

613-23 guruh Taabasi Mahmudov Asliddin \

## 1- Topshiriq

Bot yaratilish jarayoni va uni izohlar bilan screen shoti (bot silkasi yashil 24 shrifda)

The screenshot shows a Telegram chat interface with BotFather. The chat background is a green pattern with various icons. The BotFather profile is at the top left, showing 2,170,160 monthly users. The chat history includes a welcome message with links to the Bot API manual and a list of commands. The user interacts with BotFather by typing /start, /newbot, and /setname Ozod\_6bot. BotFather responds with instructions and confirmation. The chat is timestamped January 19, 1:51 PM to 1:53 PM.

**BotFather** 2,170,160 monthly users

What can this bot do?  
BotFather is the one bot to rule them all. Use it to create new bot accounts and manage your existing bots.

About Telegram bots:  
<https://core.telegram.org/bots>  
Bot API manual:  
<https://core.telegram.org/bots/api>  
Contact @BotSupport if you have questions about the Bot API.

January 19

/start 1:11 AM ✓

I can help you create and manage Telegram bots. If you're new to the Bot API, please [see the manual](#).

You can control me by sending these commands:

/newbot - create a new bot  
/mybots - edit your bots

**Edit Bots**  
/setname - change a bot's name  
/setdescription - change bot description  
/setabouttext - change bot about info  
/setuserpic - change bot profile photo  
/setcommands - change the list of commands  
/deletebot - delete a bot

**Bot Settings**  
/token - generate authorization token  
/revoke - revoke bot access token  
/setinline - toggle inline mode  
/setinlinegeo - toggle inline location requests  
/setinlinefeedback - change inline feedback settings

[1, 2] we understand that developers providing HTML5-experiences for Telegram users may expect financial returns on their work. To that end, we will introduce unified monetization tools for our Bot API later.

1:51 PM

Accept 1:52 PM ✓

You don't have any bots yet. Use the /newbot command to create a new bot first.

1:52 PM

/newbot 1:52 PM ✓

Alright, a new bot. How are we going to call it? Please choose a name for your bot.

1:52 PM

Ozod 1:53 PM ✓

Good. Now let's choose a username for your bot. It must end in 'bot'. Like this, for example: TetrisBot or tetris\_bot.

1:53 PM

Ozod\_6bot 1:53 PM ✓

Done! Congratulations on your new bot. You will find it at [t.me/Ozod\\_6bot](https://t.me/Ozod_6bot). You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

**Bot Info**

BotFather 2,170,160 monthly users

BotFather is them all. Use bot accounts existing bots  
Description  
@BotFather  
Username

Notifications

1 photo  
3 shared link

Bot Privacy P

Report  
Stop and blo

Success! Profile photo updated. [/help](#) 2:01 PM

[/setdescription](#) 2:05 PM ✓✓

Choose a bot to change description. 2:05 PM

@Ozod\_6bot 2:05 PM ✓✓

OK. Send me the new description for the bot. People will see this description when they open a chat with your bot, in a block titled "What can this bot do?".

2:05 PM

Asliddin Mahmudov 2:05 PM ✓✓

Success! Description updated. You will be able to see the changes within a few minutes. [/help](#)

2:05 PM

@Ozod\_6bot 2:01 PM ✓✓

OK. Send me the new profile photo for the bot. 2:01 PM



[/setcommands](#) 2:24 PM ✓✓

Choose a bot to change the list of commands. 2:24 PM

@Ozod\_6bot 2:24 PM ✓✓

OK. Send me a list of commands for your bot. Please use this format:

command1 - Description  
command2 - Another description

Send [/empty](#) to keep the list empty.

2:24 PM

command1-Uzbek  
command2-Russian  
command3-English

2:25 PM ✓✓

Success! Command list updated. [/help](#) 2:25 PM

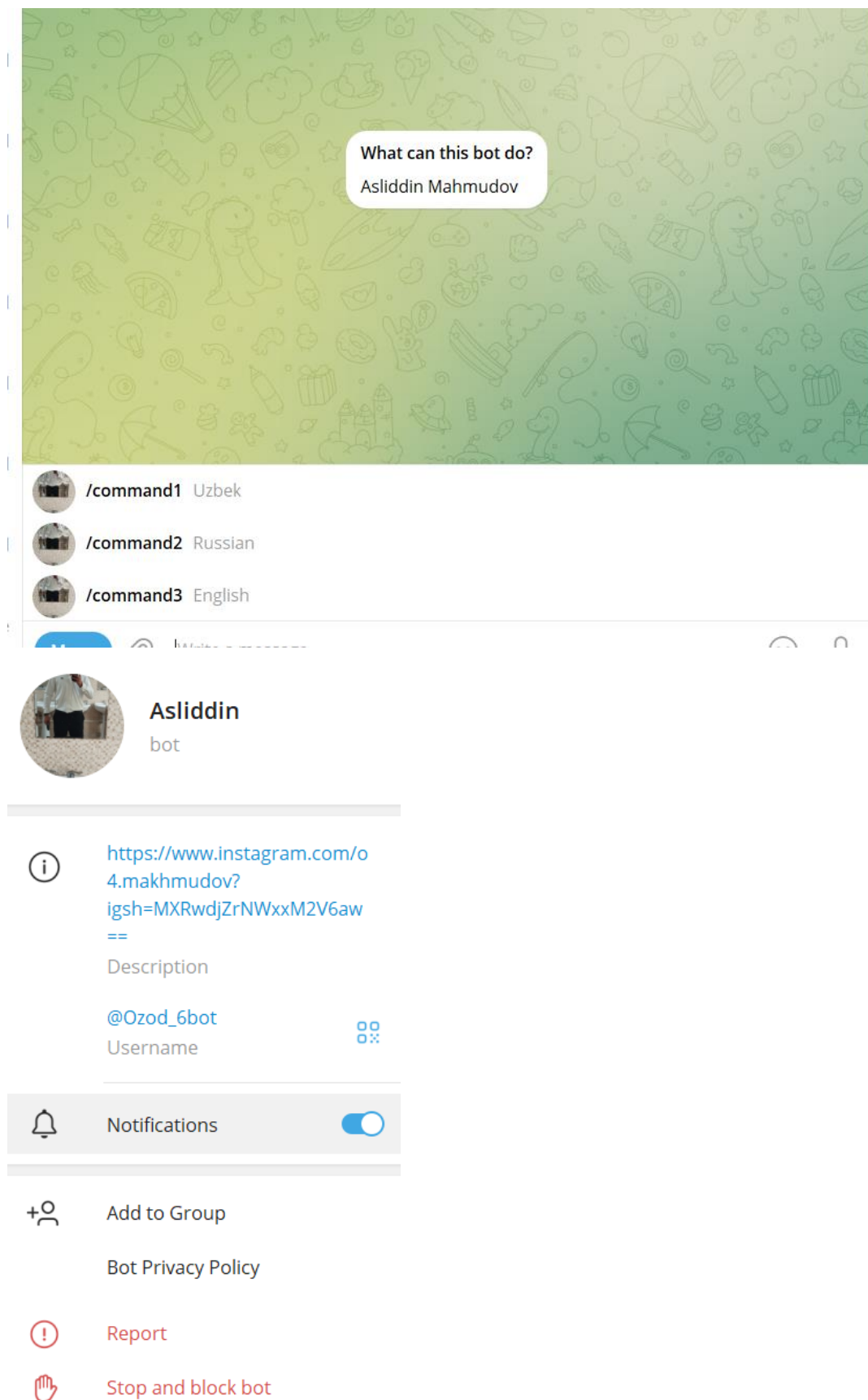
[/start](#) 2:32 PM ✓✓

OK. Send me a list of commands for your bot. Please use this format:

command1 - Description  
command2 - Another description

Send [/empty](#) to keep the list empty.

2:32 PM



## 2- Topshiriq

Teachable machineda ma'lumotlar tahlili va dastur kodi

```

<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-

```

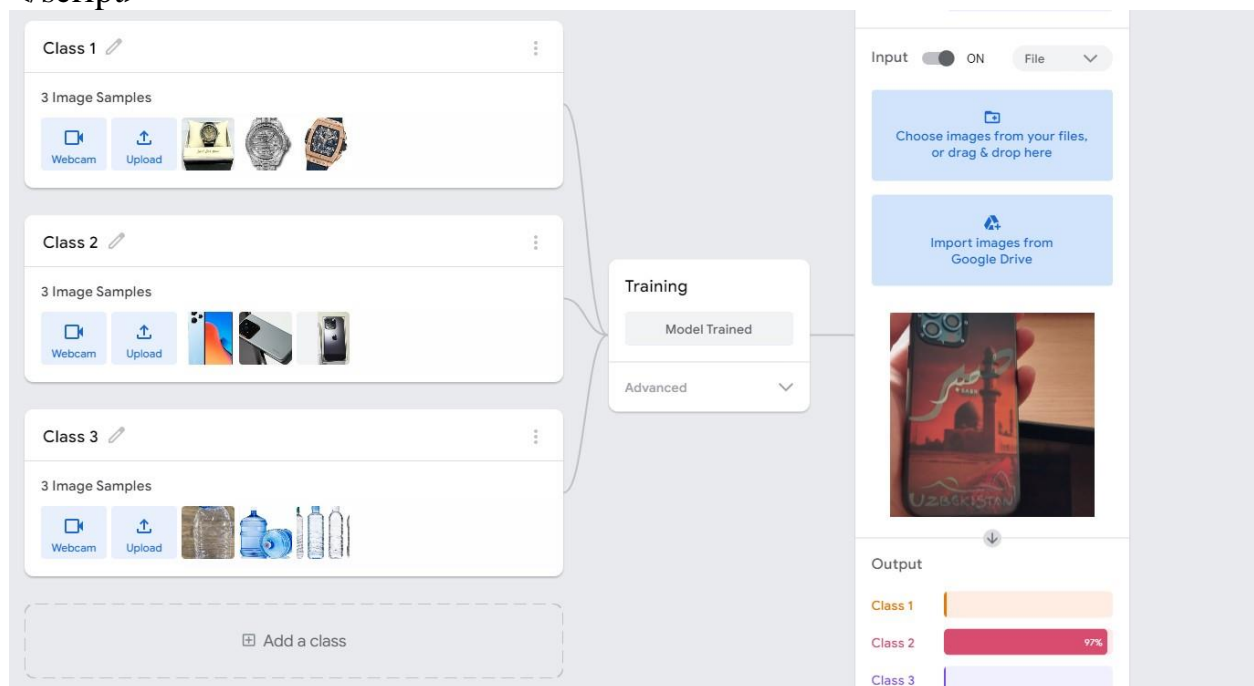
```

        container").appendChild(webcam.canvas);
        document.getElementById("label-container");
        i++) {
            //
            labelContainer.appendChild(document.createElement("div"));
        }

        async function loop() {
            webcam.update(); // update the webcam frame
            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
            prediction = await model.predict(webcam.canvas);
            for (let i = 0; i < maxPredictions; i++) {
                prediction[i].className
                prediction[i].probability.toFixed(2);
                labelContainer.childNodes[i].innerHTML = classPrediction;
            }
        }
    </script>

```



Preview



Export Model

Input



ON

File



Choose images from your files,  
or drag & drop here



Import images from  
Google Drive

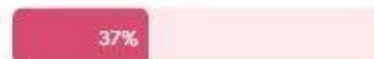


Output

Class 1



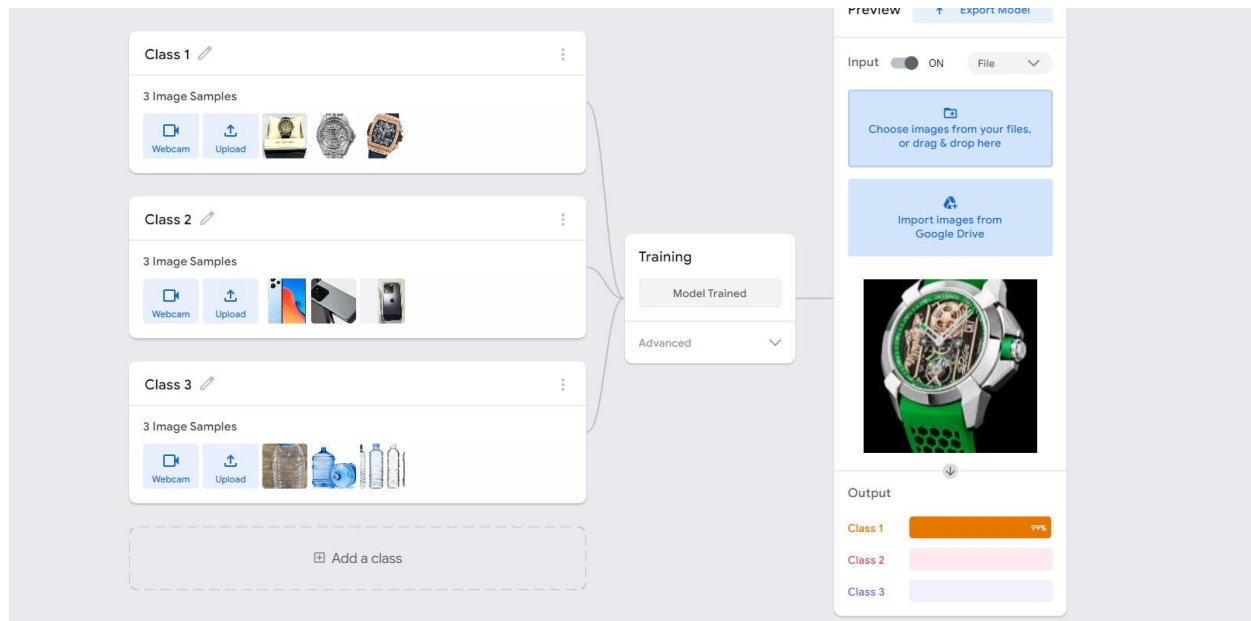
Class 2



Class 3







```

<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.tlImage)          model = await tlImage.load(modelURL,
metadataURL);          maxPredictions = model.getTotalClasses();

    // Convenience function to setup a webcam          const flip = true; //
whether to flip the webcam          webcam = new tlImage.Webcam(200, 200, flip);
// width, height, flip          await webcam.setup(); // request access to the webcam
await webcam.play();          window.requestAnimationFrame(loop);

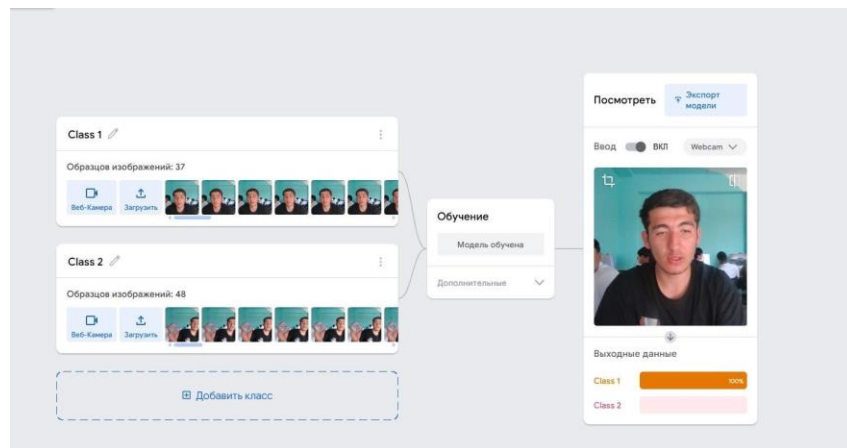
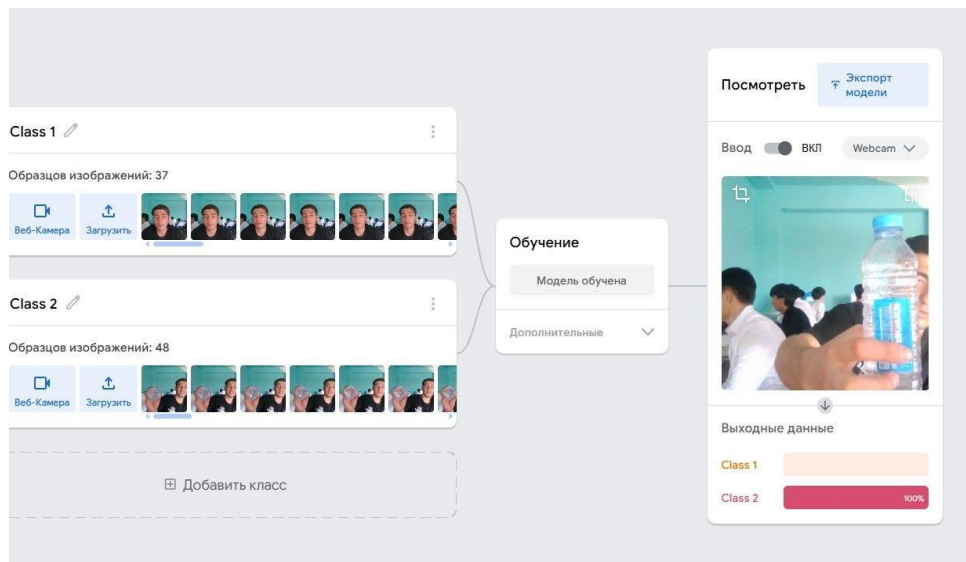
    //          append          elements          to          the          DOM
document.getElementById("webcam-
container").appendChild(webcam.canvas);          labelContainer =
document.getElementById("label-container");          for (let i = 0; i < maxPredictions;
i++)          {          //          and          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>

```





```

<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();
```

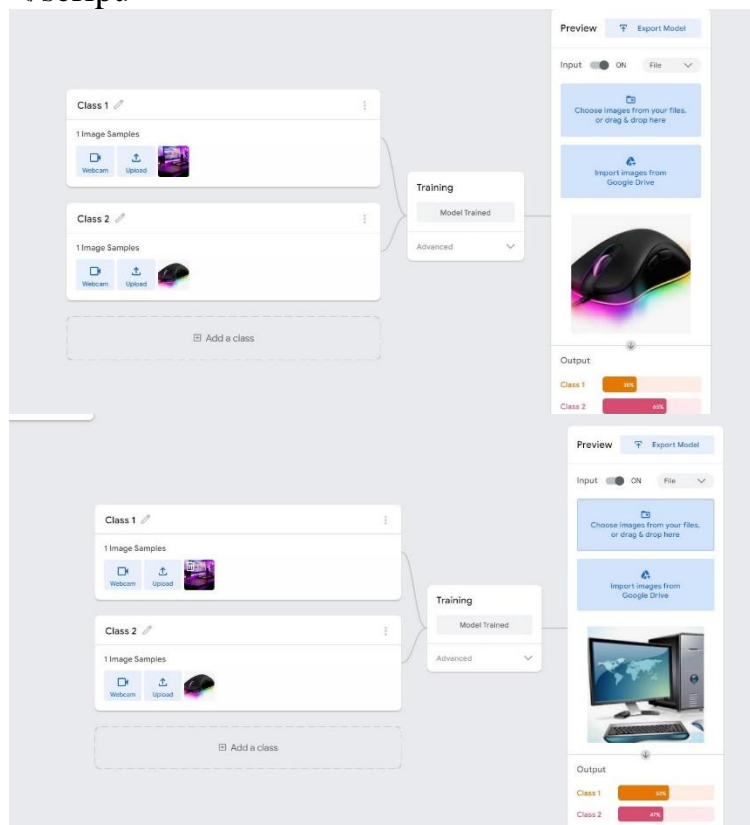
```
    // Convenience function to setup a webcam        const flip = true; //
whether to flip the webcam        webcam = new tmImage.Webcam(200, 200, flip);
// width, height, flip        await webcam.setup(); // request access to the webcam
await webcam.play();        window.requestAnimationFrame(loop);
```

```
    //        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>



### 3-topshiriq

<https://github.com/asliddinmaxmudov07>

**GitHub**  
**asliddinmaxmudov07 - Overview**  
Qarshingda kimligi emas  
Yonigda kiligi muhim. GitHub is where  
asliddinmaxmudov07 builds software.

2:27 PM ✓

**Overview** Repositories 2 Projects Packages Stars

**Popular repositories**

- [pandas-va-numpy-in-python](#) Public
- [Pandas-with-excel](#) Public

**3 contributions in the last year**

Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct

Mon

Wed

Fri

Learn how we count contributions

Less More

**Contribution activity**

2024

October 2024

Created their first repository Oct 9

**Dóstar**  
asliddinmaxmudov07

Follow

Qarshingda kimligi emas Yonigda kiligi muhim

TATUFF

Qashqadaryo viloyati Shahrisabz shahri

13:59 - same time

Joined 3 weeks ago

github.com/asliddinmaxmudov07

asliddinmaxmudov07

Overview Repositories 2 Projects Packages Stars

Popular repositories

[pandas-va-numpy-in-python](#) Public [Pandas-with-excel](#) Public

3 contributions in the last year

Contribution settings 2024

Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep

Mon

Wed

Fri

Learn how we count contributions

Less More

Contribution activity

October 2024

Created their first repository Oct 9

Dóstlar

asliddinmaxmudov07

Qarshingda kimligi emas Yonigda kiligi muhim

Edit profile

TATUFF

Qashqadaryo viloyati Shahrizabz shahri

14:10 (UTC +05:00)

asliddinmaxmudov071@gmail.com

## 4 topshiriq

Pandas va numpy kutubhonasi google colabda aks ettirish va github ga joylash

```

df = pd.DataFrame(data)
# 2. Ma'lumotlarni ko'rish
print(df)
# 3. Filtrlash
young_people = df[df['Yoshi'] < 30]
print("30 yoshdan kichiklar:\n", young_people)
# 4. O'zgartirish
df['Yoshi'] += 1 # Har bir shaxsning yoshini 1 ga oshirish
print("Yangilangan DataFrame:\n", df)
# 5. CSV formatda saqlash
df.to_csv('data.csv', index=False)

```

```

➡      Ism  Yoshi   Shahar
0     Ali    25   Toshkent
1     Vali    30  Samarqand
2  Sardor    22    Buxoro
30 yoshdan kichiklar:
      Ism  Yoshi   Shahar
0     Ali    25   Toshkent
2  Sardor    22    Buxoro
Yangilangan DataFrame:
      Ism  Yoshi   Shahar
0     Ali    26   Toshkent
1     Vali    31  Samarqand
2  Sardor    23    Buxoro

```

Далее: [Объяснить ошибку](#)

```

✓ [9] print (df.head(10))
0
ЭК.

```

```

➡      Ism  Yoshi   Shahar
0     Ali    26   Toshkent
1     Vali    31  Samarqand
2  Sardor    23    Buxoro

```



ndarray: array\_1d

ndarray with shape (5,)

```
array_1d = np.array([1, 2, 3, 4, 5])
array_2d = np.array([[1, 2, 3], [4, 5, 6]])
# 2. Matematik operatsiyalar
sum_array = np.sum(array_1d)
mean_array = np.mean(array_1d)
product_array = np.prod(array_1d)
print("1D Massiv: ", array_1d)
print("2D Massiv:\n", array_2d)
print("Massivlar yig'indisi: ", sum_array)
print("O'rtacha: ", mean_array)
print("Ko'paytma: ", product_array)
```



```
1D Massiv:  [1 2 3 4 5]
2D Massiv:
[[1 2 3]
 [4 5 6]]
Massivlar yig'indisi: 15
O'rtacha: 3.0
Ko'paytma: 120
```

```
[6] import pandas as pd
# 1. DataFrame yaratish
data = {
    'Ism': ['Ali', 'Vali', 'Sardor'],
    'Yoshi': [25, 30, 22],
    'Shahar': ['Toshkent', 'Samarqand', 'Buxoro']
```