|  |
| --- |
| @@ -0,0 +1,157 @@ |
|  |  | { |
|  |  | "cells": [ |
|  |  | { |
|  |  | "cell\_type": "markdown", |
|  |  | "metadata": {}, |
|  |  | "source": [ |
|  |  | "# Drowsiness Detection OpenCV\n", |
|  |  | "\n", |
|  |  | "\n", |
|  |  | "This code can detect your eyes and alert when the user is drowsy.\n", |
|  |  | "\n", |
|  |  | "## Applications\n", |
|  |  | "This can be used by riders who tend to drive for a longer period of time that may lead to accidents.\n", |
|  |  | "\n", |
|  |  | "### Algorithm\n", |
|  |  | "\n", |
|  |  | "Each eye is represented by 6 (x, y)-coordinates, starting at the left-corner of the eye (as if you were looking at the person), and then working clockwise around the eye:.\n", |
|  |  | "\n", |
|  |  | "<img src=\"eye1.jpg\">\n", |
|  |  | "\n", |
|  |  | "### Condition\n", |
|  |  | "\n", |
|  |  | "It checks 20 consecutive frames and if the Eye Aspect ratio is lesst than 0.25, Alert is generated.\n", |
|  |  | "\n", |
|  |  | "#### Relationship\n", |
|  |  | "\n", |
|  |  | "<img src=\"eye2.png\">\n", |
|  |  | "\n", |
|  |  | "#### Summing up\n", |
|  |  | "\n", |
|  |  | "<img src=\"eye3.jpg\">\n", |
|  |  | "\n", |
|  |  | "jupyter kernel by Manuel Romero (mrm8488@gmail.com or @mrm8488)" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 1, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "from scipy.spatial import distance\n", |
|  |  | "from imutils import face\_utils\n", |
|  |  | "import imutils\n", |
|  |  | "import dlib\n", |
|  |  | "import cv2" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 2, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "def eye\_aspect\_ratio(eye):\n", |
|  |  | "\tA = distance.euclidean(eye[1], eye[5])\n", |
|  |  | "\tB = distance.euclidean(eye[2], eye[4])\n", |
|  |  | "\tC = distance.euclidean(eye[0], eye[3])\n", |
|  |  | "\tear = (A + B) / (2.0 \* C)\n", |
|  |  | "\treturn ear" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 3, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "thresh = 0.25\n", |
|  |  | "frame\_check = 20\n", |
|  |  | "detect = dlib.get\_frontal\_face\_detector()\n", |
|  |  | "predict = dlib.shape\_predictor(\"shape\_predictor\_68\_face\_landmarks.dat\")" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 4, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "(lStart, lEnd) = face\_utils.FACIAL\_LANDMARKS\_IDXS[\"left\_eye\"]\n", |
|  |  | "(rStart, rEnd) = face\_utils.FACIAL\_LANDMARKS\_IDXS[\"right\_eye\"]" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": 5, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [ |
|  |  | "cap=cv2.VideoCapture(0)\n", |
|  |  | "flag=0\n", |
|  |  | "while True:\n", |
|  |  | "\tret, frame=cap.read()\n", |
|  |  | "\tframe = imutils.resize(frame, width=450)\n", |
|  |  | "\tgray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)\n", |
|  |  | "\tsubjects = detect(gray, 0)\n", |
|  |  | "\tfor subject in subjects:\n", |
|  |  | "\t\tshape = predict(gray, subject)\n", |
|  |  | "\t\tshape = face\_utils.shape\_to\_np(shape)#converting to NumPy Array\n", |
|  |  | "\t\tleftEye = shape[lStart:lEnd]\n", |
|  |  | "\t\trightEye = shape[rStart:rEnd]\n", |
|  |  | "\t\tleftEAR = eye\_aspect\_ratio(leftEye)\n", |
|  |  | "\t\trightEAR = eye\_aspect\_ratio(rightEye)\n", |
|  |  | "\t\tear = (leftEAR + rightEAR) / 2.0\n", |
|  |  | "\t\tleftEyeHull = cv2.convexHull(leftEye)\n", |
|  |  | "\t\trightEyeHull = cv2.convexHull(rightEye)\n", |
|  |  | "\t\tcv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)\n", |
|  |  | "\t\tcv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)\n", |
|  |  | "\t\tif ear < thresh:\n", |
|  |  | "\t\t\tflag += 1\n", |
|  |  | "\t\t\t#print (flag)\n", |
|  |  | "\t\t\tif flag >= frame\_check:\n", |
|  |  | "\t\t\t\tcv2.putText(frame, \"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ALERT!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\", (10, 30),\n", |
|  |  | "\t\t\t\t\tcv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0, 0, 255), 2)\n", |
|  |  | "\t\t\t\tcv2.putText(frame, \"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ALERT!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\", (10,325),\n", |
|  |  | "\t\t\t\t\tcv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0, 0, 255), 2)\n", |
|  |  | "\t\telse:\n", |
|  |  | "\t\t\tflag = 0\n", |
|  |  | "\tcv2.imshow(\"Frame\", frame)\n", |
|  |  | "\tkey = cv2.waitKey(1) & 0xFF\n", |
|  |  | "\tif key == ord(\"q\"):\n", |
|  |  | "\t\tcv2.destroyAllWindows()\n", |
|  |  | "\t\tcap.release()\n", |
|  |  | "\t\tbreak\n" |
|  |  | ] |
|  |  | }, |
|  |  | { |
|  |  | "cell\_type": "code", |
|  |  | "execution\_count": null, |
|  |  | "metadata": {}, |
|  |  | "outputs": [], |
|  |  | "source": [] |
|  |  | } |
|  |  | ], |
|  |  | "metadata": { |
|  |  | "kernelspec": { |
|  |  | "display\_name": "Python 3", |
|  |  | "language": "python", |
|  |  | "name": "python3" |
|  |  | }, |
|  |  | "language\_info": { |
|  |  | "codemirror\_mode": { |
|  |  | "name": "ipython", |
|  |  | "version": 3 |
|  |  | }, |
|  |  | "file\_extension": ".py", |
|  |  | "mimetype": "text/x-python", |
|  |  | "name": "python", |
|  |  | "nbconvert\_exporter": "python", |
|  |  | "pygments\_lexer": "ipython3", |
|  |  | "version": "3.6.4" |
|  |  | } |
|  |  | }, |
|  |  | "nbformat": 4, |
|  |  | "nbformat\_minor": 2 |
|  |  | } |

**0 comments on commit f82e8b4**

Please [sign in](https://github.com/login?return_to=https%3A%2F%2Fgithub.com%2Fakshaybahadur21%2FDrowsiness_Detection%2Fcommit%2Ff82e8b47c48fde9d9b1702f4b7cc5ecba8d99ffe) to comment.

**Footer**

© 2023 GitHub, Inc.

**Footer navigation**

* [Terms](https://docs.github.com/site-policy/github-terms/github-terms-of-service)
* [Privacy](https://docs.github.com/site-policy/privacy-policies/github-privacy-statement)
* [Security](https://github.com/security)
* [Status](https://www.githubstatus.com/)
* [Docs](https://docs.github.com/)
* [Contact GitHub](https://support.github.com/?tags=dotcom-footer)
* [Pricing](https://github.com/pricing)
* [API](https://docs.github.com/)
* [Training](https://services.github.com/)
* [Blog](https://github.blog/)
* [About](https://github.com/about)