SCHOOL OF COMPUTER SCIENCE
AND ENGINEERING

Test (Weightage 10%)

AUGUST 2021 SEMESTER

MODULE NAME	**: COMPUTER VISION & NLP**
MODULE CODE	**: ITS69204**
DATE/TIME	**: 19th of October 2021, 12:00 PM (MYT)**
DURATION	**: 1 HOUR**
PLATFORM	**: TIMeS**

This paper consist of THREE (3) pages, inclusive of this page

Student Name	Pong Kien Yiep	**Score out of 30**	
Student ID	0341541	**Score out 10**	

Question 1

The eye tracker has different setup. One of the setups is head-boxed setup, which can allow the user to move his or her head. This eye tracker can report the gaze coordination, but the head need to stay within the range of the head box area. One of the challenges is even though the head can be freely moved but some movement is still quite restrictive. For example, the user need to sit as still as possible with the eye tracker placed in front of the users.

Besides of the head-boxed setup, the eye tracker also provide the head-restricted setup. The challenge of this setup is the head cannot be moved and need to be fixed in a place. If the head is moved, it will cause the eye tracker hard to detect the pupil and the CR and lose the signal of the eye. Besides that, some user will sometime move their eye unintentionally, which will cause the unwanted result of the detection.

Another challenge is the pose, the eye tracker is sensitive to the pose variations. If the head movement or the viewing angle change, it will cause the rate of the eye detection drop. Especially when the rotation is high, it will be more challenge to identify the coordinate of the eye, which will result in the false eye detection.

Moreover, another challenge is the low resolution. The picture with the resolution less than 16 x 16 is considered low resolution. The low resolution does not provide much information as most of the image details will be loss, which will cause the eye detection rate decrease.

More than that, the illumination can also affect the eye detection. This is because, illumination will change the face appearance, and one side of the face will seem darker, which can affect the rate of the eye detection if some part of the image is dark and hard to be predicted by the eye tracker.

Another challenge is the eye tracker will hard to perform detection if the user does not face the camera, which make it impossible for the camera to catch the coordination of the eye. Hence, the eye tracker cannot track the user through the CCTV, and ATM camera, since most of the user will not directly face the camera of CCTV and ATM. Besides that, it will also be hard to perform detection on the user who has long eye lashes or wearing contact lenses.

Lastly, the ageing can also affect the eye detection rate. With the increase age of the human, the face feature, shape will be changes, and the wrinkle, mark, eyebrow, will be changed follow by the increase of the age. Since the eye tracker will also depends on the features mentioned, hence if the eye tracker cannot match the face feature, it will reduce the eye detection rate.

Question 2

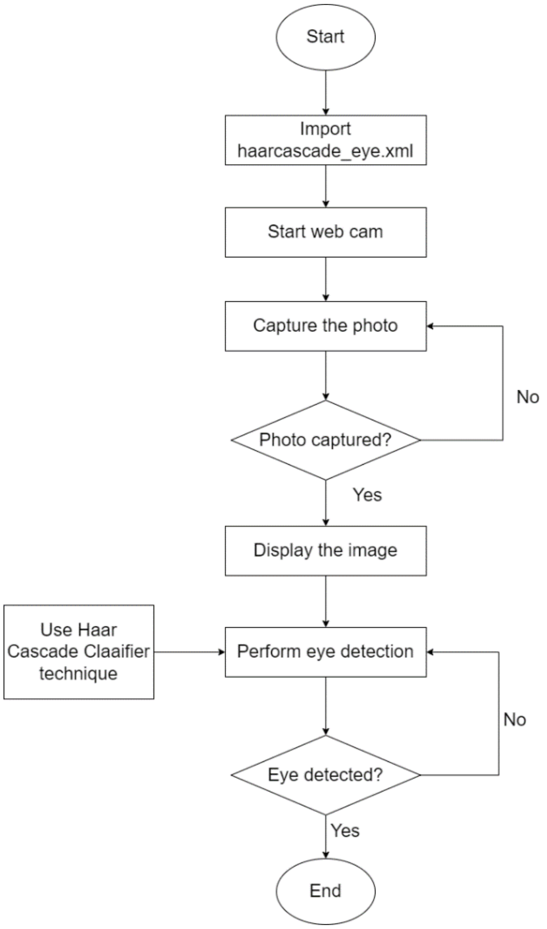
There are two important concepts to analyse the eye movement, which are fixation and saccades. The fixation is the focus of the eye vision on an object. Saccade is the fast movement of the eye, when the eye focus on different object, which also can be defined the time interval the fixations. The parameter of eye movement will be collected and recorded based on the fixed coordinate of the eye tracker. To ensure the raw data collected is reliable, the four restriction, such as z axis coordinate, pupil diameter and eye relative position need to have valid record. If any of these four restrictions is not satisfied, the eyeball will not be tracked correctly.

To ensure the data pre-processing can perform well. The practical sampling frequency need to be checked well and perform the necessary correction. If the practical sampling frequency fluctuate up and down, it mean the number of records is not consistent in each second, which will cause the data pre-processing is hard to be processed.

After the practical sampling frequency is checked, the eye movement data will be selected to identify the visual fixations. The eye movement data is obtained from either left or right eye. The movement data will be based on the pupil diameter, eyeball position, gaze position, and the record validity of left and right eye. The average eye movement data will be calculated for both eyes. If the record for left or right eye is valid, then the eye movement data will be only obtained from the valid tracked eye, but if both left and right eye records are not valid, the eye movement data will be set to 0.

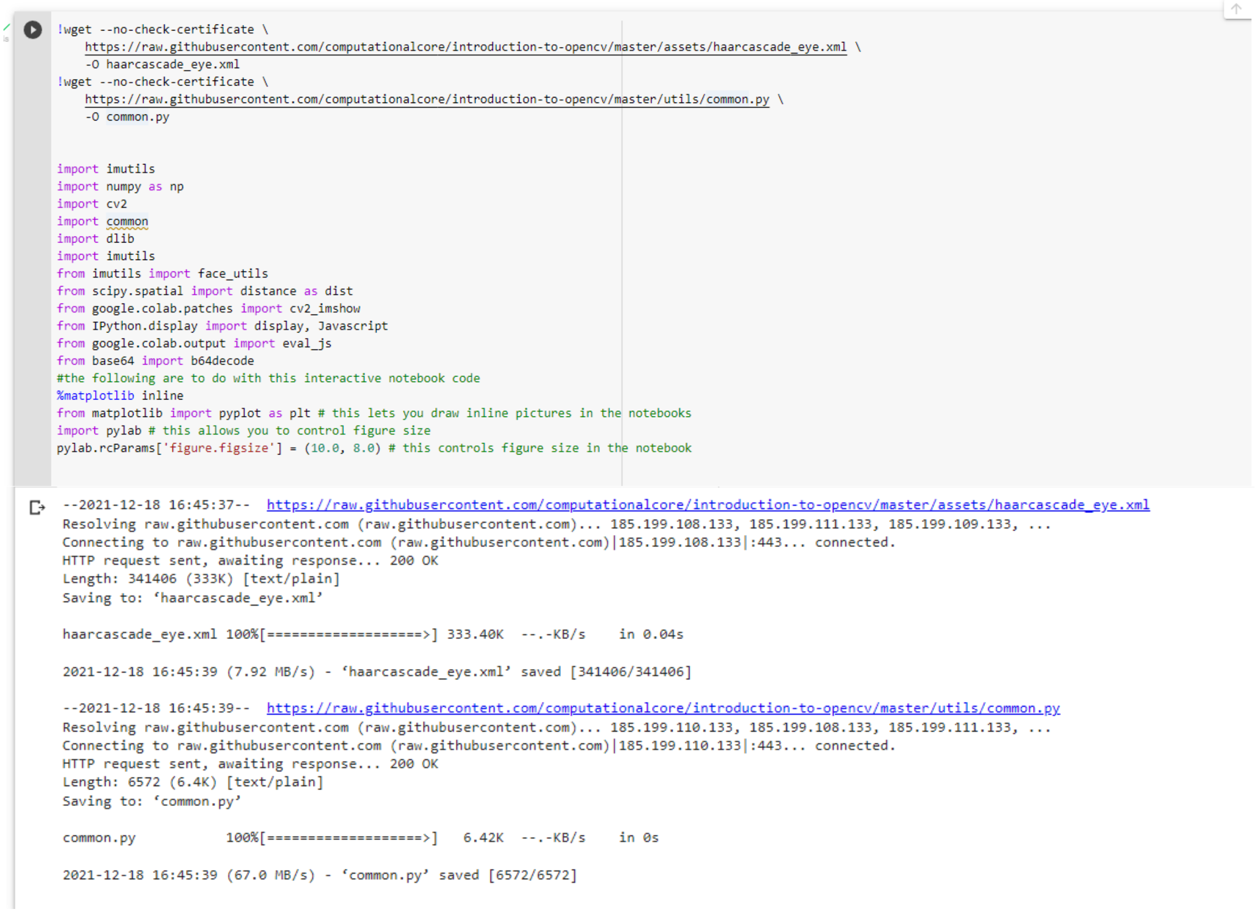
After the eye movement data is calculated, the eye tracker can ensure the accurate tracking, but it can bring noises at the same time. Hence the data filtering or smoothing process will be necessary carried out to remove the noises. After the process of data filtering, it will start to draw the visual fixations from the raw eye movement data. The different type of methods will be used to identify the fixation point. The fixation point will be drawn from the eye movement data of left eye and right eye. The result of the fixation point will show the identification strength of both eye, and then compare the strength of both eye to decide which eye is the best to carry the identification,

Question 3a



To perform the eye detection, first the haarcascade_eye xml will need to import inside the google colab environment. After that, the user can start the web cam, and then capture the photo. If the photo is captured successfully the photo will be displayed but if the photo is captured fail, then the user will need to capture the photo again. After displaying the image, the system can start to perform the eye detection by using the Haar Cascade classifier technique. If the eye is not detected, then the user will need to perform the detection again but if the eye is detected, then the process will end.

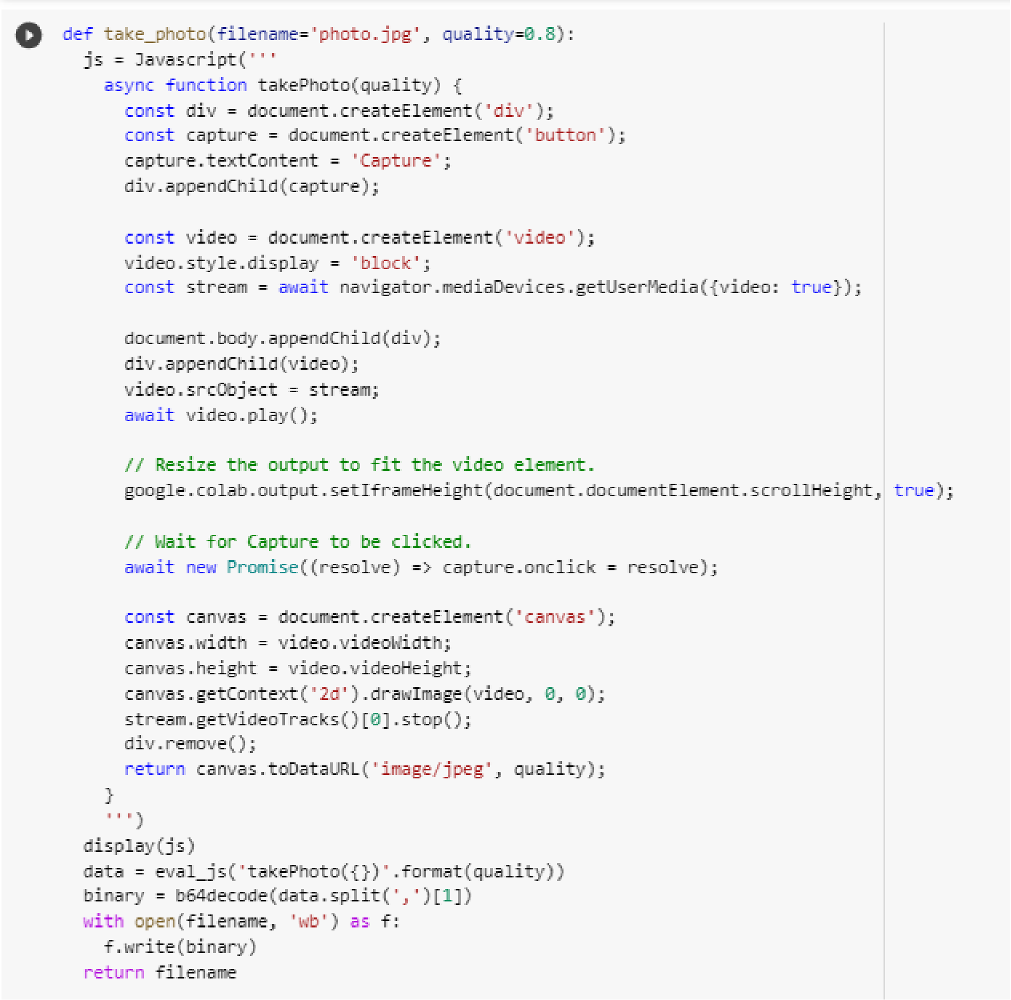
Question 3b



```
!wget --no-check-certificate \
    https://raw.githubusercontent.com/computationalcore/introduction-to-opencv/master/assets/haarcascade_eye.xml \
    -O haarcascade_eye.xml
!wget --no-check-certificate \
    https://raw.githubusercontent.com/computationalcore/introduction-to-opencv/master/utils/common.py \
    -O common.py


import imutils
import numpy as np
import cv2
import common
import dlib
import imutils
from imutils import face_utils
from scipy.spatial import distance as dist
from google.colab.patches import cv2_imshow
from IPython.display import display, Javascript
from google.colab.output import eval_js
from base64 import b64decode
#the following are to do with this interactive notebook code
%matplotlib inline
from matplotlib import pyplot as plt # this lets you draw inline pictures in the notebooks
import pylab # this allows you to control figure size
pylab.rcParams['figure.figsize'] = (10.0, 8.0) # this controls figure size in the notebook
```

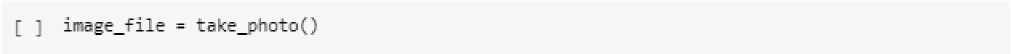
```
--2021-12-18 16:45:37--  https://raw.githubusercontent.com/computationalcore/introduction-to-opencv/master/assets/haarcascade_eye.xml
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.111.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 341406 (333K) [text/plain]
Saving to: ‘haarcascade_eye.xml’

haarcascade_eye.xml 100%[===================>] 333.40K  --.-KB/s    in 0.04s

2021-12-18 16:45:39 (7.92 MB/s) - ‘haarcascade_eye.xml’ saved [341406/341406]

--2021-12-18 16:45:39--  https://raw.githubusercontent.com/computationalcore/introduction-to-opencv/master/utils/common.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.108.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6572 (6.4K) [text/plain]
Saving to: ‘common.py’

common.py           100%[===================>]   6.42K  --.-KB/s    in 0s

2021-12-18 16:45:39 (67.0 MB/s) - ‘common.py’ saved [6572/6572]
```

Based on the code above, it basically import all the files and tools which will be used to perform the eye detection. The haarcascade_eye.xml which is a predefined model will be imported from the github.



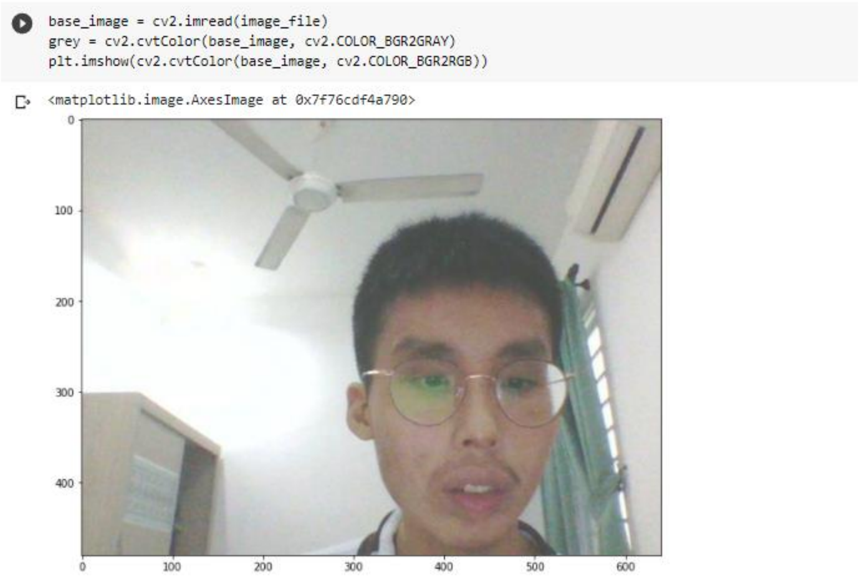
```python
def take_photo(filename='photo.jpg', quality=0.8):
  js = Javascript('''
    async function takePhoto(quality) {
      const div = document.createElement('div');
      const capture = document.createElement('button');
      capture.textContent = 'Capture';
      div.appendChild(capture);

      const video = document.createElement('video');
      video.style.display = 'block';
      const stream = await navigator.mediaDevices.getUserMedia({video: true});

      document.body.appendChild(div);
      div.appendChild(video);
      video.srcObject = stream;
      await video.play();

      // Resize the output to fit the video element.
      google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);

      // Wait for Capture to be clicked.
      await new Promise((resolve) => capture.onclick = resolve);

      const canvas = document.createElement('canvas');
      canvas.width = video.videoWidth;
      canvas.height = video.videoHeight;
      canvas.getContext('2d').drawImage(video, 0, 0);
      stream.getVideoTracks()[0].stop();
      div.remove();
      return canvas.toDataURL('image/jpeg', quality);
    }
    ''')
  display(js)
  data = eval_js('takePhoto({})'.format(quality))
  binary = b64decode(data.split(',')[1])
  with open(filename, 'wb') as f:
    f.write(binary)
  return filename
```

The code displayed at the image above will basically allow the user to start the web cam.



```python
image_file = take_photo()
```

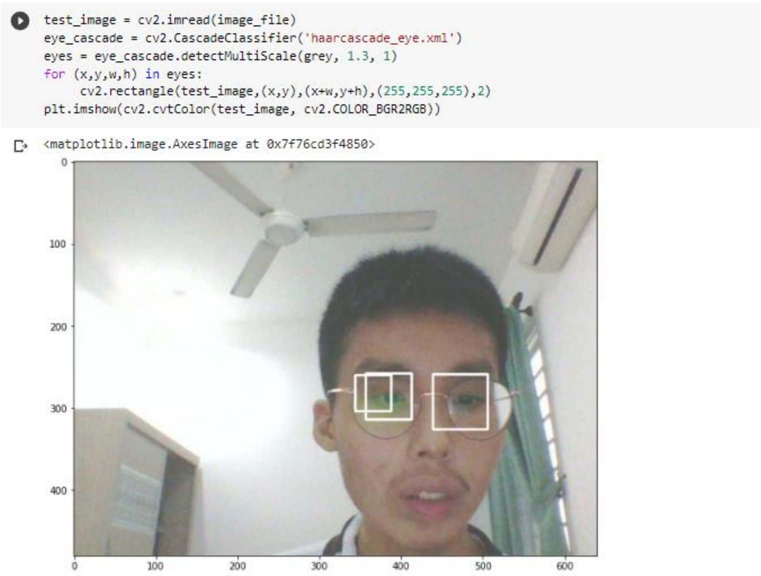
The code displayed above, will allow the user to capture the photo using webcam.



```
base_image = cv2.imread(image_file)
grey = cv2.cvtColor(base_image, cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(base_image, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7f76cdf4a790>
```

After capture the photo, the photo will be read and resized, and then the photo will be displayed



```
test_image = cv2.imread(image_file)
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')
eyes = eye_cascade.detectMultiScale(grey, 1.3, 1)
for (x,y,w,h) in eyes:
     cv2.rectangle(test_image,(x,y),(x+w,y+h),(255,255,255),2)
plt.imshow(cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7f76cd3f4850>
```

After displaying the photo, we will use the haarcascade_eye.xml which is a pre trained model to detect the eye in the picture.

Question 3c

The main technology used to perform the eye detection is the Haar Cascade classifier technique. Normally, the feature alone will produce a weak classifier, and it will need at least 6000 features to produce a good classifier. However, it can take too much time and little efficient to run through the 6000 features. Hence, the Haar Cascade classifier technique is introduced. Instead of using all the 6000 features. The features will be grouped into different classifier stages and then each stage will be applied one by one. If the first stages is failed, then the rest stages will be discarded. If the first stage is passed, the second stages will be applied, and the process will continue. The 6000 features will be divided to 38 stages, with 1, 10 ,25, and 50 features in each stage. Hence, the Haar Cascade classifier is a lot faster than running through the 6000 features at once.

Besides that, the Haar Cascade classifier can also provide the ability to detect the face. It will provide the high-speed computation which depend on the number of pixels in the image but will not depend on the pixels value of the image. The Haar Cascade classifier has 4 methods. One of the methods is called Haar feature, which is used to detect the presence of feature in an image hence the Haar Cascade classifier is a good classifier to perform to eye detection.

Question 4

One of the security risks of the Alexa is it has the camera. The newest technology of the Alexa is the implementation of the camera, which can basically capture every picture of your lifestyle. The privacy will be affected if the Alexa camera capture anything which the user don't want to let other people to view on it.

Another security concern is the Alexa will basically record what you are saying. It could violate the user privacy when the user is having an important discussion and the discussion is recorded by the Alexa and stored inside the server. Besides that, anyone can read through what the user recently saying through the user phone to know, and gain insight what the user is interested about and the secret conversation of the user. Since the Alexa is listening all the time unless the user mute it, hence it has the high possibility that the Alexa record the important discussion of the user when the user forgot to mute it, and most of the time the users are very busy and does not pay much focus on muting the Alexa.

The worst-case scenario is the Alexa is able to be hacked by the hacker. The hacker is able to take over the other user's Alexa device and then use the microphone and camera for malicious intent, which mean the hacker can hear the conversation of the user and also know what the users are doing by accessing the camera, which already affect the privacy of the user and cause danger if the hacker is listening the user is leaving their house, which can give the opportunity for the hacker to enter the user's house, without any attention. Besides that, it also happen before that the user data in the Alexa is leaked. Even though the Alexa does not record the login credentials, but it record what user has said. Hence, it gives the opportunity for the hacker to access the chat history to steal the important data like username, password or even the bank details.

Question 5

To ensure the security, the data should not be stored in the Alexa device, so that it can prevent the hacker to hack the data and misuse the data. Besides that, Alexa need to be designed in a way that only some instruction can be heard but not all conversation can be heard. Besides that, cloud service in the Amazon should provide multiple level of security, which can prevent the system to be hacked by the user. The IT department should also implement the rule that the data need to be restricted to be sent to other third party.

Besides that, the Alexa should also allow the duration of listening the instruction. For example, the user can set the Alexa to listen for the instruction for 1 hour, and after 1 hour, the Alexa will be automatically turned off without continue listening to any instruction. Instead of limiting the duration for listening, the duration of turning on the Alexa camera should also be limited and turned off as well after certain amount of duration time.

The Amazon should take full responsibility for any data breach as this will prove the accountability of the Amazon, which will gain the trust from the customers, and keep the Amazon on the right track to improve the platform and ensure the data security in the future.

Appendix

Source code

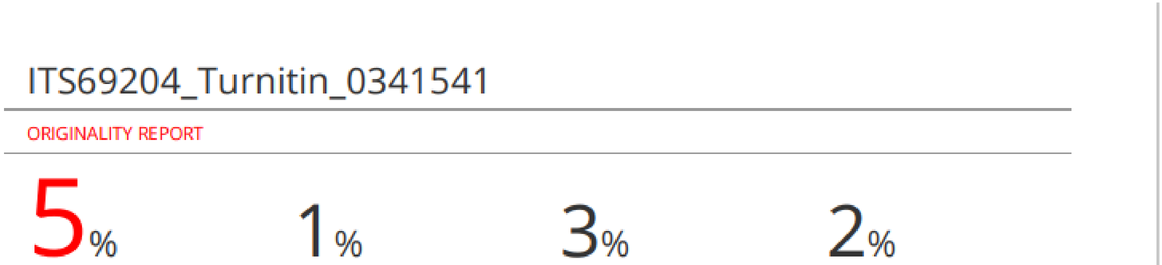
The google drive link to download the source code ipynb file.

https://drive.google.com/file/d/1IHlds3emZujHvETFqoN2nCGjmnSjzLrP/view?usp=sharing

References

1) Docs.opencv.org. 2002. *OpenCV: Cascade Classifier*. [online] Available at: <https://docs.opencv.org/4.x/db/d28/tutorial_cascade_classifier.html> [Accessed 18 December 2021].
2) Doffman, Z., 2020. *Why You Must Beware What You Ask Amazon Alexa*. [online] Forbes. Available at: <https://www.forbes.com/sites/zakdoffman/2020/08/13/amazon-alexa-cyber-attack-check-point-report-smart-speaker-warning/?sh=295aead05008> [Accessed 18 December 2021].
3) Kamarudin, N., 2019. [online] Hrpub.org. Available at: <https://www.hrpub.org/download/20191230/UJEEEB9-14990984.pdf> [Accessed 18 December 2021].
4) Liu, B., Zhao, Q. and Ren, Y., 2018. *An elaborate algorithm for automatic processing of eye movement data and identifying fixations in eye-tracking experiments - Bo Liu, Qi-Chao Zhao, Yuan-Yuan Ren, Qing-Ju Wang, Xue-Lian Zheng, 2018*. [online] SAGE Journals. Available at: <https://journals.sagepub.com/doi/10.1177/1687814018773678> [Accessed 18 December 2021].
5) Stegner, B., 2018. *7 Ways Alexa and Amazon Echo Pose a Privacy Risk*. [online] MUO. Available at: <https://www.makeuseof.com/tag/alexa-amazon-echo-privacy-risk/> [Accessed 18 December 2021].

Turnitin



ITS69204_Turnitin_0341541

ORIGINALITY REPORT

5% 1% 3% 2%