ISSN 2347 - 3983
Volume 8. No. 9, September 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter236892020.pdf
https://doi.org/10.30534/ijeter/2020/236892020

Fatigue Detection among Operators in Industry Based on Euclidean Distance Computation Using Python Software

Ahamad Zaki Mohamed Noor[1], Fairul Azni Jafar[2], Mohd Riduan Ibrahim[3], Shahril Nizam Mohamed Soid[4]

[1]System Engineering and Energy Laboratory, Universiti Kuala Lumpur Malaysian Spanish Institute, Kulim Hi – Tech Park, 09000, Kulim, Kedah, Malaysia, ahamadzaki@unikl.edu.my

[2]Centre of Smart System and Innovative Design, Faculty of Manufacturing Engineering, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100, Durian Tunggal, Melaka, Malaysia, fairul@utem.edu.my, fairul@utem.edu.my

[3]Manufacturing Section, Universiti Kuala Lumpur Malaysian Spanish Institute, Kulim Hi – Tech Park, 09000, Kulim, Kedah, Malaysia, mohdriduan@unikl.edu.my

[4]Engineering Section, Universiti Kuala Lumpur Malaysian Spanish Institute, Kulim Hi – Tech Park, 09000, Kulim, Kedah, Malaysia, shahrilnizam@unikl.edu.my

ABSTRACT

Machine – learning is one of popular technique suitable for adaptation in Industrial Revolution 4.0 (IR4). There is a dire problem whereby increasing occupational accident especially in the manufacturing sectors. The root cause of these accidents are because of fatigue while performing repetitive task in production line. To solve this problem, a research was conducted in developing fatigue detection algorithm. The software used for this algorithm development is python software since this software is an open source software. Euclidean distance computation is utilized in this algorithm in determining the eyes and mouth aspect ratio. The eyes and mouth aspect ratio were set 0.28 and 0.60 respectively. If the eyes aspect ratio is below than 0.28, the output obtained is eyes closed. If mouth aspect ratio is higher than 0.60, than the operator is yawning and fatigue alert will appear notify the line leader in manufacturing plant.

Key words: Fatigue Detection, Euclidean Distance, Image Processing and Machine – Learning

1. INTRODUCTION

Manufacturing company comprises lot of workstation in order to prepare a product. When there are more workstation, there is a dire need of people to work in a manufacturing plant. Human tends to get fatigue and drowsy if working long hours. Eventually the human operator involves in an accident. A statistic which were brought up by the Department of Occupational Safety and Health (DOSH) under Ministry of Human Resource regarding occupational accidents statistics according to sector. The statistic of accidents shown in Table 1.

Table 1: Occupational accident statistics by sector from January to July 2020 (Source: International Policy and Research Development Division)

SECTOR	NPD	PD	DEATH	TOTAL
Hotel and Restaurant	72	2	2	76
Utilities (Electricity, Gas, Water and Sanitary Service)	148	3	2	153
Finance, Insurance, Real Estate and Business Services	202	3	3	208
Construction	101	1	35	137
Transport, Storage and Communication	218	4	3	225
Manufacturing	2443	122	35	2600
Wholesale and Retail Trade	59	1	1	61
Public Services and Statutory Authorities	33		1	34
Mining and Quarrying	18		2	20
Agriculture, Forestry and Fishery	572	10	29	611
TOTAL	3866	146	113	4125

LEGEND:
PD - PERMANENT DISABILITY
NPD- NON PERMANENT DISABILITY

From Table 1, observe that the highest accident involving the manufacturing sector. Total of 2600 cases with cumulative of permanent disability, non-permanent disability and death were recorded. The research objective is developing a fatigue alert image processing based on real time situation using python software. The feature available include open and close eyes, yawn count and fatigue alert. This machine-learning algorithm could be utilize in manufacturing plant to decrease the accident rate. The line leader will obtain the alert and pull out the operator to freshen up before any harm happen to the person, or the surrounding of the person. Increase in accident reflects to higher loss incur by manufacturing company. There are several techniques suitable to be used to replace machine-learning, however [1] mentioned the weakness according to several applications. Other researchers [2] performed this human emotion using Euclidean distance formula. From this computation, abbreviates set to determine angry, happy and other emotion. An algorithm namely deep convolutional neural network to detect fermentation by [3]. However, this algorithm is suitable also for emotion or fatigue detection. Other researcher that implemented machine-learning technique shown in Table 2.

Table 2: Algorithm used by researchers for image processing

Author(s)	Algorithm	Application
[4]	Genetic Algorithm	EEG signal processing to extract mixed features
[5]	Cascade Classifier and Tensor Processing	Driver's eye recognition and fatigue monitoring
[6]	Fuzzy Logic approach	Smile intensity recognition
[7]	Convolutional Neural Network	Recognition of facial expression
[8]	Image mining	Intelligent drowsy eye detection
[9]	Blind Image Separation	Mixing and Estimating Sample Images
[10]	Convolutional Neural Network	Brain tumor image classification

From table 2, there are lot of suitable techniques used and implemented in this research. Each of this tool were utilize according to designated application. However, generally, Euclidean distance adapted by several researcher to recognize, detect and monitor activity on face.

2. METHODOLOGY

In this paper, the method of input and output of this fatigue detection algorithm were discussed through illustrated flowchart. Figure 1 shows the flowchart of developing fatigue detection algorithm.

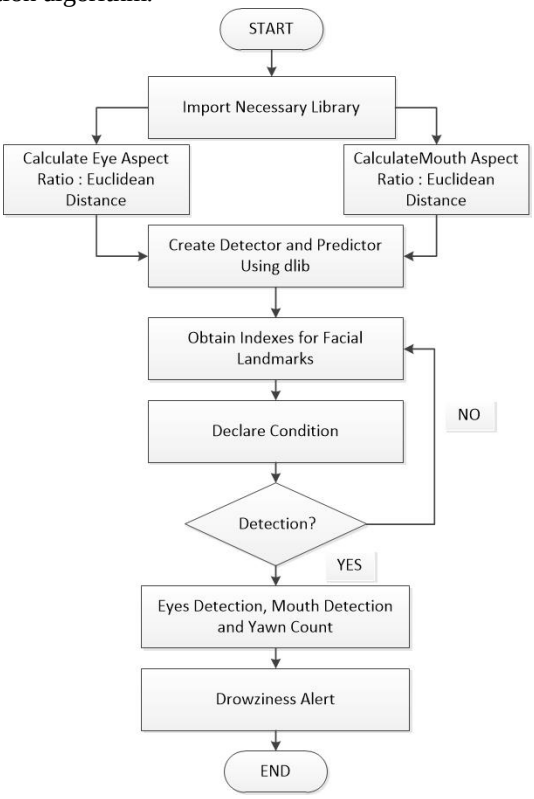

Figure 1: ARIMA Algorithm Flowchart

2.1 Phase 1: Import Library

There are necessary library need to be included while doing python programming. The necessary library or package to be included is distance from scipy.spatial to compute Euclidean distance. Import face utils from imutils, dlib and cv2 from openCV. Figure 2 shows the command to import packages for this algorithm development.

```
from scipy.spatial import distance as dist
from imutils import face_utils
import imutils
import dlib
import cv2
```

Figure 2: Import necessary library in Python

2.2 Phase 2: Computation of Euclidean Distance

Once the necessary library imported, this algorithm were started with the computation of Euclidean distances on eyes and mouth. Both eyes and mouth aspect ratio were obtained through the Euclidean distance between horizontal and vertical at eye and mouth position. Figure 3 shows the computation of Euclidean distance and average. Figure 3 shows the computation of Euclidean distance computed in Python software.

```
def eye_aspect_ratio(eye):
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])
    C = dist.euclidean(eye[0], eye[3])
    ear = (A + B) / (2.0 * C)
    return ear

def mouth_aspect_ratio(mou):
    X   = dist.euclidean(mou[0], mou[6])
    Y1  = dist.euclidean(mou[2], mou[10])
    Y2  = dist.euclidean(mou[4], mou[8])
    Y   = (Y1+Y2)/2.0
    mar = Y/X
    return mar
```

Figure 3: Computation of Euclidean distance

2.3 Phase 3: Detector and Predictor using dlib

Then, the camera were turned on and process real time image. Each eye and mouth aspect ratio were defined. Dlib function were utilize in creating detector and predictor on face. Figure 4 is the utilization of dlib for detector and predictor of frontal face.

```
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(predictor_path)
```

Figure 4: Utilization of dlib function

2.4 Phase 4: Declare Closed Eyes Condition

Next phase of this algorithm development is to declare condition. If eyes ratio is below blink, then there is an increment in the blink frame counter. The eyes closed signal uses red colour. The same goes for the font colour of fatigue alert. The fatigue alert will be displayed Figure 5 is the eyes

closed condition. In this research, the declared eyes aspect ratio was 0.28. If the eyelid is below than 0.28, display shows fatigue alert and eyes closed.

```
if ear < EYE_AR_THRESH:
    COUNTER += 1
    cv2.putText(frame, "Eyes Closed ", (10, 30),cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)

if COUNTER >= EYE_AR_CONSEC_FRAMES:
        cv2.putText(frame, "FATIGUE ALERT!", (10, 50),
            cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
```

Figure 5: Declare eyes closed condition

2.5 Phase 5: Declare Opened Eyes Condition

Next condition declared for the case of eyes that does not closed. If the eyes ratio is above blink, hence the counter and fatigue alert alarm were reset. Figure 6 shows the condition of eyes open. In this research, the declared eyes aspect ratio was 0.28. If the eyelid is above than 0.28, display shows eyes opened. There were no fatigue alert declared since the outcome displayed eyes open.

```
else:
    COUNTER = 0
    cv2.putText(frame, "Eyes Open ", (10, 30),cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255, 0), 2)

cv2.putText(frame, "EAR: {:.2f}".format(ear), (480, 30),
    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
```

Figure 6: Declare eyes opened condition

2.6 Phase 6: Computation and Condition for Yawn Count

Last phase of this algorithm development is the yawning detection. Yawning detection is based on the mouth aspect ratio. There will significant different in the mouth aspect ratio to detect if the operator in manufacturing plant yawning. Increase in mouth aspect ratio resulting the output of "yawn count". Mouth aspect ratio declared in this algorithm was 0.60. If the Euclidean distance computation gives ratio more than 0.60, hence the operator is yawning. Each time the operator yawning, there will be count. This yawn count shows another indication if the human operator working in manufacturing plant is fatigue. Figure 7 shows the programming to declare condition and cumulative of operator that yawns.

```
if mouEAR > MOU_AR_THRESH:
    cv2.putText(frame, "Yawning ", (10, 70),cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
    yawnStatus = True
    output_text = "Yawn Count: " + str(yawns + 1)
    cv2.putText(frame, output_text, (10,100),cv2.FONT_HERSHEY_SIMPLEX, 0.7,(255,0,0),2)
else:
    yawnStatus = False

if prev_yawn_status == True and yawnStatus == False:
    yawns+=1

cv2.putText(frame, "MAR: {:.2f}".format(mouEAR), (480, 60),
    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
cv2.putText(frame,"Fatigue Detection",(370,470),cv2.FONT_HERSHEY_COMPLEX,0.6,(0,0,255),1)
```

Figure 7: Yawn count computation and declaration

3. RESULT AND DISCUSSION

From the proposed methodology, an experiment was conducted to observe the functionality of the fatigue detection algorithm. Each condition were tested. The limit set to eyes and mouth aspect ratio were tested. Figure 8 is the illustration of closed eyes.

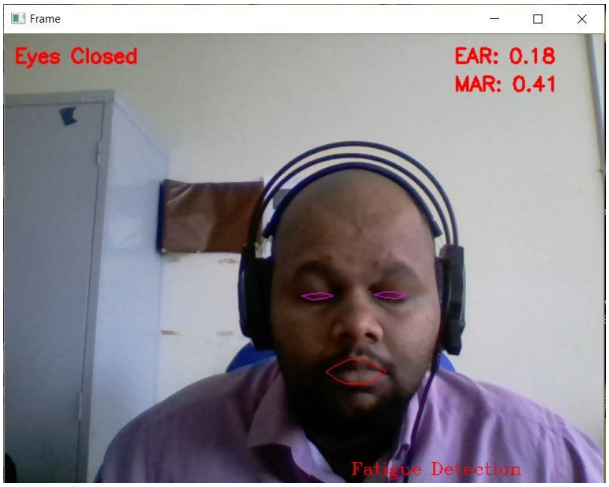

Figure 8: Closed eyes

From figure 8, the reading of eye aspect ratio and mouth aspect ratio were shown on the top right corner. The outcome displayed on the top left corner. The reading of eyes aspect ratio is 0.18 whereby below than 0.28. Hence, the outcome displayed is eyes closed. As for mouth aspect ratio, the computation obtained was 0.41 whereby the reading shown is lower than 0.60. This signify that the operator falls asleep while working. Next is to validate the algorithm output of fatigue detection. Figure 9 illustrate the display of fatigue alert.

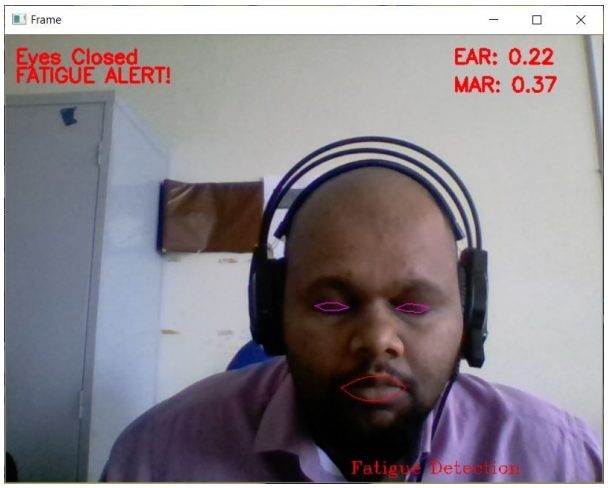

Figure 9: Fatigue alert

From figure 9, the reading of ear aspect ratio and mouth aspect ratio shows not much of a different compared to figure 8. The readings shows below the defined value of eyes and mouth aspect ratio shows below 0.28 and 0.6 respectively. The fatigue alert appear when the eyes closed for 2 seconds.This signal will be sent to the line leader and notify the line leader before any accident may take place upon this operator. To further experiment the development of fatigue detection algorithm, the yawning condition was tested. Figure 10 ilustrate the yawn count each time the operator yawns.

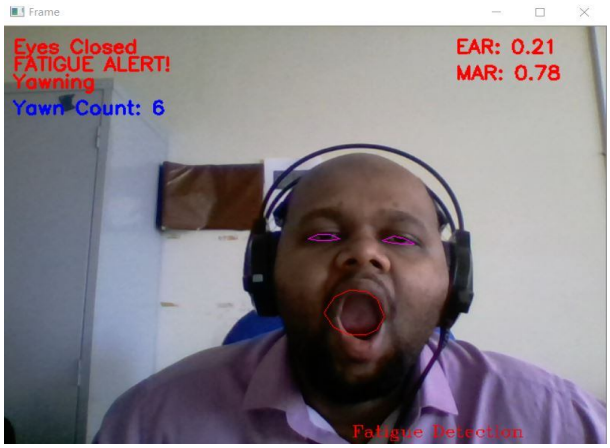

Figure 10: Yawn count

From figure 10, the reading of ear and mouth aspect ratio have slight difference. The eye aspect ratio is 0.21 whereby still less compared to the defined eyes aspect ratio 0.28. However, the mouth aspect ratio have increased from the set aspect ratio. The reading of the mouth aspect ratio is 0.78. This signify that the operator yawn while working. Observe on the top left of this frame. The yawn count is six times. The line leader can observe that this operator is unfit and need to put a stop from what the operator is assigned for. Next phase of validification process is to validate the string function in Python software. Figure 11 illustrate the cumulative of yawn count.

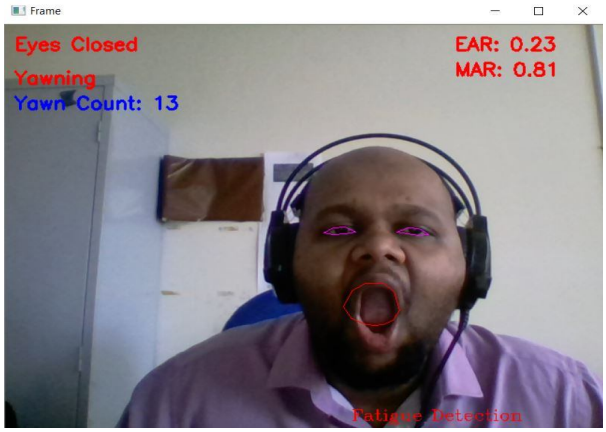

Figure 11: String function in yawn function

Figure 11 shows that the yawn count have increased from six to thirteen. Each time an operator yawn will be cumulative unless the "q" key is pressed to reset all count. This key act as a destroy function in Python to stop and reset. If this function did not utilize, another operator results increase cumulative from thirteen instead of one. However, from the image there is slight error in eyes closed. There is a need to revise the eyes aspect ratio. However, different operator may yawn while the eyes closed or open. Before this algorithm is deployed, a calibration on eyes and mouth aspect ratio need to be identified. Last validation of this experiment is to observe the eyes open condition. Figure 12 illustrate the condition for eyes open.

Figure 12: Opened eyes

From figure 12, the reading of opened eyes were obtained. The reading for eyes aspect ratio and mouth aspect ratio were 0.30 and 0.41 respectively whereby higher that the initial set aspect ratio of both eyes and mouth. This condition will not display the yawn count and fatigue alert continuously for the ease of observing the operator's condition while working.

4. CONCLUSION

To conclude, machine-learning concept was utilize in developing the algorithm of fatigue detection. Since different person may have different size and shape of frontal face, hence there is a need to calibrate the eyes and mouth aspect ratio. However, this is the weakness of this algorithm in need of unsupervised learning algorithm. In another word, artificial intelligence optimization tools need to be deployed in this algorithm to autonomously set the eyes and mouth aspect ratio. There is also lacking of processing two face in one camera whereby the readings may overlap making the line leader difficult to determine the condition of the operator. To ease determination of operator's condition, different camera need to be deploy and will cause high consumption of expenses. In short, machine – learning by utilizing Euclidean distance is the best to identify the fatigue condition in an operator. This method may result in reducing the industrial accident especially in manufacturing sectors.

ACKNOWLEDGEMENT

The authors would like to acknowledge to the financial support provide by Universiti Kuala Lumpur. The present work is a part of Short Term Research Grant (STR19067). Thank you also to those who have given comments and improvement on this research paper.

REFERENCES

1. A. Z. Mohamed Noor, M. H. F. Md Fauadi, F. A. Jafar, N. R. Mohamad, and A. S. Mohd Yunos, **A review of techniques to determine alternative selection in design for remanufacturing**, *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 257, no. 1, pp. 1–10, 2017.
2. J. S. Mehta N., **Facial Emotion recognition using Log Gabor filter and PCA**, *Proc. - 2nd Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2016*, pp. 1–5, 2017.
3. G. Kimutai, A. Ngenzi, R. N. Said, A. Kiprop, and A. Förster, **An optimum tea fermentation detection model based on deep convolutional neural networks**, *Data*, vol. 5, no. 2, pp. 1–26, 2020.
4. Y. Li, L. Wu, T. Wang, N. Gao, and Q. Wang, **EEG Signal Processing Based on Genetic Algorithm for Extracting Mixed Features**, *Int. J. Pattern Recognit. Artif. Intell.*, vol. 33, no. 6, 2019.
5. B. Cyganek and S. Gruszczyński, **Hybrid computer vision system for drivers' eye recognition and fatigue monitoring**, *Neurocomputing*, vol. 126, pp. 78–94, 2014.
6. C. Vinola and K. Vimala Devi, **Smile intensity recognition in real time videos: fuzzy system approach**, *Multimed. Tools Appl.*, vol. 78, no. 11, pp. 15033–15052, 2019.
7. Y. Liu and Y. Chen, **Recognition of facial expression based on CNN-CBP features**, *Proc. 29th Chinese Control Decis. Conf. CCDC 2017*, pp. 2139–2145, 2017.
8. A. Emam, **Intelligent drowsy eye detection using image mining**, *Inf. Syst. Front.*, vol. 17, no. 4, pp. 947–960, 2015.
9. M. Mekhfioui, R. Elgouri, A. Satif, and L. Hlou, **Real time hardware co-simulation for blind image separation algorithm using ZYNG 7000 & xilinx system generator**, *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 2, pp. 365–371, 2020.
10. B. Anilkumar and P. Rajesh Kumar, **Tumor classification using block wise fine tuning and transfer learning of deep neural network and KNN classifier on MR brain images**, *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 2, pp. 574–583, 2020.