Kian Parnis (0107601L)

ICS 2207 Assignment Documentation

Contents

Kian Parnis (0107601L)

Outline of file contents

- **Artifact1-tests-|**
- **|-Contains the images used for testing**
- **Artifact2-tests-|**
- **|-Contains two pre recordings testing the second artifact**
- **Cascades-|**
- **|-Contains all stages trained for Artifact one**
- **Artifact1-|**
- **|-Python script which runs the first artifact**
- **Artifact2-|**
- **|-Python script which runs the second artifact (camera needed)**

Below are two folders which surpassed the 100mb limit, and are stored in google drive:
https://drive.google.com/drive/folders/1jFxFerawpD5LYAAlLGNfu31iyeAWSN42?usp=sharing

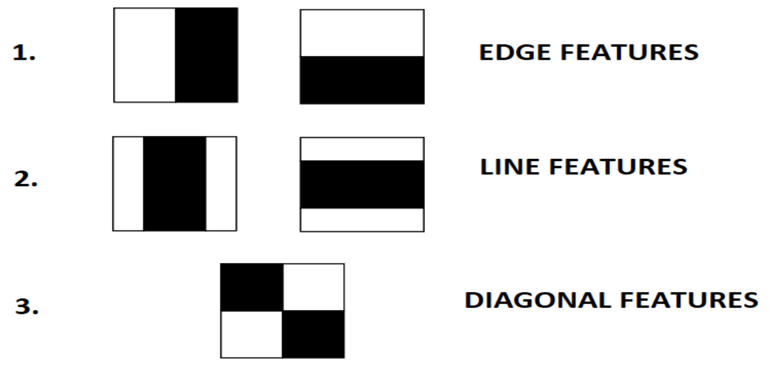
- **Opencv-| { opencv\build\x64\vc15\bin }**
- **|- Training tools used to annotate and train the cascade classifier**
- **TrainingFile-|**
- **|-neg.txt, pos.txt and pos.vec used for training**
- **|-negative-|**
- **| |-All negative images used for training**
- **|-positive-|**
- **|-All positive images used for training**

Viola-Jones Object Detection/Haar Cascades

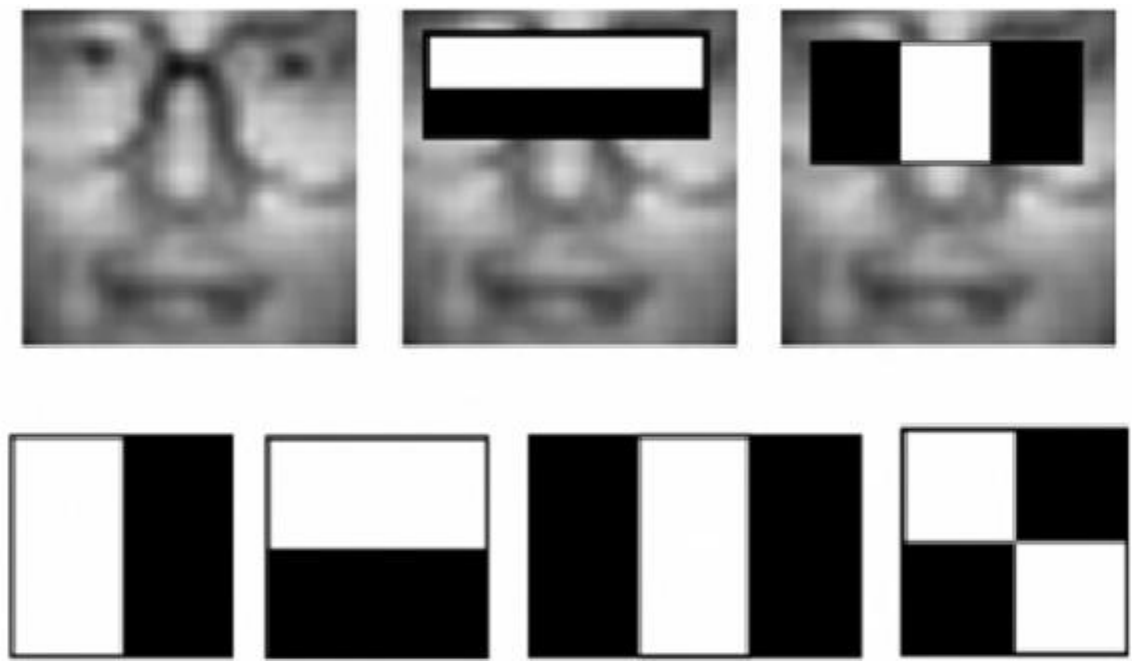
Face detection has several applications in our real-life world. Security cameras use algorithms such as the Viola-Jones to detect faces and recognize the difference between a person from a potential stranger. One of the applications that this algorithm has is the ability to apply filters on faces, another would be that it can be used for security in phones to auto-unlock a face when it is detected. This happens in real time and is thanks to Paul Viola and Michael Jones who published a paper coined “Rapid object detection using a boosted cascade of simple faces” back in 2001 [1]. This algorithm although simplistic is still used today with large apps such as Snapchat making a use for it in its filters today [2].

The algorithm itself makes use of frontal face images where the features of faces are used to detect what is a face and what isn’t. The breakdown of the algorithm is set at four stages. The first step is that of detecting **Haar Features**, which are used to detect the features of a face and an **Integral image** is calculated to fasten the processing. **Adaboost** Training is then followed to properly locate and train the features and enhance processing time. Finally, a **cascade classifier** is used to distinguish whether a window contains a face or not. These steps will now be broken down and explained in a more thorough manner.

Starting from Haar Feature Selection, a Haar Feature is a rectangular region that masks over an image. Within each rectangle the summation of pixels is calculated and then the difference between the black and white regions is then calculated. These features are classified into three basic groups: edge features, line features and four-rectangle features.



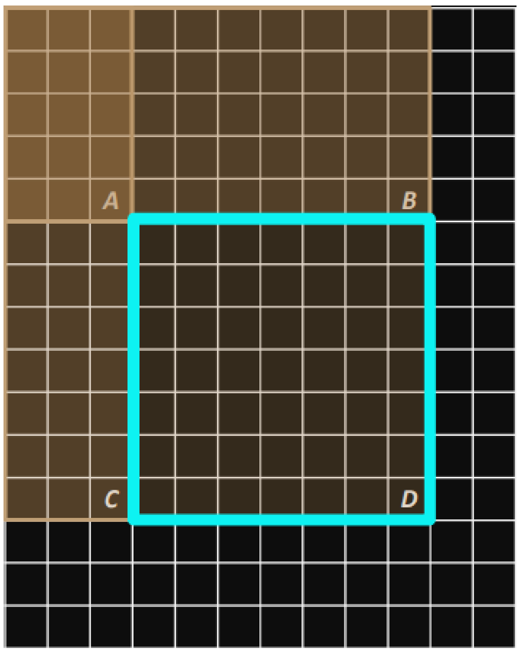
These features are the used to locate certain objects in an image, for example a line feature can be used to detect the location of an eye based on the color difference between the eye and eyebrow. For this to be done however the image itself would have to first be converted to grayscale so that the distinction between the lighter and darker areas can be properly identified due to the values taken being based solely on the value of intensity between light and dark in a pixel selection.



However due to the number of different image features differentiating from one to another (example one nose from another) the detection of a Haar Feature would have to be calculated in the thousands to correctly identify a face feature and these images would be classified into variations of the basic groups of features. Due to the large complexity that would be present from calculating all these different sums, the integral image was developed by Pual Viola and Michael Jones to better fasten the process.

The Integral Image or Summed Area Table is used to better calculate the sum of values in a part of an image (normally taken as a rectangular area). Within this area any point(pixel) taken would contain a value and the Summed Area table would take each point and sum it with the value itself, the value positioned right above it, the values positioned to the left of it and finally we subtract the value to the top-left of the value or more formally s(x,y)=i(x,y)+s(x-1,y) + s(x, y-1) – s(x-1, y-1) [3].

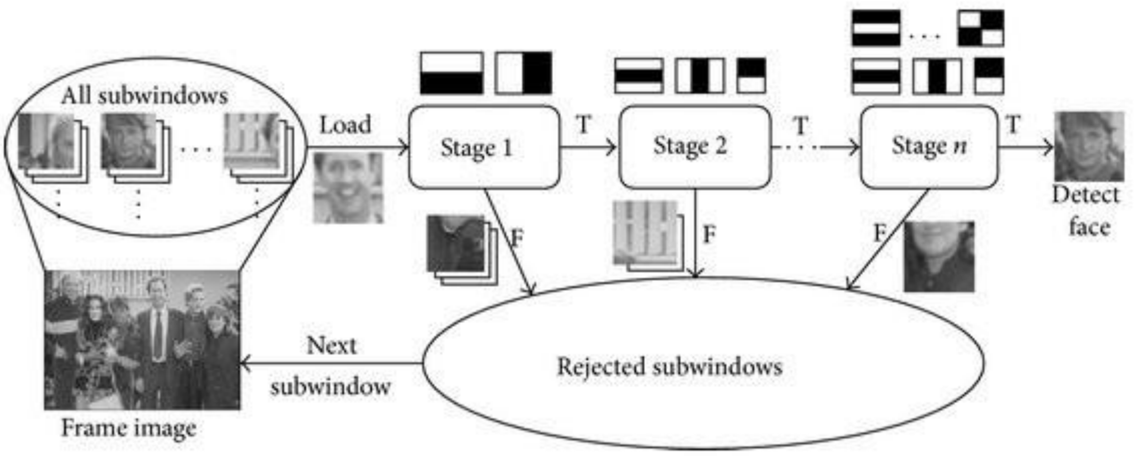
For example, an image subset $\begin{vmatrix} 10 & 5 & 25 \\ 88 & 64 & 35 \\ 2 & 45 & 78 \end{vmatrix}$ this, would result in integral image $\begin{vmatrix} 10 & 15 & 40 \\ 98 & 167 & 227 \\ 100 & 214 & 352 \end{vmatrix}$. With the Summed Area Table filled in the sum of pixels within the rectangle becomes that of constant time O(1) and to achieve area a reference to four locations A,B,C,D need to be kept and making use of the equation s(A) + S(D) – S(B) – S(C) the area can be quickly calculated.



So, for the area of $\begin{vmatrix} & 167 & 227 \\ & 214 & 252 \end{vmatrix}$ the formula would translate to 352 – 100 – 40 + 10 which gives a result of 222 which corresponds to the addition of area $\begin{vmatrix} & 64 & 35 \\ & 45 & 78 \end{vmatrix}$ which also results in area 222. The significance of this would be that taking an image that's for example 100x100 after calculating the integral area once, calculating the area of specific regions would take 4 operations instead of 10,000 and this would greatly reduce the time taken to calculate dozens of Haar Features.

When calculating Haar Features there would potentially be many features which are calculated that aren't of useful information, therefore Ada-boost training is used to eliminate these undesired features to get accurate results much faster and construct a classifier. For this algorithm to work two large set of images need to be prepared, one which contains positive images (faces) and another which contains negative images (non-faces) in terms of weighted error.

After the Ada-boost training is completed a set of features are identified, a simple classifier is chosen based on image’s sub-window which first accepts many positive images and rejects many of the negative images. After a slightly more complex classifier is chosen which again accepts positives and rejects negatives and so on. This is done on all sub-windows till a face is detected. A good classifier is one that can detect a face with a with a very low false-positive rate and a high true-positive rate. [4]



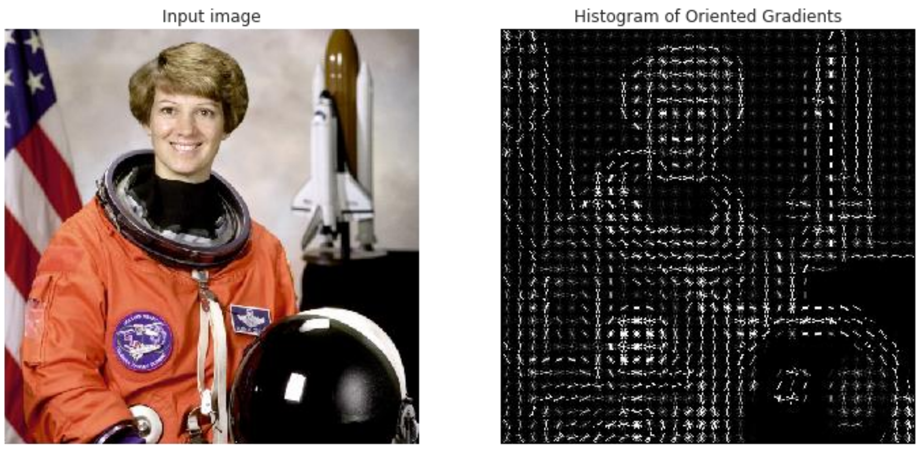
In summary the viola jones algorithm will take a grayscale image, perform the integrated image, open a window and slide it across the image and generate sub-windows for every sub-window the cascade classification will be done with previously obtained features. If the window passes all the stages of the cascade classification are accepted, then a face will be detected, and a new window will be studied until no new windows can be opened.

Discuss why they are especially useful for real-time applications such as face detection in digital cameras

Viola-Jones are especially useful for facial recognition in faces for several reasons, first being that the algorithm itself has a rather high accuracy when detecting faces if trained well with very high true positives and a very low false positive rate. Secondly since its face detection and not facial recognition, the Viola Jones algorithm is designed around the premise of detection face from non faces so this provides a very good starting use case. Lastly since the cascade classifier takes subframes and calculates the true-positives based on those sub-frames, viola jones can be paired up with a tracking algorithm such as the KLT algorithm to initially detect a face and track its movements to provide a low-latency detection on the where the face moves to. [5]

Alternative to Viola Jones

One alternative of the Viola Jones algorithm is the Histogram of Oriented Gradients method, which is also used for face detection in images. Whereas Viola Jones uses Haar Casacades to map features on faces, HOG on the other hand is based on feature descriptors which extracts useful information present in an image and discards any unnecessary information present. As noted in [10] the HOG algorithm tends to be more accurate than that of Viola-Jones and doesn't struggle to detect faces which have accessories such as eyeglasses like the viola jones does.



Methodology

Artifact 1

When developing artifact one the first consideration was the dataset size, since the approach taken was one which had to manually create bounding boxes around faces, a reasonability sized dataset which contains roughly a thousand pos/neg where to be found, any less would heavily deteriorate the detection rate of the cascade. Another consideration when picking the dataset was that of populations, the dataset had to be diverse covering different races, age groups and gender so no specific population would be omitted from the detection of the algorithm.

The final dataset was chosen with around a thousand positive images and four thousand negatives and managed to cover the population size mentioned prior. This dataset can be found here [6]. When starting the process of training the cascade classifier a guide was followed which helped with the setup of the tools used and creation of the text files used by the trainer [7]. The tools mentioned are OpenCV's integrated annotation tool which lets you manually set the bounding boxes for each of the positive images [8]. A negative text file was generated to be used by the tool when training the cascade classifier with the location of each image inside the file. To generate the text positive text files the program *opencv_annotation.exe* was used to generate the positive text file with each images bounding box coordinates. *opencv_createsamples.exe* was then used to create a vector file which contains the samples of all the positive images used. After these files, the training process itself could start, a folder 'cascades' was created that housed each stage of the training.

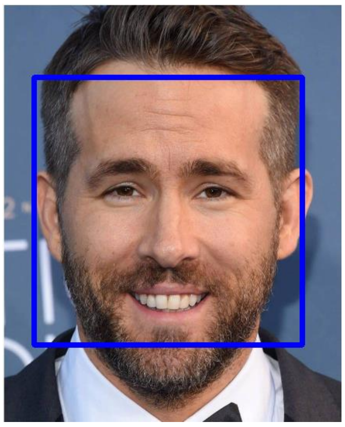
To do the actual training *opencv_traincascade.exe* was used which took the files generated as arguments *{pos.vec, neg.txt}* and the number of stages for the trainer to take at first a high stage count was taken, and testing was done to see the result. At first the result wasn't satisfactory, so the number of stages was changed several times until a satisfactory result was give from the testing. The final stage count for this cascade was 14 stages. A simple python script 'Artifact1' was created to test the accuracy of each cascade generated which takes an image and coverts it to grayscale for viola jones to function correctly.

Rectangles for where the faces are estimated to be are then drawn and the result is shown to the user.

Testing

The following are the outcomes when testing the cascade classifier with images which are not present in the training set:

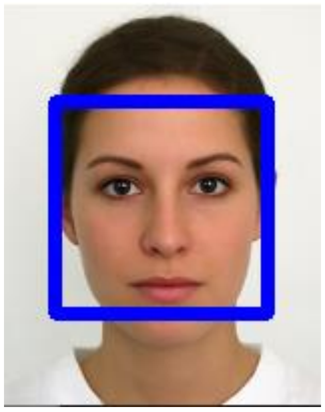
Frist test case



Expected: Correct bounding box around the person was created.

Actual: Bounding boxed created successfully

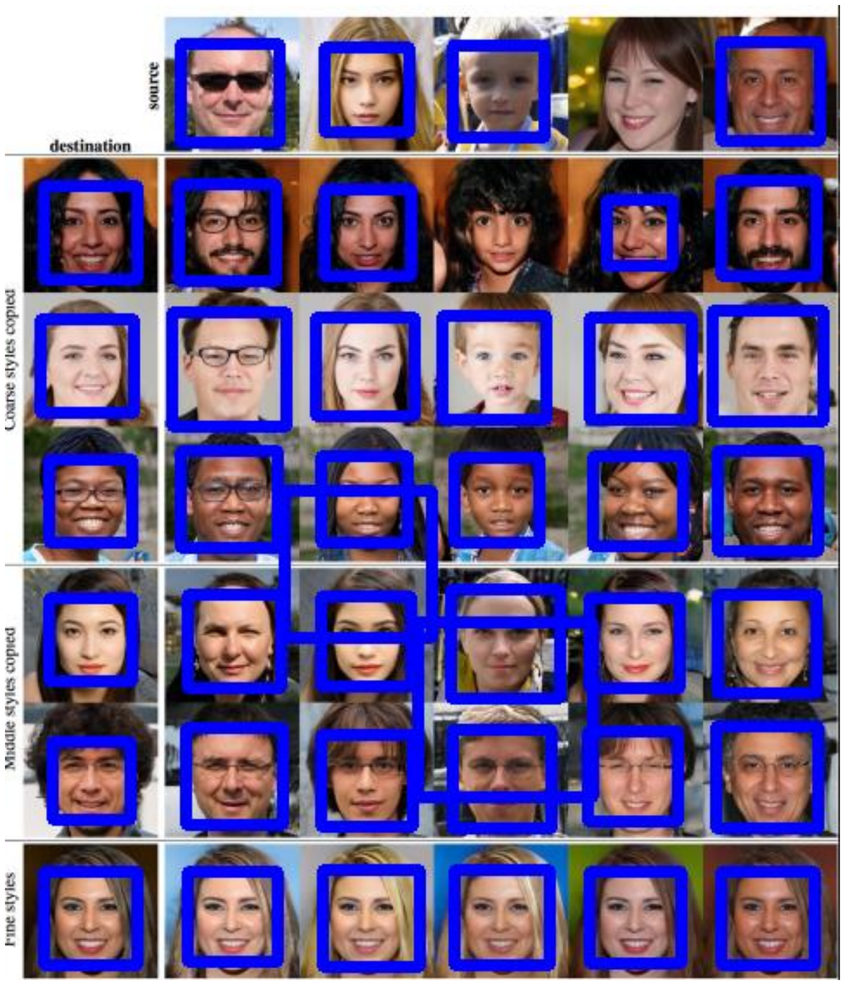
Second test case



Expected: Correct bounding box around the person was created.

Actual: Bounding boxed created successfully

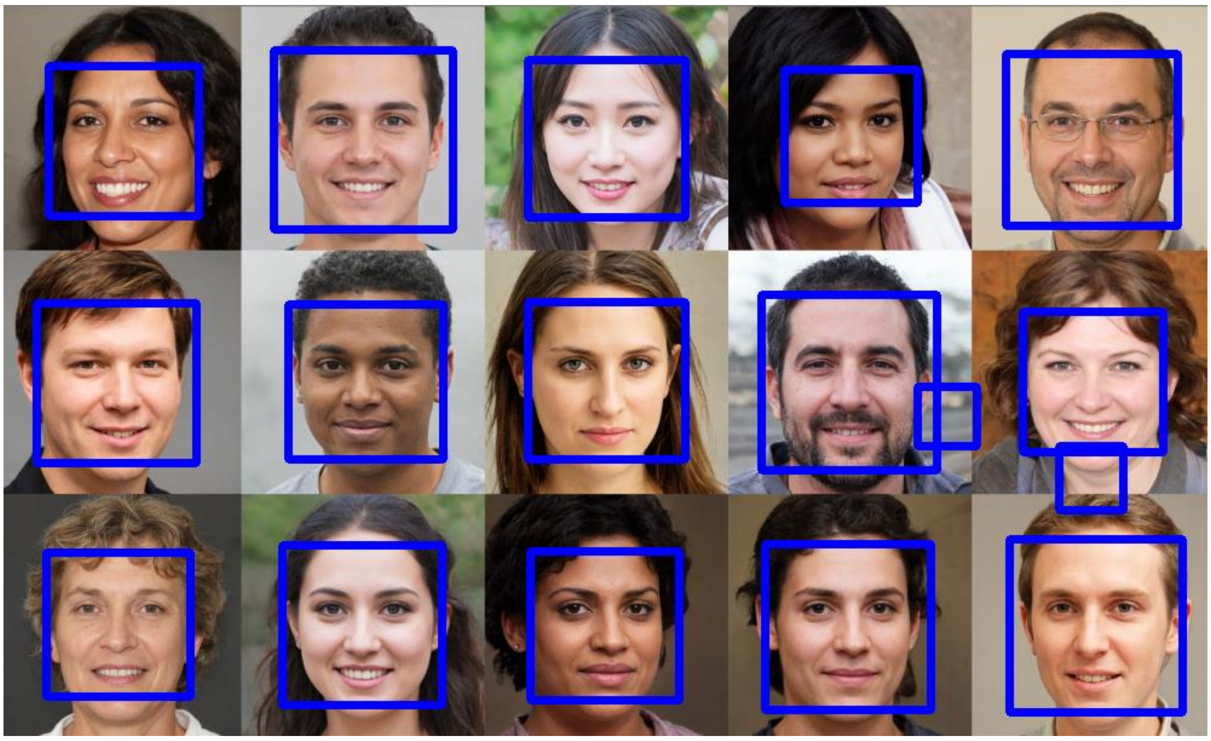
Third test case



Expected: Correct bounding box around each person was created.

Actual: A high percentage of faces where matched successfully, each of different race and age. To specific faces where not detected and a few note worthy errors are also present.

Forth test case



Expected: Correct bounding box around each person was created.

Actual: Each person was successfully detected by the algorithm with minor incorrect errors detected.

Limitations

Due to the size of training being limited in size the detection isn't 100% accurate. Given more time a larger set can be used with the same methodology to give a higher accurate detection rate.

Artifact 2

Task two was implemented using a user's camera to perform real time face and eye tracking. The cascades used to perform this detection can be found here [9] and with the use of the opencv's package two cascades '*haarcascade_frontalface_default.xml'* and '*haarcascade_eye.xml'* where used to draw rectangles around a persons face and eyes respectively. Faces are clearly outlined by a green triangle while eyes are outlined in red. A guide was followed when the mapping of the rectangles on the eyes and faces where programmed and this can be found here [11]. To load the second artifact simply click on the 'Artifact2' script and the detection will start. If you want to exit the program kindly press the key 'w' and the window will close.

Testing

Two videos are present within *Artifact2-tests* labeled 'low-light' and 'too-much-light' which were taken in different lighting conditions to see how the accuracy of the cascade varies. It was noted that on average they both had high accuracy at correctly identifying faces. It is noted however that in low light the classifier has a slightly harder time at correctly mapping the eyes correctly. This result is as expected due to the Viola jones algorithm using Haar Features for detection. Since a much smaller difference between the dark and light pixels is present in low light this process of feature detection will struggle to correctly identify distinctions to correctly map a haar feature. In *low-light* a test is also made for when glasses are wore vs taken off, the face detector managed to consistently identity a face during the process, while the eye detector had a far higher detection rate when the glasses where taken off. Lastly, as expected, the cascade classifier works best at front face detection, any slight shifts in a direction would render a face unidentified.

Statement of completion

Statement of completion – MUST be included in your report

Item	**Completed (Yes/No/Partial)**
Artefact 1	Yes
Artefact 2	Yes
Artefact 2 – real time face/eye tracking	Yes
Technical discussion on Viola-Jones method	Yes
Experiments and evaluation	Yes
If partial, explain what has been done	

References

[1] *Cs.cmu.edu*, 2022. [Online]. Available: https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf. [Accessed: 20- Jan- 2022]

[2] "The Inner Workings Of Snapchat's Faceswap Technology", *Forbes*, 2022. [Online]. Available: https://www.forbes.com/sites/quora/2017/03/17/the-inner-workings-of-snapchats-faceswap-technology/?sh=10a24fa564c7. [Accessed: 20- Jan- 2022]

[3] "Computer Vision – The Integral Image", *Computer Science: Source*, 2022. [Online]. Available: https://computersciencesource.wordpress.com/2010/09/03/computer-vision-the-integral-image/. [Accessed: 20- Jan- 2022]

[4] *Courses.cs.washington.edu*, 2022. [Online]. Available: https://courses.cs.washington.edu/courses/cse455/16wi/notes/15_FaceDetection.pdf. [Accessed: 20- Jan- 2022]

[5] *Arpnjournals.org*, 2022. [Online]. Available: http://www.arpnjournals.org/jeas/research_papers/rp_2016/jeas_1216_5414.pdf. [Accessed: 20- Jan- 2022]

[6] "Natural Images", *Kaggle.com*, 2022. [Online]. Available: https://www.kaggle.com/prasunroy/natural-images. [Accessed: 20- Jan- 2022]

[7] *Youtube.com*, 2022. [Online]. Available: https://www.youtube.com/watch?v=XrCAvs9AePM&t=1359s. [Accessed: 20- Jan- 2022]

[8] "OpenCV: Installation in Windows", *Docs.opencv.org*, 2022. [Online]. Available: https://docs.opencv.org/3.4.11/d3/d52/tutorial_windows_install.html. [Accessed: 20- Jan- 2022]

[9] "opencv/data/haarcascades at master · opencv/opencv", *GitHub*, 2022. [Online]. Available: https://github.com/opencv/opencv/tree/master/data/haarcascades. [Accessed: 20- Jan- 2022]

[10] 2022. [Online]. Available: https://www.researchgate.net/publication/338847058_Comparison_of_Viola-Jones_Haar_Cascade_Classifier_and_Histogram_of_Oriented_Gradients_HOG_for_face_detection. [Accessed: 20- Jan- 2022]

[11] *Youtube.com*, 2022. [Online]. Available: https://www.youtube.com/watch?v=mPCZLOVTEc4&t=740s. [Accessed: 20- Jan- 2022]

Plagiarism Declaration Form

Plagiarism is defined as *"the unacknowledged use, as one's own, of work of another person, whether or not such work has been published, and as may be further elaborated in Faculty or University guidelines*" (University Assessment Regulations, 2009, Regulation 39 (b)(i), University of Malta).

I, the undersigned, declare that the report submitted is my work, except where acknowledged and referenced. I understand that the penalties for committing a breach of the regulations include loss of marks; cancellation of examination results; enforced suspension of studies; or expulsion from the degree programme.

Work submitted without this signed declaration will not be corrected, and will be given zero marks.

Kian Parnis	ICS2207	1/20/2022
Student's full name	**Study-unit code**	**Date of submission**

Title of submitted work: ICS 2207 - Assignment Submission - Kian Parnis

Student's Signature

[illegible]