PEGHEADS.ORG
GLOBAL DIGITAL AI EDUCATIONAL ORGANIZATION

TESTING THE FACIAL EMOTION RECOGNITION (FER) ALGORITHM ON ANIMATIONS

1. Input the animation video to the algorithm
2. Run the Facial emotion recognition Algorithm
3. Analyze the results
4. Crop and resize the images for better results
5. Repeat first 2 steps and analyze the results again

GOAL

Recognize different emotions in the animation video

STEP 1

Prepare the input animation video

You can use any video format as input, if you do not have a video and have only animated images. You can combine those images into a video format with the following code.

```
import cv2
import os

# Update the path to your images folder
images_folder = ""

video_name = "video.avi"
images = [img for img in os.listdir(images_folder) if img.endswith(("png"))]
frame = cv2.imread(os.path.join(images_folder, images[0]))
height, width, layers = frame.shape

video = cv2.VideoWriter(video_name, 0, 1, (width, height))

for image in images:
    video.write(cv2.imread(os.path.join(images_folder, image)))

cv2.destroyAllWindows()
video.release()
```

We join all the images that end with .png. All the images must of some file format.
As you can see we are using the opencv library which has the function **VideoWriter**, this creates a video with the video_name specified.

The **cv2.VideoWriter** requires five parameters:

- The first parameter is the path to the output video file.
- Secondly, we need to supply the **fourcc** codec.
- The third argument is the desired FPS of the output video file.
- We then have the **width** and **height** of the output video. It's important that you set these values correctly, otherwise OpenCV will throw an error if you try to write a frame to a file that has different dimensions than the ones supplied to a **cv2.VideoWriter**.
- Finally, the last parameter controls whether or not we are writing color frames to file. A value of **True** indicates that we are writing color frames. Supplying **False** indicates we are not writing color frames.

STEP 2

Run the Facial emotion recognition Algorithm

The model is a convolutional neural network with weights saved to HDF5 file in the data folder relative to the module's path.

We are experimenting on this to see the results with animated images and try to modify it to get better results of emotion recognition in animations.

The algorithm divides the video into frames and saves the frames in which it recognizes the emotions in a separate output folder.

For more information about fer **https://github.com/justinshenk/fer**

```
from fer import Video
from fer import FER

# Update the path to your video file
video_filename = "./video.avi"
video = Video(video_filename)

# Analyze video, displaying the output
detector = FER(mtcnn=True)
video.analyze(detector, display=True)
```

STEP 3

Analyze the results

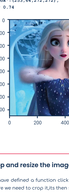
We can compare the results of human faces to the results of animated images.

```
[{'box':(135,122,206,206),
happy 1.66
```

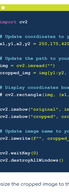
As you can see it clearly identifies human face emotion

Some of the animated images, it didn't recognize any emotions.

```
[]
None None
```



```
[]
None None
```



But it does recognize animated human faces as shown below.

```
{'box':(253,66,272,272),
fear 0.74
```

So we tried cropping the images and resizing them to see if we get better predictions as shown in the next step.

STEP 4

Crop and resize the images for better results.

We have defined a function click event in order to get the exact coordinates of the image where we need to crop XJXs then saved in the variables x1,x2,y1 and y2

```
import cv2

# Update the path to your image
image = cv2.imread("")

def click_event(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:
        cv2.putText(image, str(x) + ",
" + str(y), (x, y),
        cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,0), 2)

        cv2.imshow("image", image)

        print(f"{x}, {y}.format(x, y))

cv2.imshow("image", image)
cv2.setMouseCallback("image", click_event)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In the code shown below ,we define the cropping box size using the variable x1,x2,y1,y2 and then crop the image using img[y1:y2, x1:x2]

```
import cv2

# Update coordinates to your needs
x1,y1,x2,y2 = 250,175,420,450

# Update the path to your image
img = cv2.imread("")
cropped_img = img[y1:y2, x1:x2]

# Display coordinates box around the image
# cv2.rectangle(img, (x1, y1), (x2, y2), (0,0,0), 2)

cv2.imshow("Original", img)
cv2.imshow("Cropped", cropped_img)

# Update image name to your own
cv2.imwrite(f"", cropped_img)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

To resize the cropped image to the required size, use the following code

Resize Thumbnail

```
from PIL import Image
import os

# Update the path to your images folder
images_folder = ""

images = [img for img in os.listdir(images_folder) if
    img.endswith(("png"))]
for image in images:
    img = Image.open(f"{images_folder}/{image}")

    # Images cannot be resized if it is smaller than the thumbnail size
    img.thumbnail((400, 400))

    img.save(f"resized_{image}", "png")
print(f"{image} was resized")
```

The img.thumbnail will do the resizing for you, you just need to specify the size. Here we have made all the images of size 400x600

STEP 5

Repeat the first 2 steps and analyze the results again

As we saw in the above steps the model didn't recognize any emotions on furry peg or bat peg, so we resized the images and made a video from those images using step 4 and step 1.

Then we ran FER again with this video input as explained in step 2.

The results with the cropped and resized inputs are as follows.

It now detects a few of the frames with emotions. The detailed results can be found in our code.

Try our code in jupyter notebook. The git link is provided below.

As an improvement to this experiment, we are trying to use GAN (Generative adversarial network), which can be discussed in further tutorials.

STAY TUNED FOR OUR NEXT TUTORIAL

If you would like to contribute to this tutorial with any information or tests and upload to our GitHub please do. This is an open platform to help anyone learn AI and machine learning skills.

https://github.com/Pegheads-inc/Pegheads-Tutorial-3/issues

PEGHEADS.ORG
GLOBAL DIGITAL AI EDUCATIONAL ORGANIZATION