Task Driven Low-Light Image Enhancement - Dark Image Face Detection

Steven Curtis (1657041)
School of Computer Science and Applied Mathematics
University of the Witwatersrand
Johannesburg, South Africa
1657041@students.wits.ac.za

Supervised by Dr Hairong Wang

***Index Terms*—Viola-Jones, face detection, low-light, enhancement, LIME, OpenCV**

***Abstract*—Face detection can be used for a variety of different tasks in everyday situations such as security surveillance. It can be, however, an exceedingly difficult task. Different factors such as pose and image quality affect the performance and ability to detect faces in a face detection model. The main issue with face detection is the illumination of the image being investigated. An image with a low illumination can cause a model's ability to detect faces in the image incredibly challenging. To address this problem this report proposes to first use a model to improve the low illumination of the images to be used in a face detection model to improve its accuracy. The low-light illumination model is based on the Low-light Image Enhancement (LIME) model which estimates the illumination map of images to improve their illumination. This model is tested on the LOw Light paired dataset (LOL) which contains paired images of low and normal illumination. The face detection model is based on Viola-Jones which uses Haar-like features to detect faces in the image. Experiments will show that the accuracy of this face detection model will improve when tested on images with improved illumination compared to images with low illumination.**

I. Introduction

Face detection is becoming increasingly more important and is being used for multiple different applications. These include unmanned aerial vehicles (UAVs), autonomous driving, security surveillance, search and rescue and many more. Images captured in good illumination are ideal for face detection. Face detection for images with low-light, however, can be very challenging and unable to detect faces accurately. The quality of images captured in a low-light environment would be significantly influenced by noise, low contrast and low detail, which impacts the face detection methods ability to perform accurately [1]. Many low-light image enhancement methods can improve the illumination of an image for face detection. This is however also a challenging task. Low-light enhancement techniques must "focus on restoring brightness and contrast and ignore the influences of noise" [2, p.1]. Improving the low-light illumination of an image may help face detection methods to run more accurately and therefore be more successful. There are many different methods for face detection. The one that we are implementing in this report is based on the Viola-Jones model [3], which is based on hand-crafted features like Haar [4]. A modified version of the ExDark dataset [5] is used for the face detection. The method we are using to solve the low-light image enhancement problem is the LIME model [6] which estimates the illumination map of images to improve their illumination. To test the best parameters to use in our implementation of the LIME model, we will use the LOw Light paired dataset (LOL) [7] which is a dataset that contains low and normal-light image pairs and see how LIME improves the illumination of the low-light images compared to that of the normal-light images by using mean squared error and peak signal-to-noise ratio (PSNR).

A. Problem Statement

A face detection method's ability to detect faces is greatly impacted in poorly illuminated images. This places a large restriction on the number of images that can be used for face detection. This problem also limits the capability of using face detection for most images that are captured at night or in poorly illuminated areas. This impacts many applications for face detection.

B. Significance of Research

Face detection has many applications. These include security, advertising, criminal identification, payment verification and healthcare. Many images that are vital to the performance of these applications, especially criminal identification, may be captured in poorly illuminated areas. This can have a major impact on the effectiveness of the application. Low-light image enhancement techniques such as LIME can be applied to these images to help improve the accuracy of face detection in the image. This can improve the effectiveness of many of these applications.

C. Research Aim and Objectives

1) Aim: The aim of this report is task-driven low-light image enhancement for face detection. This means that we are improving the low-light issue of images to improve the accuracy of detecting faces in them.

2) Research Question: A Low-light Image Enhancement (LIME) like model is used to perform the low-light image enhancement of the images fed into it and a Viola-Jones like model is used for face detection. Is there an improved accuracy in face detection when the illumination of the input image is first improved?

3) Objectives: The main objective is to use face detection on images that contain dark faces. The LOw Light paired dataset (low and normal-light image pairs) is used to obtain optimal parameters for the LIME model by comparing the enhanced low-light images to their normal-light pairs using mean squared error and the peak signal-to-noise ratio. Once these optimal parameters have been found, the images in the modified ExDark dataset, which contains dark faces, need to be enhanced. These improved illumination images are then fed into the face detection model for face detection and returns the images with a box around the faces detected. The number of detected faces is counted to find the accuracy of the model. This accuracy is compared to that of using this face detection model without first enhancing the input images. The improved accuracy of our face detection model when using input images of enhanced illumination solves the research question.

D. Report Structure

This problem of face detection in low-light images is solved using the LIME and Viola-Jones like model. These two models are researched and explained and comparisons to other models that could also accomplish the low-light enhancement and face detection are made. A detailed methodology is created using LIME and a model based on Viola-Jones as the two methods being used for low-light enhancement and face detection respectively. This shows the steps needed to implement and solve this research problem. A breakdown of the experimental setup, datasets used and the results obtained from the experiments are explained. Finally, a conclusion summarises what has been explained and proved through results obtained through experiments in the research report and show that the research question has been solved.

II. Background and Related Work

A. Introduction

The problem of solving the issue of low-light images for face detection can be solved using LIME for low-light image enhancement and a model based on Viola-Jones for face detection. LIME is one of the more popular methods for low-light image enhancement [8]. Viola-Jones is one of the most famous face detection models [9].

B. Low-light Image Enhancement

Many images are captured in low illuminated environments. These images contain poor visibility, low contrast and unexpected noise and are often unusable in most applications. Traditional methods such as histogram equalisation increase the contrast of images by stretching their dynamic range instead of looking at the illumination [2]. This causes the original bright regions to become saturated and you may lose the corresponding details [6]. We will therefore use a Retinex-based method, LIME, to enhance the contrast by estimating the illumination map. Retinex theory is the idea that images are the product of illumination and reflectance.

*1) **LIME Method**:* Low-light Image Enhancement (LIME) is a Retinex-based method that estimates the illumination of each pixel by looking for the maximum value in the red, green and blue section of the image [6], [10]. This is called estimating the illumination map. It then exploits the structure of the illumination and executes a structure-aware smoothing model which further refines the illumination map [2]. The method built upon the model, L = I∘T, which shows the formation of a low-light image [6]. L is the captured image, I is the desired recovery of the image and T is the illumination map. The captured image is, therefore, the element-wise multiplication of the desired recovery and the illumination map of the image. This estimation only boosts the global illumination of the image.

Once the max of the red, green and blue channels for each pixel has been found ($\hat{T}(x)$) to initialise the illumination map, the map needs to be further refined. The structure-aware smoothness matrix, W, is found from the formula $W_h(x) \leftarrow \Sigma_{y\in\Omega(x)} \frac{G_\sigma(x,y)}{|\Sigma_{y\in\Omega(x)} G_\sigma(x,y)\nabla_h \hat{T}(y)|+\epsilon}$ for the horizontal components and the same done but for the vertical components. $\Omega(x)$ is a region centred at a pixel x, and y is the location index within the region, $G_\sigma(x,y)$ is produced by the Gaussian kernel with a σ value of 3 in our experiments and ϵ is a small number used to avoid dividing by 0 [6]. With this value, you solve the linear system, $min_T||\hat{T} - T||_F^2 + \alpha||W \circ \nabla T||_1$ where α balances the two terms, $||\cdot||_F$ denotes the Frobenius norm and $||\cdot||_1$ the l_1 norm. This results in the refined illumination map and you perform gamma correction on this map then divide the original image by this refined illumination map to enhance the image as from L = I∘T, we want I and so I = $\frac{L}{T}$. The LIME method keeps the overall original structure of the image while smoothing its textural details. The very low-light input hides intensive noises in the dark. After LIME processes an image, the details are enhanced but so too is the noise in the image.

There are some disadvantages to using LIME. The colour of the images processed by LIME appear less natural [8] and it causes colour distortion in images [2] shown in Figure 1. When compared to histogram equalisation, LIME takes a lot longer to process the image. It can leave images over-exposed [11] as shown in Figure 2. Increasing the sizes of the images used in LIME also impacts its performance, making it more time-consuming [10]

Fig. 1: Output LIME method (second row) applied to an input image (first row) [6]

Fig. 2: Over exposure of an image after LIME method has been applied [11]

Two other possible models that can be used to solve the low-light enhancement problem are MSR-Net [12] and Dong [13]. Table I and Figure 3 shows the comparison between LIME, Dong and MSR-Net. SSIM shows how close the enhanced image is to the ground truth image and a lower NIQE value shows higher image quality. It can be seen that MSR-Net obtains the best results with the LIME model not too far off from it and still achieves good results. Due to this fact and that MSR-Net has a much slower running time, shown in Table II, with LIME having the fastest running time shows that LIME is a suitable model to use for our low-light enhancement problem.

TABLE I: Quantitative measurement results using SSIM/NIQE on test images [12]

Dataset	Ground truth	Dong	LIME	MSR-Net
2000 test images	1/3.67	0.69/4.16	0.84/3.89	0.92/3.46

TABLE II: Comparison of average running time on 100 images(seconds) [12]

Image Size	Dong	LIME	MSR-Net
500 x 500	0.321	0.188	2.422
750 x 750	0.681	0.359	5.043
1000 x 1000	1.103	0.521	8.962

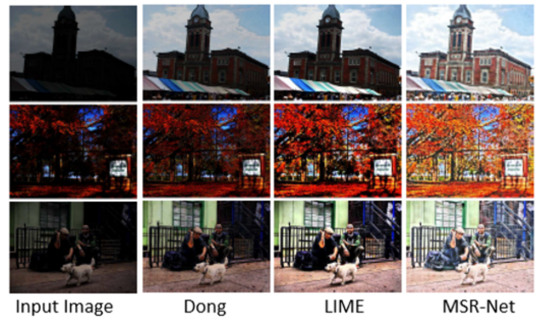

Fig. 3: Dong, LIME and MSR-Net output image comparison [12]

C. Face Detection

Possible methods for face detection include R-CNN, Faster R-CNN, Viola-Jones, HoG-SVM, Dual Shot Face Detector (DSFD), Pyramidbox and Single Stage Headless Face Detector (SSH). There are two different types of face detection methods. One is methods based on hand-crafted features while the other is methods that use a deeply learned approach. Viola-Jones and HoG-SVM make use of hand-crafted features like Haar while R-CNN and Faster R-CNN are examples of deeply learned methods [4]. R-CNN, Fast R-CNN and Faster R-CNN use a deep convolutional neural network and are used for many different computer vision tasks [14]. Deeply learned methods need to be trained on a dataset for face detection while hand-crafted features methods can be trained or make use of built-in functions to perform face detection. A lack of large-scale low-light face datasets [2] can create accuracy issues in models that need to be trained [15]. The low-light image enhancement method will first need to improve the illumination of these images before being used in a face detection model. Due to this reason, there will be limited images for training and testing a face detection model by using these images. This creates a major accuracy issue as the methods will significantly overfit the data [16]. It is beneficial to look at face detection methods that do not need to be trained but are still relatively accurate such as Viola-Jones.

*1) **Viola-Jones Method**:* Viola-Jones is a widely used method for face detection [17]. It is a hand-crafted method that uses Haar-like features to detect faces in the image. The method uses rectangular features in the image instead of its pixels [9]. The method makes use of three sections - integral image, AdaBoost, and the attentional cascade [3]. These methods are all used to increase the accuracy of the model from the basic Haar-like features [3]. The integral image section computes the rectangle features of the image rapidly. It does this by using an interpretation for the image which is called the integral image. This image at position (x,y) is made up of the sum of the pixels above and to the left of this position using the formula:

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Here, *ii(x,y)* is the integral image and *i(x,y)* is the input image [3]. This breaks the image up into sets of two adjacent rectangles as shown in Figure 4 [17].

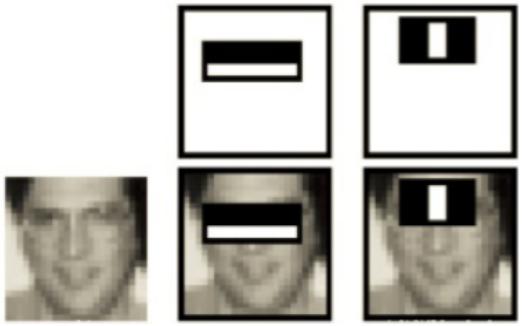

Fig. 4: Shows adjacent rectangles applied to image [3]

The AdaBoost learning algorithm is used to improve the performance of the classification. It does this by making the method search for possible sections and chooses the ones that have the best features in the image. This results in a refined image with meaningful sections [3]. The AdaBoost training results in a numerical description of the image by using "the voting mechanism to form a strong classifier from the weak one." [9, p.2]. This ensures that the section of the image that is being tested is precise to obtain high accuracy. The last section is the attentional cascade. It eliminates as many false positives in the image as possible to maintain a small error [17]. It does this by identifying a potential face object in the image and the attentional cascade discards it as a face option if it is not a face. This decreases the number of possible non-faces in the image. This increases the number of correct detections while decreasing false positives [17].

This method works best when the contrast of the image is high between the background and objects [9]. It detects faces efficiently and effectively but is sensitive to illumination, poses, and image quality [15]. It is accurate for front positioned faces while less accurate for tilted or side profile faces [17]. An example of the output images for Viola-Jones is shown in Figure 5.

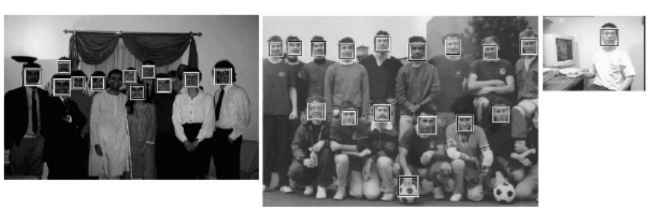

Fig. 5: Viola-Jones output images [3]

With a large dataset, Faster R-CNN achieves a high accuracy compared to traditional hand-crafted methods such as Viola-Jones [18]. This is shown in Table III. Their computational cost is very high which causes a limitation for many different applications [4]. Faster R-CNN's time complexity is also high compared to the Viola-Jones method [18]. These time complexities are shown in Table IV. The efficiency of the less accurate Viola-Jones method is another benefit over CNN methods [15]. So too is the fact that Viola-Jones doesn't require as large a training dataset as required for CNN methods [15]. Another disadvantage of this method, just like Viola-Jones, is that its accuracy decreases in situations involving cross-pose, low-lighting and low-resolution [15]. When detecting small faces, Faster R-CNN's accuracy also suffers as the background and the face are on the same pixel position on the feature map [4]. Overall, Viola-Jones is most suited for this report's problem. The accuracy of Viola-Jones also needs to be calculated for low-light images and enhanced illumination images to solve the research question. If the accuracy improves when the images are enhanced, the research question is solved and so we are able to choose the less accurate but more efficient model to see this improvement.

TABLE III: Accuracy of Faster R-CNN and Viola-Jones on two datasets. The mean average precision (mAP) measures the average precision of the detector and the detection performance is AUC [18]

Detector	FDDB Data		Casablanca Data	
	AUC	mAP (%)	AUC	mAP (%)
Viola-Jones	0.6654	67.16	0.42	37.68
Faster R-CNN	0.9253	91.73	0.55	56.37

TABLE IV: Average time and memory complexity for face detection on FDD (dataset) [18]

Detector	Time		Memory Consumption (GB)
	GFLOPS	FPS	
Viola-Jones	0.6	60	0.1
Faster R-CNN	223.9	5.8	2.1

III. Detailed Methodology

The research question states, "is there an improved accuracy in face detection when the illumination of the input image is improved?". To answer this research question, the accuracy of the face detection model, like Viola-Jones, needs to be found in two different scenarios. The first being the accuracy of the model with input images that have not had their illumination improved. The second being the accuracy of the model with input images where their illumination has been improved using a model like LIME. These two accuracies can then be compared and if the second accuracy is better than the first, there is an improved accuracy in face detection when the illumination of the input image is improved.

The steps needed to be taken to perform the low-light enhancement:

1) A dataset to test the low-light enhancement model needs to be used. The model needs both the low and normal-light versions of the same image to test the performance of the model. The full LOw Light paired dataset (LOL) is therefore used as it contains many (789) paired images in low and normal-light.

2) These images are all loaded into an array of images.
3) To make sure the images are consistent, all images are converted to RGB format.
4) We then perform the low-light enhancement method using the steps in the LIME model described in the background and related work section on LIME.
5) In this model we set $\gamma = 0.61$, $\alpha = 0.16$, $\sigma = 5$, and $\epsilon = 0.001$
6) To perform the low-light enhancement, the image size is first checked to save on computational time. If the size is >3000x3000, the image is resized to be a third of its original size and if >1500x1500, half of the original size.
7) A Gaussian kernel is created using the sigma value and size=$2(4\sigma + 0.5) + 1$ and the formula for the Gaussian kernel being $g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$
8) The image is then normalised and the initial estimation of the illumination map is created by finding the maximum value from the red, green or blue colour channels of the image for each pixel. $\hat{T} = max_{c\in\{R,G,B\}} L$ where L is the original image.
9) Using this estimated illumination map, we compute the smoothness weights for the vertical and horizontal components (Wv and W_h) using the formula $W_h(x) \leftarrow \Sigma_{y\in\Omega(x)} \frac{G_\sigma(x,y)}{|\Sigma_{y\in\Omega(x)} G_\sigma(x,y) \triangledown_h \hat{T}(y)|+\epsilon}$ where the derivative of the image is found using openCV's Sobel function. Adding ϵ in the denominator avoids dividing by 0.
10) We can now solve the linear system, $min_T ||\hat{T} - T||_F^2 + \alpha||W \circ \triangledown T||_1$, which results in the refined illumination map.
11) Gamma correction is then performed by multiplying this refined illumination map by the γ value. This allows for further control of the improved illumination enhancement of our image.
12) Lastly, since L = I∘T where L is the captured image, I is the desired recovery of the image (improved illumination image) and T is the illumination map, we manipulate the formula to obtain I = $\frac{L}{T}$. From this formula, we divide the original image by our newly found illumination map after gamma correction and the result is our image with enhanced illumination. This enhanced image is then saved in a new folder containing all the enhanced images for comparison or face detection purposes.
13) The steps 6 to 12 are repeated for all images in our input dataset in step 1 [19].
14) To see how well our implementation of the low-light enhancement model performs, we compare our enhanced images with that of their normal-light equivalent images in the LOL dataset by finding the mean squared error and PSNR values. Mean squared error is found using the formula, $\frac{\Sigma_{i=1}^n (Y_i - \hat{Y}_i)^2}{n}$ where n is the number of images, Y_i is the normal-light image and $\hat{Y}_i$ is the enhanced image. PSNR is Peak signal-to-noise ratio and is calculated using the formula, $20 \times log_{10}(\frac{255}{\sqrt{MSE}})$. The average MSE and PSRN are found for all the images.

For face detection:

1) The dataset needed for the face detection model must contain images that have poorly lit faces in them. We, therefore, use a modified version of the ExDark dataset explained in the "Experimental Results" section.
2) These images are loaded and also need to be converted to RGB as mentioned above.
3) For this implementation of our model that is similar to that of Viola-Jones, we convert the current image to be a grayscale image using OpenCV.
4) We then create our cascade classifier by using OpenCV's Cascade Classifier method to load our haarcascade_frontalface_alt classifier that is used in the Viola-Jones model. It uses Haar-like features to detect the faces in the image and his classifier makes use of AdaBoost learning which selects a small number of key features in the image. Haar-like features are the digital features of the image. This classifier ignores non-face objects and detects faces in the image.
5) We then detect the faces in the image by using this classifier and OpenCV's detectMultiScale on the grayscale image with a scale factor of 1.1 and the minimum number of neighbours being 2. Detect Multiscale is used at it returns the detected faces as a list of rectangles (coordinates of the 4 corners around a detected face) bounding the faces.
6) The detected faces (list of rectangles) are then looped through to draw a green bounding box around the detected faces using OpenCV's rectangle method at the position of the returned list of rectangles to indicate the detected faces on the image.
7) This image is then converted back to RGB and contains a green bounding box around all the detected faces in the image. This image is then saved in a new folder containing all the face detection images with bounding boxes so the accuracy of the model can be found.
8) Steps 3 to 7 are repeated for all images in our dataset in step 1 to obtain a folder containing all the images with bounding boxes indicating the face detection.

To solve the research question, we need to perform the low-light image enhancement followed by the face detection by performing the following steps:

- The face detection first needs to be done on the modified ExDark dataset before any low-light image enhancement is done. The accuracy of face detection then needs to be found from these images.
- The face detection then needs to be done on the modified ExDark dataset after the images have been enhanced from our low-light enhancement model. To do this, the low-light enhancement model is executed on the modified ExDark dataset. These enhanced images are then used in

the face detection model. The accuracy of face detection then needs to be found from these images.

- The accuracy of face detection is found by counting the number of faces detected in the image and store this number as a total for all faces found in all the images. Then divide the total faces detected by the total faces in the images $\frac{totalFacesDetected}{totalFaces}$. The first accuracy is found for the original input images with no low-light enhancement being applied to the image. The second accuracy is calculated from first applying the low-light enhancement model, explained above, to the input images. These enhanced images are then used as the input images by the face detection model and the accuracy is found.
- The same input images are used in both scenarios to keep the experiment fair. The only difference is the illumination of the input images for both scenarios.
- A confusion matrix is calculated by counting the number of correctly detected faces (true positive), counting the number of missed faces (false negative), counting the number of incorrectly identified faces (false positive) and counting the number of images that contain no faces where no faces were identified (true negative),
- These two accuracies are then compared to show that there is an improved accuracy in face detection when the illumination of the input image is improved.
- A flowchart of the process can be seen below in Figure 6.

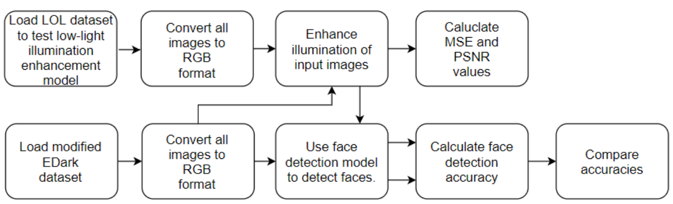

Fig. 6: Process to solve research question

IV. Experimental Results

A. Experimental Setup

All code was implemented in python. Before the final results are obtained to solve the research problem, the best parameters need to be chosen for our low-light enhancement model. To do this, our test dataset (LOL) was run with different values for γ, α, σ, and ϵ. The mean square error and the peak signal-to-noise ratio was found for each of these implementations. The higher the peak signal-to-noise ratio, the closer the enhanced illumination image is to that of the same image but in normal-light in the LOL dataset. Good PSNR values usually lie between 40 and 60. A lower MSE also shows less of a difference between the two images. Examples of some of the different combinations of parameters are shown in Table V. The chosen parameters for this model was $\gamma = 0.61$, $\alpha = 0.16$, $\sigma = 5$, and $\epsilon = 0.001$

TABLE V: Comparison of different parameters

γ	α	σ	ϵ	MSE	PSNR
0.5	0.12	3	0.001	0.2905	53
0.8	0.17	2	0.001	0.5987	50
0.61	0.16	5	0.001	0.0807	59

B. Datasets

In the experiments, two datasets are used, the LOw Light paired dataset (LOL) and the ExDark dataset. The LOw Light paired dataset (LOL) contains two different groups of images. The first group contains images taken in buildings of scenery that does not contain any human faces in low-light. The second group contains the same images but in normal-light. An example of these images can be seen in Figure 7. The full dataset of 790 images is used.

The second dataset is the modified ExDark dataset. This dataset contains 300 images taken from the ExDark dataset. From the 300 images, 90 of them contain no human faces at all to test for false positives. The other images contain one or more faces to be detected. In total there are 407 human faces in this dataset. This face detection dataset contains mostly dark faces but also some normal-light to see if LIME affects those negatively and causes them not to be detected.

Fig. 7: Low-light vs normal-light image from LOL dataset

C. Results

Following the methodology for the implementation of the low-light image enhancement using the parameters, $\gamma = 0.61$, $\alpha = 0.16$, $\sigma = 5$, and $\epsilon = 0.001$ results in a MSE value of 0.0807 and a PSNR value of 59. This shows the implementation of our low-light image enhancement results in images that are similar to that of the real normal-light images. The PSNR ratio is a quality measurement between the original and the enhanced image. The higher the PSNR value, the higher the quality of the enhanced image [20]. An example of our low-light image enhancement can be seen in Figure 8

Running the face detection model on the modified ExDark dataset leads to the following results:

- 246 faces being correctly identified from the 407 faces (true positive).

- 161 faces being missed from the 407 faces (false negative).
- 56 faces were found where none exist (false positive).
- 89 out of the 90 images that did not contain any faces were correctly classified (true negative).

This confusion matrix can be seen in Figure 10. Using the formula, $\frac{TP+TN}{TP+TN+FP+FN}$, the accuracy can be found and is therefore 60.69%. The precision is found using the formula $\frac{TP}{TP+FP}$ which results in a value of 81.46%. Precision is the fraction of relevant (positive) cases among the obtained cases. The recall is found using the formula $\frac{TP}{TP+FN}$ which results in a value of 60.44%. The recall is the fraction of the total amount of relevant cases that were obtained. Low accuracy is expected due to the low-light nature of the images

Running the face detection model on the modified ExDark dataset after being enhanced by out low-light enhancement model leads to the following results:

- 309 faces being correctly identified from the 407 faces (true positive).
- 98 faces being missed from the 407 faces (false negative).
- 58 faces were found where none exist (false positive).
- 82 out of the 90 images that did not contain any faces were correctly classified (true negative).

This confusion matrix can be seen in Figure 11. The accuracy is 71.48%. The precision is 84.19%. The recall is 75.92%. These results along with the previous results can be seen in Table VI.

From both sets of image results (before and after low-light enhancement) it can be seen that our face detection model is susceptible to tilted heads, extreme side views of faces as well as people who wore low tilted hats as can be seen in Figure 12. It can be seen that our face detection model performs much better after the low-light image enhancement has been performed on the images from our accuracy, precision and recall results. An Example of face detection before and after low-light enhancement can be seen in Figure 13. Faces in paintings and on screens can also be detected using this model.

Fig. 8: Low-light vs enhanced light using our model vs normal light

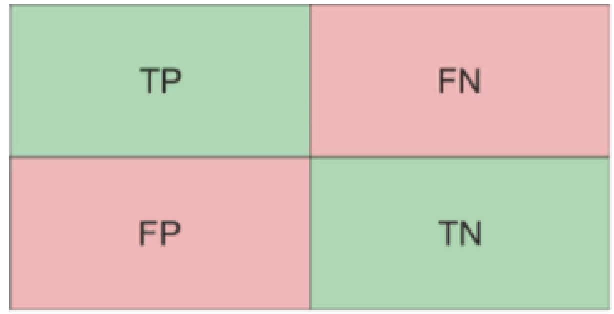

Fig. 9: Confusion matrix layout

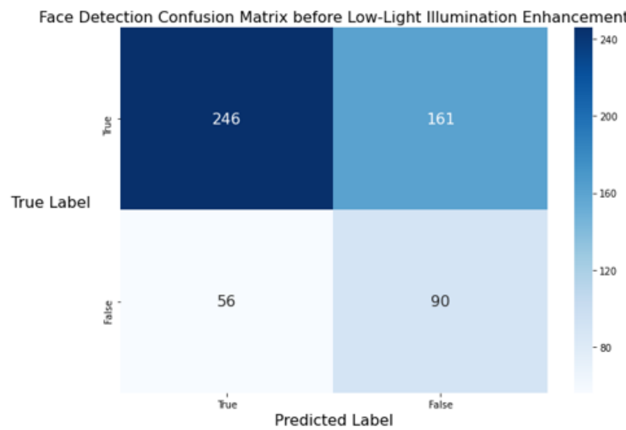

Fig. 10: Confusion matrix for face detection before low-light enhancement

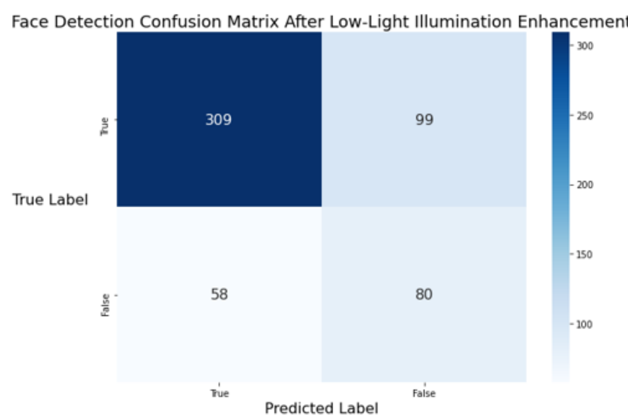

Fig. 11: Confusion matrix for face detection after low-light enhancement

TABLE VI: Accuracy, precision and recall of face detection model before and after enhancement

Face Detection	Accuracy	Precision	Recall
Before low-light enhancement	60.69%	81.46%	60.44%
After low-light enhancement	71.48%	84.19%	75.92%

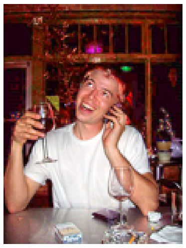

Fig. 12: Face not detected due to extreme tilt in head

Fig. 13: Face detection example before vs after low-light enhance

V. Conclusion

This report looks at solving the issue of face detection in low-light images. Face detection methods rely heavily on the image quality for detection. One of the main contributors to a low-quality image is low-light. For the low-light image enhancement, a method based on the LIME model was used due to its low computational cost and it enhanced the low-light images well. For this research problem, the face detection method with the best accuracy does not need to be chosen. We need to compare the accuracy of the face detection method we chose without using low-light image enhancement and with using it to see if there is an improved accuracy. For face detection, a method based on Viola-Jones was used. Although it does have lower accuracy than other methods like Faster R-CNN, its ability to detect faces in images is much faster. It also has a much smaller memory consumption and is more efficient. From these models, a detailed methodology was created and contains all the steps needed to be completed for the research question to be answered. By executing the steps in the methodology, results were obtained for the face detection before and after low-light enhancement was performed on the input images (a modified version of ExDark). These results show that our face detection model improves drastically when the input images are first enhanced. There is a 10.79% increase in accuracy, a 2.73% increase in precision and a 15.48% increase in recall. This proves that there is an improved accuracy in face detection when the illumination of the input image is first improved and thus solves the research question.

References

[1] Y. Yuan, W. Yang, W. Ren, J. Liu, W. Scheirer, and Z. Wang, "UG^{2+} track 2: A collective benchmark effort for evaluating and advancing image understanding in poor visibility environments," Sep. 2019, https://arxiv.org/abs/1904.04474.

[2] F. Lv, Y. Li, and F. Lu, "Attention-guided low-light image enhancement," Nov. 2019, https://arxiv.org/abs/1908.00682.

[3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, Feb 2001, pp. I–I, https://ieeexplore.ieee.org/document/990517.

[4] Y. Zhou, D. Lui, and T. Huang, "Survey of face detection on low-quality images," in *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*. IEEE, Apr. 2018, pp. 769–773, https://arxiv.org/abs/1804.07362.

[5] Y. P. Loh and C. S. Chan, "Getting to know low-light images with the exclusively dark dataset," vol. 178, 2019, pp. 30–42.

[6] X. Guo, "LIME: A method for low-light image enhancement," in *Proceedings of the 24th ACM international conference on Multimedia*, May. 2016, pp. 87–91, http://arxiv.org/abs/1605.05034.

[7] LOw Light paired dataset (LOL) can be found at https://daooshee.github.io/BMVC2018website/.

[8] Y. Guo, X. Ke, J. Ma, and ZhangJ., "A pipeline neural network for low-light image enhancement," *IEEE Access*, vol. 7, pp. 13 737–13 744, Jan 2019, https://ieeexplore.ieee.org/document/8607964?denied=.

[9] M. Ahad, T. Paul, U. Shammi, and S. Kobashi, "A study on face detection using viola-jones algorithm for various backgrounds, angels and distances," in *Applied Soft Computing*, May 2018, https://www.researchgate.net/publication/327415969_A_Study_on_Face_Detection_Using_Viola-Jones_Algorithm_for_Various_Backgrounds_Angels_and_Distances.

[10] Z. Rahman, M. Aamir, Y. PU, F. Ullah, and Q. Dai, "A smart system for low-light image enhancement with color constancy and detail manipulation in complex light environments," in *Symmetry*, vol. 10, Dec 2018, p. 718, https://www.researchgate.net/publication/329424020_A_Smart_System_for_Low-Light_Image_Enhancement_with_Color_Constancy_and_Detail_Manipulation_in_Complex_Light_Environments.

[11] C. Wei, W. Wang, W. Yang, and J. Liu, "Deep retinex decomposition for low-light enhancement," in *British Machine Vision Conference (BMVC)*, Aug. 2018, https://arxiv.org/abs/1808.04560.

[12] L. Shen, Z. Yue, F. Feng, Q. Chen, S. Liu, and J. Ma, "Msr-net:low-light image enhancement using deep convolutional network," Nov. 2017, https://arxiv.org/abs/1711.02488.

[13] X. Dong, G. Wang, Y. Pang, W. Li, J. Wen, W. Meng, and Y. Lu, "Fast efficient algorithm for enhancement of low lighting video," in *2011 IEEE International Conference on Multimedia and Expo*, vol. 7, Aug 2011, pp. 1–6, https://ieeexplore.ieee.org/document/6012107.

[14] H. Jiang and E. Learned-Miller, "Face detection with the faster R-CNN," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, Jun. 2017, pp. 650–657, https://arxiv.org/abs/1606.03473.

[15] A. Shepley, "Deep learning for face recognition: A critical analysis," vol. 1, Jul 2019, https://arxiv.org/abs/1907.12739.

[16] G. Hu, X. Peng, Y. Yang, T. Hospedales, and J. Verbeek, "Frankenstein: Learning deep face representations using small data," in *IEEE Transactions on Image Processing*, vol. 27, Sep 2017, pp. 293–303, https://arxiv.org/abs/1603.06470.

[17] K. Enriquez, “Faster face detection using convolutional neural networks and the viola-jones algorithm,” May. 2018, https://www.csustan.edu/sites/default/files/groups/University%20Honors%20Program/Journals/01_enriquez.pdf.

[18] L. Nguyen-Meidine, E. Granger, M. Kiran, and L. Blais-Morin, “A comparison of cnn-based face and head detectors for real-time video surveillance applications,” Nov 2017.

[19] S. Attaiki, 04 2020, https://github.com/pvnieo/Low-light-Image-Enhancement.git.

[20] A. Tanchenko, “Visual-psnr measure of image quality,” vol. 25, 07 2014.

Declaration
University of the Witwatersrand, Johannesburg
School of Computer Science and applied Mathematics
SENATE PLAGIARISM POLICY

I, Steven Michael Curtis, (Student number: 1657041) am a student registered for COMS4059A, Research Project: Computer Science in the year 2020.

I hereby declare the following:

- I am aware that plagiarism (the use of someone else's work without their permission and/or without acknowledging the original source) is wrong.
- I confirm that ALL the work submitted for assessment for the above course is my own unaided work except where I have explicitly indicated otherwise.
- I have followed the required conventions in referencing the thoughts and ideas of others.
- I understand that the University of the Witwatersrand may take disciplinary action against me if there is a belief that this in not my own unaided work or that I have failed to acknowledge the source of the ideas or words in my writing.

Signature: SM Curtis **Date**: 29/11/2020