

Face filters

Session 16





Topics covered

- Introduction to face filter
- Activity: Face filter







Face filter







www.ai.thestempedia.com



Introduction to Artificial Intelligence

Face filters work by using a combination of facial recognition technology and augmented reality. Facial recognition software identifies key points on the face, such as the eyes, nose, mouth, and cheekbones, and maps them onto a 3D model. Augmented reality technology then overlays virtual objects onto the face, such as hats, glasses, or other accessories. The result is a live image of the user's face with a virtual filter applied.

In this lesson, you will gain knowledge through constructing a project. Let's apply a filter to your face that will move in any direction as you move.







Filters



ACTIVITY Face filter

Let's create a face filter which move along with the face position.

Step – by – step:

- Follow the steps below:
- Click on the **PictoBlox** icon on your in this session,
 - Open a **New file** in PictoBlox you will learn how face filters work, by going through the following steps:
 - Select the coding environment as Python Coding
 - Add a sprite named "Face Filters" from the library.
 - Go to the "Face Filter" py file from the Project Files.
 - We need to initiate Face detection () and Pen() objects to use their functions.
- Turn on the camera using the video([1], [2]) function. We will set the transparency parameter [2] to 0.
- Disable the bounding box using **disablebox()** function.
- Use set threshold ([1]): To set the threshold/confidence. It takes in one parameter. This parameter decides the threshold/confidence with which a face gets detected, 0 to 100%.







ACTIVITY Face filter

fd = FaceDetection()
fd.video("on", 0) # to turn on video with 0% transparency
fd.disablebox() # to disable bounding box
fd.setthreshold(0.4) # to set the threshold at 0.4



- Start the loop with while 1 to run forever. Inside this loop we add the following two functions:
 - **fd.analysecamera()**: to analyse the video from the camera, on PictoBlox stage.
 - pen.clear(): to clear any pen markings from the previous run of the code.
- If the face is detected, we will now start a for loop to apply the filter to each face detected by the camera. To do this, we will use the range() function in Python, with the parameters start, stop, and step.

for i in range(1, fd.count()+1):

• We will now Use landmarksx and landmarksy to calculate dx and dy between right and left eye face landmarks for face number [1], landmark number [2]. the landmark.





6

ACTIVITY Face filter

• Having calculated dx and dy, we can now use the math library's functions to determine the Angle of tilt of the detected face.

dx = float(fd.landmarksx(i, "right eye")) - float(fd.landmarksx(i, "left eye"))
dy = float(fd.landmarksy(i, "right eye")) - float(fd.landmarksy(i, "left eye"))

- math.atan2(dx, dy) This function calculates the angle of tilt in 'radians'
- math.degrees() This function converts the unit of angle calculated from 'radians' to 'degrees'.

Angle = math.degrees(math.atan2(dx, dy))

- Now, we will set various properties of the Face Filters sprite, using the respective sprite functions:
 - **setx()**: To set the x-position of the sprite.
 - **sety()**: To set the y-position of the sprite
 - **setsize()**: To set the size of sprite
 - **setdirection()**: To set the angle of sprite
- Using the pen.stamp() function to stamp the filter image.
- If no faces are detected, the sprite should state "No Face Detected









• Full code is provided on the next page. Press the run button to test the code.

sprite.setx(fd.x(i))
sprite.sety(fd.y(i))
sprite.setsize(fd.width(i))
sprite.setdirection(Angle)







Lets Code

sprite = Sprite('Face Filters') import math fd = FaceDetection() pen = Pen() fd.video("on") fd.disablebox() fd.setthreshold(0.4) while 1: fd.analysecamera() pen.clear() if fd.count() > 0: sprite.say("Face Detected ")

for i in range(1, fd.count()+1):

float(fd.landmarksx(i, "left eye")) float(fd.landmarksy(i, "left eye")) sprite.setx(fd.x(i)) sprite.sety(fd.y(i)) sprite.setsize(fd.width(i)) sprite.setdirection(Angle) pen.stamp() else:



```
dx = float(fd.landmarksx(i, "right eye")) -
```

```
dy = float(fd.landmarksy(i, "right eye")) -
```

```
Angle = math.degrees(math.atan2(dx, dy))
```

```
sprite.say("No Face Detected")C
```



Lets Code

SAVING THE PROGRAM Save the project file as **Face Filter**.





Output







www.ai.thestempedia.com