DAFTAR PUSTAKA

Cui, Y., Lin, K., Chen, Y., & Zhu, J. (2021). A quantum-inspired model for statistical analysis of repairable systems. Computers & Industrial Engineering, 161, 107613. https://doi.org/10.1016/j.cie.2021.107613

Dhananjay, K., & Salman, E. (2021). Charge Based Power Side-Channel Attack Methodology for an Adiabatic Cipher. Electronics, 10(12), 1438. https://doi.org/10.3390/electronics10121438

Hajian Heidary, M., & Aghaie, A. (2019). Risk averse sourcing in a stochastic supply chain: A simulation-optimization approach. Computers & Industrial Engineering, 130, 62–74. https://doi.org/10.1016/j.cie.2019.02.023

Li, H., Li, C., Wang, J., Yang, A., Ma, Z., Zhang, Z., & Hua, D. (2023). Review on security of federated learning and its application in healthcare. Future Generation Computer Systems, 144, 271–290. https://doi.org/10.1016/j.future.2023.02.021

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2017). Focal Loss for Dense Object Detection. 2017 IEEE International Conference on Computer Vision (ICCV), 2999–3007. https://doi.org/10.1109/ICCV.2017.324

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2015). SSD: Single Shot MultiBox Detector. https://doi.org/10.1007/978-3-319-46448-0_2

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788. https://doi.org/10.1109/CVPR.2016.91

Sundarambal, B., Mathana, J. M., Subramanian, S., Sandesh, H., & Omprakash, G. (2020). A Detailed Investigation on Reduction of False Positive Rate in Breast Cancer Detection. 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 1077–1079. https://doi.org/10.1109/ICACCS48705.2020.9074382

Wang, W., Wang, H., Ran, Z.-Y., & He, R. (2021). Learning Robust Feature Transformation for Domain Adaptation. Pattern Recognition, 114, 107870. https://doi.org/10.1016/j.patcog.2021.107870

Zhang, D., Xu, Z., Wang, L., Song, C., & Xu, Z. (2019). Digital Logic Experiment Design Based on FPGA Development Board and OV2640 Camera. 2019 14th International Conference on Computer Science & Education (ICCSE), 671–675. https://doi.org/10.1109/ICCSE.2019.8845424

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv. DOI: 10.48550/arXiv.1804.02767

Kim, J., Lee, S., & Kim, Y. (2020). Real-time facial recognition for online exam monitoring. Springer. DOI: 10.1007/s11042-020-08769-7

Zhao, Z., Li, W., & Liu, Y. (2020). Behavior analysis in online exams using artificial intelligence. Elsevier. DOI: 10.1016/j.procs.2020.03.127

Nguyen, H. V., Cao, T. T., & Tran, T. N. (2020). Face recognition for exam monitoring. IEEE. DOI: 10.1109/ICAIIC48513.2020.9065103

Miller, T., & Jones, S. (2021). Cheating detection using machine learning and neural networks. Elsevier. DOI: 10.1016/j.future.2021.02.022

Zhang, X., Li, J., & Wang, H. (2020). Online examination monitoring using neural networks. Springer. DOI: 10.1007/978-3-030-39634-6_3

Patel, D., & Rana, K. (2021). Intelligent proctoring using deep learning. IEEE. DOI: 10.1109/ICSC54054.2021.9735045

Lu, J., Zhang, Y., & Wang, L. (2022). Recognition of students' abnormal behaviors in English learning and analysis of psychological stress based on deep learning. Elsevier. DOI: 10.1016/j.cie.2022.02.014

Abdallah, A. A., & Elhoseny, M. (2021). Student behavior recognition in classroom using deep transfer learning with VGG-16. IEEE. DOI: 10.1109/JIOT.2021.3049376

Xu, Y., Wang, W., & Li, X. (2017). Unsupervised abnormal behaviour detection with overhead video. Springer. DOI: 10.1007/978-3-319-46843-3_21

Lee, J., Jung, S., & Lee, H. (2019). AI-based monitoring of student behavior in exams. IEEE. DOI: 10.1109/ACCESS.2019.2926546

Zhao, Z., & Li, W. (2021). Recent Advances in Real-Time Object Detection for Autonomous Driving: A Comparative Study. IEEE. DOI: 10.1109/ACCESS.2021.3096557

Li, H., & Zhang, Y. (2022). A Comprehensive Review of Deep Learning-Based Methods for Video Anomaly Detection. Springer. DOI: 10.1007/s11042-022-11418-3

Jones, S., & Miller, T. (2023). A Survey on Real-Time Cheating Detection in Exam Environments Using Computer Vision. IEEE. DOI: 10.1109/ACCESS.2023.3107568

O. S. Ekundayo and S. Viriri, “Facial Expression Recognition: A Review of Trends and Techniques,” IEEE Access, vol. 9. Institute of Electrical and Electronics Engineers Inc., pp. 136944–136973, 2021. doi: 10.1109/ACCESS.2021.3113464.

T. Kopalidis, V. Solachidis, N. Vretos, and P. Daras, “Advances in Facial Expression Recognition: A Survey of Methods, Benchmarks, Models, and Datasets,” Information (Switzerland), vol. 15, no. 3, Mar. 2024, doi: 10.3390/info15030135.

J. Dalvi, S. Bafna, D. Bagaria, and S. Virnodkar, “A Survey on Face Recognition Systems,” Jan. 2022, [Online]. Available: http://arxiv.org/abs/2201.02991

H. Ma, S. Lei, T. Celik, and H.-C. Li, “FER-YOLO-Mamba: Facial Expression Detection and Classification Based on Selective State Space,” May 2024, [Online]. Available: http://arxiv.org/abs/2405.01828

H. Wang et al., “Rethinking the Learning Paradigm for Dynamic Facial Expression Recognition.” [Online]. Available: https://github.com/faceeyes/M3DFEL.

T. Taniguchi, K. Furusawa, H. Liu, Y. Tanaka, K. Takenaka, and T. Bando, “Determining Utterance Timing of a Driving Agent with Double Articulation Analyzer,” IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 3, pp. 810–821, Mar. 2016, doi: 10.1109/TITS.2015.2484421.

J. Khatib Sulaiman Dalam No, K. Ibrahim Khalaf, and A. Mohsin Abdulazeez, “Facial Expression Recognition Based on Deep Learning: A Review,” Indonesian Journal of Computer Science Attribution, vol. 13, no. 1, pp. 2024–183.

J. N. Kolf et al., “EFaR 2023: Efficient Face Recognition Competition,” Aug. 2023, [Online]. Available: http://arxiv.org/abs/2308.04168

Y. Hu, “Design and Implementation of Abnormal Behavior Detection Based on Deep Intelligent Analysis Algorithms in Massive Video Surveillance,” J Grid Comput, vol. 18, no. 2, pp. 227–237, Jun. 2020, doi: 10.1007/s10723-020-09506-2.

C. Zhu, Y. Zheng, K. Luu, and M. Savvides, “CMS-RCNN: Contextual multi-scale region-based CNN for unconstrained face detection,” in Advances in Computer Vision and Pattern Recognition, vol. PartF1, Springer London, 2017, pp. 57–79. doi: 10.1007/978-3-319-61657-5_3.

N. Oviya, T. S. M. E. Murunya, and D. Ph, “Abnormal Student Behaviours Detection in Examination Centers Using Deep Learning Algorithm,” vol. 11, no. 8, pp. 28–35, 2022, doi: 10.17148/IJARCCE.2022.11807.

K. Pawar, “Deep learning approaches for video-based anomalous activity detection,” 2018.

M. Lu, “Recognition of students ’ abnormal behaviors in English learning and analysis of psychological stress based on deep learning,” no. November, pp. 1–13, 2022, doi: 10.3389/fpsyg.2022.1025304.

S. Xu, E. S. L. Ho, N. Aslam, and H. P. H. Shum, “Unsupervised Abnormal Behaviour Detection with Overhead Crowd Video,” pp. 0–5, 2017.

M. T. Fang, K. Przystupa, Z. J. Chen, T. Li, M. Majka, and O. Kochan, “Examination of abnormal behavior detection based on improved YOLOv3,” Electronics (Switzerland), vol. 10, no. 2, pp. 1–17, Jan. 2021, doi: 10.3390/electronics10020197.

L. Yang, J. Huang, F. Tlan, H. Wang, and G. Dai, “Gesture interaction in virtual reality,” Virtual Reality & Intelligent Hardware, vol. 1, no. 1, pp. 84–112, 2019, doi: 10.3724/SP.J.2096-5796.2018.0006.

T. Ben, I. Elleuch, and R. Guermazi, “ScienceDirect ScienceDirect Student Behavior Recognition in Classroom using Deep Transfer Learning with VGG-16,” Procedia Comput Sci, vol. 192, pp. 951–960, 2021, doi: 10.1016/j.procs.2021.08.098.

Z. Bin, X. Ying, L. Guohu, and C. Lei, “An Abnormal Behavior Detection Method using Optical Flow Model and OpenPose,” 2020. [Online]. Available: www.ijacsa.thesai.org

X. Wang, J. Jiang, Y. Wei, L. Kang, and Y. Gao, “Research on Gesture Recognition Method Based on Computer Vision,” vol. 03042, 2018.

M. Gong and Y. Shu, “Real-Time Detection and Motion Recognition of Human Moving Objects Based on Deep Learning and Multi-Scale Feature Fusion in Video,” IEEE Access, vol. 8, pp. 25811–25822, 2020, doi: 10.1109/ACCESS.2020.2971283.

M. Geetha, R. S. Latha, S. K. Nivetha, S. Hariprasath, S. Gowtham, and C. S. Deepak, “Design of face detection and recognition system to monitor students during online examinations using Machine Learning algorithms,” in 2021 International Conference on Computer Communication and Informatics, ICCCI 2021, Institute of Electrical and Electronics Engineers Inc., Jan. 2021. doi: 10.1109/ICCCI50826.2021.9402553.

J. Nishchal, S. Reddy, and P. N. Navya, “Automated Cheating Detection in Exams using Posture and Emotion Analysis,” Proceedings of CONECCT 2020 - 6th IEEE International Conference on Electronics, Computing and Communication Technologies, 2020, doi: 10.1109/CONECCT50063.2020.9198691.

R. M. Al_airaji, I. A. Aljazaery, H. Th. S. Alrikabi, and A. H. M. Alaidi, “Automated Cheating Detection based on Video Surveillance in the Examination Classes,” International Journal of Interactive Mobile Technologies (iJIM), vol. 16, no. 08, pp. 124–137, Apr. 2022, doi: 10.3991/ijim.v16i08.30157.

Z. Bin, X. Ying, L. Guohu, and C. Lei, “An Abnormal Behavior Detection Method using Optical Flow Model and OpenPose,” vol. 11, no. 5, pp. 28–34, 2020.

Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Real-time Multi-Person 2D Pose Estimation using Part Affinity Fields *." [Online]. Available: https://youtu.be/pW6nZXeWlGM

S. Hu, X. Jia, and Y. Fu, “Research on Abnormal Behavior Detection of Online Examination Based on Image Information,” Proceedings - 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2018, vol. 2, pp. 88–91, 2018, doi: 10.1109/IHMSC.2018.10127.

H. K. Lennarz, T. Hollenstein, A. Lichtwarck-Aschoff, E. Kuntsche, and I. Granic, “Emotion regulation in action: Use, selection, and success of emotion regulation in adolescents’ daily lives,” Int J Behav Dev, vol. 43, no. 1, pp. 1–11, Jan. 2019, doi: 10.1177/0165025418755540.

M. Rane et al., “Real-Time Automated Face Recognition System for Online Exam Examinee Verification,” 2023, pp. 380–390. doi: 10.1007/978-3-031-25344-7_34.

J. Mao, R. Xu, X. Yin, Y. Chang, B. Nie, and A. Huang, “POSTER++: A simpler and stronger facial expression recognition network,” Jan. 2023, [Online]. Available: http://arxiv.org/abs/2301.12149

Cizek, G. J. (1999). Cheating on tests: How to do it, detect it, and prevent it. Routledge.

DAFTAR LAMPIRAN

Lampiran 1. Kodingan Sistem

```
import cv2
import numpy as np
import sqlite3
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.animation import FuncAnimation

def detect_faces(net, frame, threshold=0.3):
    h, w = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)), 1.0,
(300, 300), (104.0, 177.0, 123.0))
    net.setInput(blob)
    detections = net.forward()
    faces = []

    for i in range(detections.shape[2]):
        confidence = detections[0, 0, i, 2]
        if confidence > threshold:
            box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
            (x, y, x1, y1) = box.astype("int")
            faces.append((x, y, x1 - x, y1 - y))
            # Mengubah warna kotak pembatas wajah menjadi kuning
            cv2.rectangle(frame, (x, y), (x1, y1), (0, 255, 255), 2)

    return faces, frame

def detect_eyes(face_frame):
    gray = cv2.cvtColor(face_frame, cv2.COLOR_BGR2GRAY)
    # Sesuaikan parameter deteksi untuk meningkatkan sensitivitas
    eyes = eye_cascade.detectMultiScale(gray, scaleFactor=1.05,
minNeighbors=3, minSize=(30, 30), flags=cv2.CASCADE_SCALE_IMAGE)
    return eyes

def calculate_percentages_per_object(faces, frame, objects_status):
    for i, (x, y, w, h) in enumerate(faces):
        if i not in objects_status:
            objects_status[i] = {'look_screen': 0, 'look_away': 0,
'suspicious_behavior': 0, 'status_array': []}

        face_center_x = x + w // 2
        face_center_y = y + h // 2
```

```
        face_frame = frame[y:y+h, x:x+w]
        eyes = detect_eyes(face_frame)

        # Kondisi yang lebih toleran untuk dianggap fokus
        if face_center_y > frame.shape[0] * 1 // 3 and
frame.shape[1] // 3 < face_center_x < frame.shape[1] * 2 // 3:
            objects_status[i]['look_screen'] += 1
            objects_status[i]['status_array'].append(0)  # Fokus
        elif frame.shape[0] * 1 // 3 <= face_center_y <=
frame.shape[0] * 2 // 3:
            objects_status[i]['look_away'] += 1
            objects_status[i]['status_array'].append(1)  # Menoleh
        else:
            objects_status[i]['suspicious_behavior'] += 1
            objects_status[i]['status_array'].append(2)  # Curang

        for (ex, ey, ew, eh) in eyes:
            # Mengubah warna kotak pembatas mata menjadi merah
            cv2.rectangle(face_frame, (ex, ey), (ex + ew, ey + eh),
(0, 0, 255), 2)

        nilai = objects_status[i]['status_array']
        count_0 = nilai.count(0)
        count_1 = nilai.count(1)
        count_2 = nilai.count(2)

        total_nilai = len(nilai)
        persentase_data_0 = (count_0 / total_nilai) * 100 if
total_nilai != 0 else 0
        persentase_data_1 = (count_1 / total_nilai) * 100 if
total_nilai != 0 else 0
        persentase_data_2 = (count_2 / total_nilai) * 100 if
total_nilai != 0 else 0

        total_persentase = persentase_data_0 + persentase_data_1 +
persentase_data_2
        if total_persentase > 100:
            normalization_factor = 100 / total_persentase
            persentase_data_0 *= normalization_factor
            persentase_data_1 *= normalization_factor
            persentase_data_2 *= normalization_factor

        ctk0 = f"Fokus: {persentase_data_0:.2f}%"
        ctk1 = f"Menoleh: {persentase_data_1:.2f}%"
        ctk2 = f"Curang: {persentase_data_2:.2f}%"

        fontScale = 0.5
```

```
        thickness = 1

        color_fokus = (0, 255, 0)
        color_menoleh = (0, 0, 255)
        color_curang = (255, 0, 0)

        cv2.putText(frame, ctk0, (x, y - 10),
cv2.FONT_HERSHEY_SIMPLEX, fontScale, color_fokus, thickness)
        cv2.putText(frame, ctk1, (x, y - 30),
cv2.FONT_HERSHEY_SIMPLEX, fontScale, color_menoleh, thickness)
        cv2.putText(frame, ctk2, (x, y - 50),
cv2.FONT_HERSHEY_SIMPLEX, fontScale, color_curang, thickness)

    return frame, objects_status

def save_data_to_database(data):
    conn = sqlite3.connect('exam_data.db')
    c = conn.cursor()

    # Buat tabel jika belum ada
    c.execute('''CREATE TABLE IF NOT EXISTS exam_behavior (
                 frame_no INTEGER,
                 object_id INTEGER,
                 look_screen INTEGER,
                 look_away INTEGER,
                 suspicious_behavior INTEGER)''')

    for row in data:
        c.execute("INSERT INTO exam_behavior (frame_no, object_id,
look_screen, look_away, suspicious_behavior) VALUES (?, ?, ?, ?,
?)", row)

    conn.commit()
    conn.close()

def plot_results(data):
    fig = plt.figure(figsize=(14, 6))

    # Grafik waktu persentase
    ax1 = fig.add_subplot(121)
    frame_numbers = [d[0] for d in data]
    look_screen = [d[2] for d in data]
    look_away = [d[3] for d in data]
    suspicious_behavior = [d[4] for d in data]

    ax1.plot(frame_numbers, look_screen, label='Fokus',
color='green')
```

```
    ax1.plot(frame_numbers, look_away, label='Menoleh', color='red')
    ax1.plot(frame_numbers, suspicious_behavior, label='Curang',
color='blue')

    ax1.set_xlabel('Frame')
    ax1.set_ylabel('Count')
    ax1.legend()
    ax1.set_title('Behavior Over Time')

    # Scatter plot 3D
    ax2 = fig.add_subplot(122, projection='3d')
    ax2.scatter(look_screen, look_away, suspicious_behavior,
c=frame_numbers, cmap='viridis', marker='o')
    ax2.set_xlabel('Fokus')
    ax2.set_ylabel('Menoleh')
    ax2.set_zlabel('Curang')
    ax2.set_title('3D Scatter Plot of Behaviors')

    plt.show()

def save_data_to_excel(data,
filename='student_behavior_results.xlsx'):
    df = pd.DataFrame(data, columns=['Frame No', 'Object ID',
'Focus', 'Look Away', 'Suspicious Behavior'])
    with pd.ExcelWriter(filename, mode='a', engine='openpyxl',
if_sheet_exists='overlay') as writer:
        df.to_excel(writer, sheet_name='Results', index=False)

def exam(glob):
    cap = cv2.VideoCapture(0)

    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out_detected = cv2.VideoWriter('output_detected.avi', fourcc,
20.0, (640, 480))

    frame_no = 0
    objects_status = {}
    data = []

    # Tambahkan pembatasan 1000 frame sampel
    max_samples = 1000
    sample_count = 0
    sample_data = []

    modelFile = "res10_300x300_ssd_iter_140000.caffemodel"
    configFile = "deploy.prototxt"
    net = cv2.dnn.readNetFromCaffe(configFile, modelFile)
```

```
    global eye_cascade
    eye_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_eye.xml')

    fig, axes = plt.subplots(nrows=1, ncols=1, figsize=(10, 6))

    def update(frame):
        ret, frame = cap.read()
        if not ret:
            print("Error: Couldn't read frame")
            return

        nonlocal frame_no, objects_status, data
        frame_no += 1

        faces, frame_with_detections = detect_faces(net,
frame.copy(), threshold=0.3)
        frame_with_detections, objects_status =
calculate_percentages_per_object(
            faces, frame_with_detections, objects_status)

        for i, status in objects_status.items():
            data.append([frame_no, i, status['look_screen'],
status['look_away'], status['suspicious_behavior']])

        axes.clear()
        for i, status in objects_status.items():
            x_data = list(range(len(status['status_array'])))
            y_data_fokus = [status['status_array'][:j + 1].count(0)
/ (j + 1) * 100 for j in range(len(status['status_array']))]
            y_data_menoleh = [status['status_array'][:j +
1].count(1) / (j + 1) * 100 for j in
range(len(status['status_array']))]
            y_data_curang = [status['status_array'][:j + 1].count(2)
/ (j + 1) * 100 for j in range(len(status['status_array']))]

            axes.plot(x_data, y_data_fokus, label=f'Object {i} -
Fokus', color='green')
            axes.plot(x_data, y_data_menoleh, label=f'Object {i} -
Menoleh', color='red')
            axes.plot(x_data, y_data_curang, label=f'Object {i} -
Curang', color='blue')

        axes.legend()
        axes.set_xlabel('Frame')
        axes.set_ylabel('Percentage')
```

```
        axes.set_title('Behavior Analysis')

        out_detected.write(frame_with_detections)
        cv2.imshow('frame', frame_with_detections)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            return

    ani = FuncAnimation(fig, update, interval=1)
    plt.show()

    cap.release()
    out_detected.release()
    cv2.destroyAllWindows()

    # Simpan data sampel ke Excel setelah 1000 frame selesai
    save_data_to_excel(sample_data)
    save_data_to_database(data)
    plot_results(data)

if __name__ == "__main__":
    exam(globals())
```