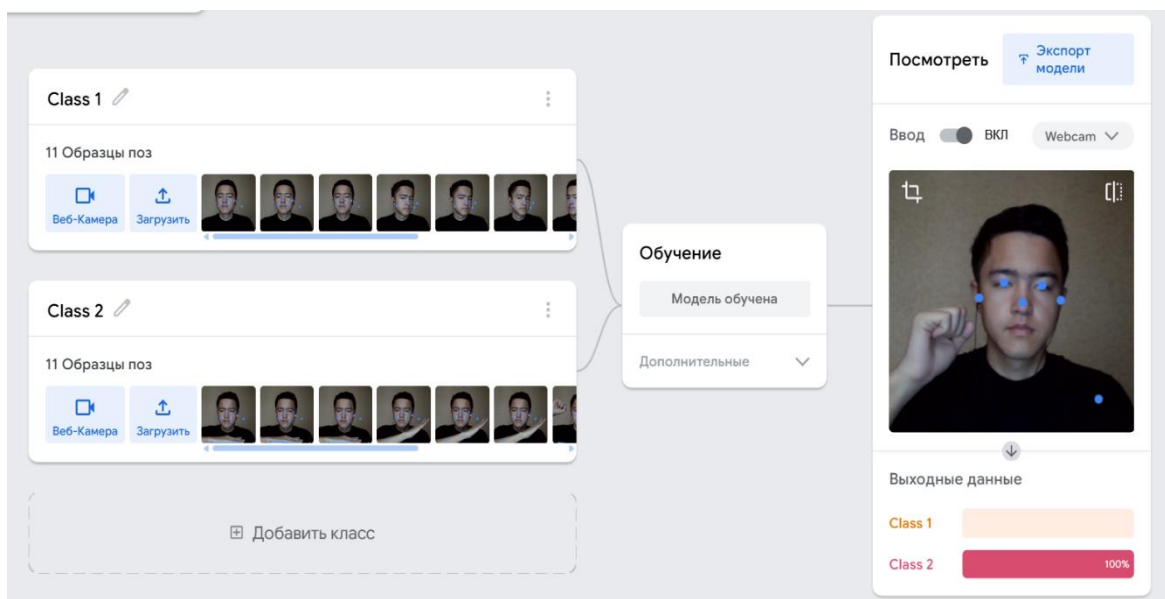
Pozalar modeli

Poza modelini yaratish uchun biz o'rgatiladigan mashinada poza loyihasini tanlashimiz kerak. Misolda, biz turish va o'tirish va tegishli tasvirlarni yuklash uchun ikki xil sinf yaratmoqdamiz. Biz modelni o'rgatishdan keyin boshlaymiz.



7-rasm

Trening tugagach, modelni eksport qilishdan oldin natijani ko'rish va uning samaradorligini ko'rish uchun istalgan rasmni yuklash orqali modelning chiqishini oldindan ko'ramiz. Quyidagi rasmda biz oldindan ko'rish uchun yuklagan rasmning chiqishi to'g'ri, ya'ni o'tirgan holda ekanligini ko'rishimiz mumkin. Bu model yaxshi ishlayotganini anglatadi.



8-Rasm

```
<div>Teachable Machine Pose Model</div>
<button type="button" onclick="init()">Start</button>
<div><canvas id="canvas"></canvas></div>
<div id="label-container"></div>
<script
  src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>
<script
  src="https://cdn.jsdelivr.net/npm/@teachablemachine/pose@0.8/dist/teachablemac
  hine-pose.min.js"></script>
<script type="text/javascript">

  // More API functions here:

  // https://github.com/googlecreativelab/teachablemachine-
  community/tree/master/libraries/pose

  // the link to your model provided by Teachable Machine export panel
  const URL = "./my_model/";
  let model, webcam, ctx, labelContainer, maxPredictions;

  async function init() {
    const modelURL = URL + "model.json";
    const metadataURL = URL + "metadata.json";

    // load the model and metadata
    // Refer to tmImage.loadFromFiles() in the API to support files from a file picker
    // Note: the pose library adds a tmPose object to your window (window.tmPose)
    model = await tmPose.load(modelURL, metadataURL);
    maxPredictions = model.getTotalClasses();

    // Convenience function to setup a webcam
    const size = 200;
    const flip = true; // whether to flip the webcam
```

```

await webcam.setup(); // request access to the webcam
await webcam.play();
window.requestAnimationFrame(loop);

// append/get elements to the DOM
const canvas = document.getElementById("canvas");
canvas.width = size; canvas.height = size;
ctx = canvas.getContext("2d");
labelContainer = document.getElementById("label-container");
for (let i = 0; i < maxPredictions; i++) { // and class labels
    labelContainer.appendChild(document.createElement("div"));
}
}

async function loop(timestamp) {
    webcam.update(); // update the webcam frame
    await predict();
    window.requestAnimationFrame(loop);
}

async function predict() {
    // Prediction #1: run input through posenet
    // estimatePose can take in an image, video or canvas html element
    const { pose, posenetOutput } = await model.estimatePose(webcam.canvas);
    // Prediction 2: run input through teachable machine classification model
    const prediction = await model.predict(posenetOutput);

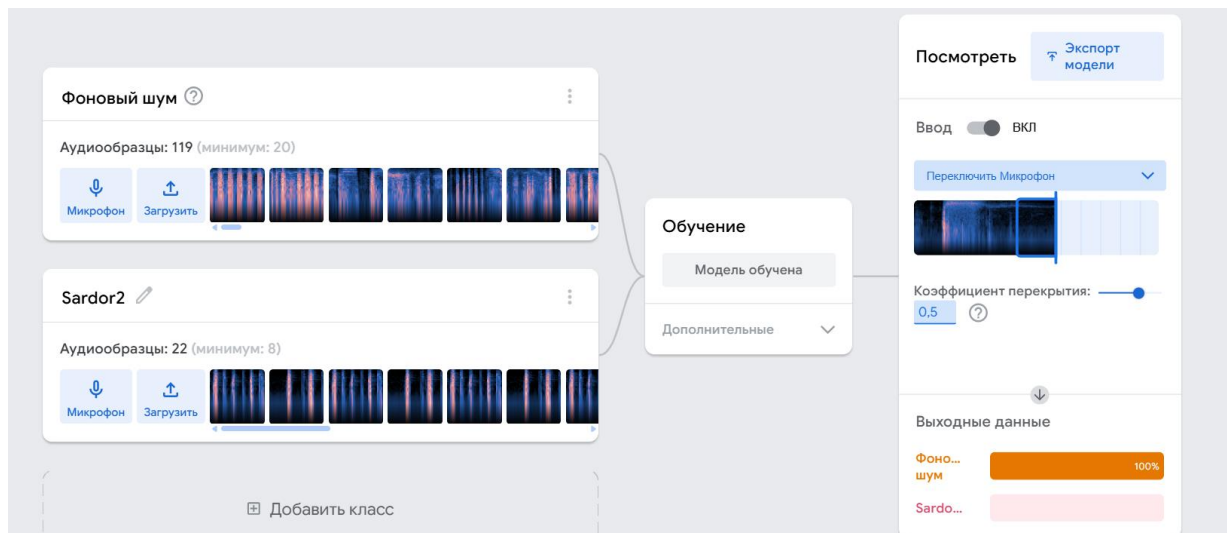
    for (let i = 0; i < maxPredictions; i++) {
        const classPrediction =

```

```
prediction[i].className + ": " + prediction[i].probability.toFixed(2);
    labelContainer.childNodes[i].innerHTML = classPrediction;
}
// finally draw the poses
drawPose(pose);
}
function drawPose(pose) {
    if (webcam.canvas) {
        ctx.drawImage(webcam.canvas, 0, 0);
        // draw the keypoints and skeleton
        if (pose) {
            const minPartConfidence = 0.5;
            tmPose.drawKeypoints(pose.keypoints, minPartConfidence, ctx);
            tmPose.drawSkeleton(pose.keypoints, minPartConfidence, ctx);
        }
    }
}
</script>
```

Audio modeli

Ovozni aniqlaydigan modelni yaratish uchun biz audio loyiha yaratamiz. Biz uchta sinf yaratdik - fon shovqini, qarsak chalish, yomg'ir va momaqaldiraq. Modelni o'qitgandan so'ng, oldindan ko'rish bo'limida biz model samaradorligini tekshirish uchun shovqindan foydalandik. Va oldindan ko'rish chiqishida biz fon shovqinining foizi yuqoriroq ekanligini ko'rishimiz mumkin. Modelni samaraliroq qilish uchun ma'lumotlardan modelni yaxshiroq o'rganish uchun minglab namunalarni qo'shishimiz kerak.



9-rasm

```
<div>Teachable Machine Audio Model</div>
```

```
<button type="button" onclick="init()">Start</button>
```

```
<div id="label-container"></div>
```

```
<script
```

```
src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>
```

```
<script
```

```
src="https://cdn.jsdelivr.net/npm/@tensorflow-models/speech-commands@0.4.0/dist/speech-commands.min.js"></script>
```

```
<script type="text/javascript">
```

```
// more documentation available at
```

```
// https://github.com/tensorflow/tfjs-models/tree/master/speech-commands
```

```
// the link to your model provided by Teachable Machine export panel
```

```
const URL = "./my_model/";
```

```
async function createModel() {  
    const checkpointURL = URL + "model.json"; // model topology  
    const metadataURL = URL + "metadata.json"; // model metadata  
  
    const recognizer = speechCommands.create(  
        "BROWSER_FFT", // fourier transform type, not useful to change  
        undefined, // speech commands vocabulary feature, not useful for your  
models  
        checkpointURL,  
        metadataURL);  
  
    // check that model and metadata are loaded via HTTPS requests.  
    await recognizer.ensureModelLoaded();  
  
    return recognizer;  
}
```

```
async function init() {  
    const recognizer = await createModel();  
    const classLabels = recognizer.wordLabels(); // get class labels  
    const labelContainer = document.getElementById("label-container");  
    for (let i = 0; i < classLabels.length; i++) {  
        labelContainer.appendChild(document.createElement("div"));  
    }  
  
    // listen() takes two arguments:  
    // 1. A callback function that is invoked anytime a word is recognized.  
    // 2. A configuration object with adjustable fields  
    recognizer.listen(result => {  
        const scores = result.scores; // probability of prediction for each class  
        // render the probability scores per class
```

```
for (let i = 0; i < classLabels.length; i++) {  
    const classPrediction = classLabels[i] + ": " + result.scores[i].toFixed(2);  
    labelContainer.childNodes[i].innerHTML = classPrediction;  
}  
}, {  
    includeSpectrogram: true, // in case listen should return result.spectrogram  
    probabilityThreshold: 0.75,  
    invokeCallbackOnNoiseAndUnknown: true,  
    overlapFactor: 0.50 // probably want between 0.5 and 0.75. More info in  
    README  
});  
  
// Stop the recognition in 5 seconds.  
// setTimeout(() => recognizer.stopListening(), 5000);  
}  
</script>
```

Teachable Machine hech qanday kodlashsiz mashinani o'rganish modellarini yaratish uchun veb-platformadir. Bu turli xil ilovalarda ishlatilishi mumkin bo'lgan modellarni o'rgatish va eksport qilish uchun foydalanuvchi grafik interfeysini ta'minlovchi Google loyihasidir.

Teachable Machine bilan mashinani o'rganish modelini yaratish qadamlari:

Ma'lumot to'plash: Modelni o'rgatmoqchi bo'lgan tasvirlar, tovushlar yoki boshqa ma'lumotlar to'plamini to'plang. Ikkilik tasniflash modelini o'rgatish uchun kamida ikkita ma'lumotlar sinfiga ega bo'lishingiz kerak.

Ma'lumotlarni yuklash: Ma'lumotlarni Teachable Machine-ga yuklang va har bir sinfni belgilang. Ma'lumotlar to'plami hajmini oshirish uchun o'rnatilgan ma'lumotlarni kengaytirish vositalaridan ham foydalanishingiz mumkin.

Modelni o'rgating: Teachable Machine uch xil o'qitish variantini taqdim etadi: tasvir, ovoz yoki poza. Foydalanayotgan ma'lumotlar turiga qarab tegishli ta'lim turini tanlang. Platforma oldindan o'rgatilgan chuqur o'rganish algoritmidan foydalangan holda modelni avtomatik ravishda o'rgatadi va sizga modelning aniqligini ko'rsatadi.

Modelni sinab ko'ring: Trening ma'lumotlar to'plamiga kiritilmagan yangi tasvirlar, tovushlar yoki pozalarni yuklash orqali modelni sinab ko'ring. Model yangi kirishlarni o'qitilgan sinflardan biriga tasniflaydi

Modelni eksport qiling: Modelni o'zingizning ilovangizda ishlatish uchun eksport qiling. Teachable Machine bir nechta eksport variantlarini qo'llab-quvvatlaydi, jumladan TensorFlow.js, TensorFlow Lite va Core ML.

Mashinani o'rganish modelini yaratish uchun Teachable Machine-dan foydalanishning ba'zi afzalliklari:

Foydalanish oson: Teachable Machine hech qanday kodlash qobiliyatini talab qilmaydigan foydalanuvchilarga qulay interfeysga ega.

Ma'lumotlar faniga oid tajriba talab qilinmaydi: Teachable Machine mashinani o'rganish modelini yaratish bilan bog'liq ko'plab vazifalarni avtomatlashtiradi va uni mutaxassis bo'lmaganlar uchun ochiq qiladi.

Bir nechta kiritish turlarini qo'llab-quvvatlaydi: Teachable Machine tasvir, tovush va poza mashqlarini qo'llab-quvvatlaydi, bu uni turli xil ilovalar uchun ko'p qirrali qiladi.

Tez takrorlash: Teachable Machine real vaqt rejimida modelning aniqligi haqida fikr-mulohazalarni taqdim etadi, bu esa modelni takrorlash va yaxshilashni osonlashtiradi.

Bir nechta eksport variantlarini qo'llab-quvvatlaydi: Teachable Machine bir nechta eksport opsiyalarini qo'llab-quvvatlaydi, bu modelni ilovangizga integratsiyalashni osonlashtiradi.

Nazorat savollari:

1. Teachable Machine qanday ishlaydi?
2. Teachable Machine qanday o'rganish amalga oshiradi?
3. Teachable Machine nima uchun foydalaniladi?
4. Teachable Machine haqida boshqa muhim tafsilotlar nima?

Topshiriqlar:

1. Uy jihozlarini aniqlab bera oladigan suniy intellektni yarating
2. Mobil telefonlar modelini aniqlab bera oladigan suniy intellektni yarating
3. Guruhdagi talabalarni rasmlariga ko'ra aniqlab bera oladigan suniy intellektni yarating.
4. Guruhdagi talabalarni ovozi aniqlab bera oladigan suniy intellektni yarating.
5. O'quv qurollarni aniqlab bera oladigan suniy intellektni yarating.
6. Ingliz tilida alifboni o'rgatadigan suniy intellektni yarating.
7. Xona jihozlarini aniqlab bera oladigan suniy intellektni yarating

Maxsus topshiriq:

Har bir talaba o'zini rasmini aniqlaydigan tahlilchi yaratishi lozim. Bunda har bir talaba ozidan keyin tartib boyicha 2 ta talabani rasmini ham joylashtiradi.

Topshiriqni bajarish bosqichlari:

1. Google Teachable Machine-ga kirish:

Google Teachable Machine platformasini qisqacha ko'rib chiqing. Uning maqsadini, yaratishi mumkin bo'lgan model turlarini (tasvir, tovush va poza) va foydalanish qulayligini tushuntiring.

2. Tasniflash uchun toifalarni tanlash:

Modelingiz tasniflaydigan uchta alohida toifani tanlang. Kategoriyalar vizual ravishda farqlanishiga ishonch hosil qiling.

Kategoriyalarga misollar:

- Meva turlari (olma, banan, apelsin)
- Qo'l imo-ishoralari (bosh barmog'i, tinchlik belgisi, ochiq qo'l)
- Kundalik buyumlar (qalam, chashka, kitob)

3. Ma'lumotlar yig'ish:

Veb-kamerangizdan foydalanib, har bir toifa uchun rasmlar ma'lumotlar to'plamini to'plang yoki yarating yoki oldindan mavjud rasmlarni yuklang. Yaxshi o'qitilgan modelni ta'minlash uchun har bir toifaga kamida 20 ta rasmni maqsad qilib qo'ying.

4. Modelni o'rgatish:

Rasmlaringizni Google Teachable Machine-ga yuklang.

Ekrandagi ko'rsatmalarga rioya qilib, modelni o'rgating.

Trening jarayonini boshlash uchun "Poyezd modeli" tugmasidan foydalaning. Platforma tomonidan taqdim etilgan aniqlik ko'rsatkichlariga rioya qiling.

5. Modelni sinovdan o'tkazish:

Uning ishlashini baholash uchun yangi tasvirlar yordamida modelni sinab ko'ring.

Har qanday noto'g'ri tasniflash yoki noaniqliklarni qayd qiling va modelning ishlashini yaxshilash uchun qo'shimcha o'quv ma'lumotlarini qo'shishni o'ylab ko'ring.

6. Modelni eksport qilish:

Modelning aniqligidan qoniqsangiz, uni eksport qiling. Google Teachable Machine modelni TensorFlow.js, yuklab olinadigan fayl yoki hatto havola sifatida ishlatish uchun eksport qilishga imkon beradi.

7. Oddiy veb-ilovani yaratish:

O'rganilgan modelingizni Google Teachable Machine tomonidan taqdim etilgan oddiy HTML sahifasiga joylashtiring.

Yangi kirishlarni to'g'ri tasniflash uchun modelni veb-ilovada sinab ko'ring.

8. Taqdim etish talablari:

Loyihani umumlashtiruvchi qisqacha hisobot (2-3 bet), jumladan:

- Tanlangan toifalar va ular nima uchun tanlangan.
- Ma'lumotlarni yig'ish jarayoni va modelni o'rgatish.
- Modelning ishlashi va duch kelgan har qanday qiyinchiliklar.

Veb-ilovaga havola yoki modelni amalda ko'rsatadigan video.

Loyihaning turli bosqichlarida (ma'lumotlarni yig'ish, o'qitish va sinovdan o'tkazish) Teachable Machine interfeysining skrinshotlari.

Baholash mezonlari:

- **Modelning aniqligi:** Model tanlangan toifalarni qanchalik yaxshi tasniflaydi.
- **Ijodkorlik:** toifalarni tanlash va ular trening ma'lumotlarida qanchalik yaxshi aks ettirilganligi.
- **Hisobot sifati:** hisobotning aniqligi, tashkiliyligi va to'liqligi.