



## Perbandingan Metode Deteksi Wajah Menggunakan OpenCV Haar Cascade, OpenCV Single Shot Multibox Detector (SSD) dan DLib CNN

Lia Farokhah

Prodi Teknik Informatika, fakultas Teknologi dan Desain, Institut Teknologi dan Bisnis ASIA Malang  
farokhah@asia.ac.id

### Abstract

Comparison of methods in face detection is needed to provide recommendation of best method. This study compared three methods in face detection, namely OpenCV haar cascade, OpenCV Single Shot Multibox Detector (SSD) and Dlib CNN. Face detection is focused on five challenging conditions, namely face detection in head position obstacles, wearing face masks, lighting, background images that have a lot of noise, differences in expression. Data testing is taken randomly on google with reference to one image consisting of more than one detected face with wild condition. The results of the comparative analysis in wild condition show that the OpenCV haar cascade has more weaknesses with a performance percentage of 20% compared OpenCV SSD and Dlib CNN method. Performance results of SSD and Dlib CNN have the same performance in the five conditions tested, which is about 80%.

Keywords: Face detection, Open CV, Dlib, Performance comparison

### Abstrak

Perbandingan metode dalam deteksi wajah sangat dibutuhkan untuk memberi gambaran rekomendasi pemakaian metode terbaik. Penelitian ini membandingkan tiga metode dalam deteksi wajah yaitu OpenCV Haar Cascade, OpenCV Single Shot Multibox Detector (SSD) dan Dlib CNN. Deteksi wajah difokuskan pada lima kondisi tantangan yang ada yaitu deteksi wajah pada halangan posisi kepala, memakai masker wajah, pencahayaan, latar gambar yang memiliki banyak noise, perbedaan ekspresi. Data uji diambil secara acak di google dengan acuan satu gambar lebih dari satu wajah yang dideteksi dengan posisi liar. Hasil analisis perbandingan uji gambar pada kondisi liar menunjukkan bahwa OpenCV haar cascade memiliki lebih banyak kelemahan dengan prosentase kinerja 20% dibandingkan deteksi wajah menggunakan OpenCV SSD dan Dlib CNN. Hasil Kinerja SSD dan Dlib CNN memiliki kinerja yang sama dalam lima kondisi yang diuji yaitu sekitar 80%.

Kata kunci: Deteksi wajah, Open CV, Dlib, Perbandingan kinerja

### 1. Pendahuluan

Deteksi wajah berperan penting pada proses pengenalan wajah[1][2], pengenalan ekspresi wajah[3][4] dan berbagai aplikasi interaksi manusia dan komputer[5]. Metode deteksi wajah berkembang sangat cepat. Metode Haar cascade viola jones merupakan metode tertua yang paling banyak dipakai sampai sekarang. Metode ini diklaim masih relavan dan mampu mendeteksi wajah dengan baik[6][7][8][9][10].

Selain itu, metode yang cukup terkenal adalah single shot box detector (SSD). SSD merupakan salah satu metode deteksi objek yang dikembangkan untuk menutupi kelemahan haar cascade. Metode ini sudah diimplementasikan pada deteksi wajah dengan halangan kacamata hitam, syal, alat pernafasan, topi, topeng,

kalung, masker[11], SSD mampu menunjukkan kinerja dengan baik.

Metode deteksi wajah lainnya menggunakan dlib library. Dlib CNN dianggap cukup tajam dan menghasilkan deteksi wajah terbaik dibandingkan metode sebelumnya[12]. Penelitian ini akan membandingkan kinerja ketiga algoritma deteksi wajah dengan berbagai kondisi liar atau wild condition. Deteksi wajah pada kondisi liar lebih mendekati implementasi dunia nyata daripada deteksi wajah yang terlihat dari depan dan tanpa halangan apapun atau lab controlled.

Penelitian yang dilakukan sebelumnya banyak berfokus pada deteksi satu wajah dalam satu gambar. Pada kehidupan nyata, implementasi deteksi wajah dengan

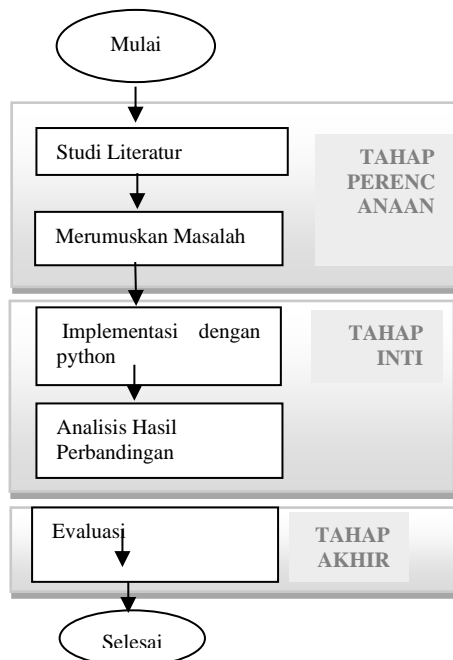
banyak orang dalam satu gambar merupakan tantangan tersendiri. Selain itu, penelitian sebelumnya banyak berfokus pada implementasi suatu algoritma tanpa perbandingan atau melakukan perbandingan dalam satu atau dua kondisi saja. Beberapa kondisi yang dianalisis pada penelitian ini adalah deteksi wajah pada halangan posisi kepala, memakai masker wajah, pencahayaan, latar gambar yang memiliki banyak gangguan atau *noise*, perbedaan ekspresi wajah yang sangat beragam. Perbandingan algoritma ini cukup penting untuk memberikan gambaran dan rekomendasi algoritma dalam deteksi wajah.

## 2. Metode Penelitian

### 2.1 Alur Penelitian

Alur penelitian dibagi menjadi tiga yaitu tahap perencanaan, tahap inti dan tahap akhir. Pada tahap perencanaan atau tahap pertama, dilakukan studi literatur tentang penggunaan ketiga metode dalam deteksi wajah. Adapun beberapa hal yang dicari adalah kinerja ketiga metode dan Kelemahan alur publikasi yang sudah ada.

Pada tahap kedua atau tahap inti, dilakukan uji coba deteksi gambar dengan beberapa wajah dalam satu gambar menggunakan *python*. Pada tahap kedua ini, deteksi dilakukan dalam lima kondisi yaitu deteksi wajah pada halangan posisi kepala, memakai masker wajah, pencahayaan, latar gambar yang memiliki banyak *noise*, perbedaan ekspresi. Pada tahap ketiga atau tahap evaluasi, dilakukan evaluasi untuk analisis akhir. Bagan Alur Penelitian ditunjukkan pada Gambar 1.



Gambar 1. Alur Penelitian

### 2.2 Data Uji atau Testing

Adapun data gambar uji pada kelima kondisi diambil secara acak di *google* dengan memadukan tantangan apa saja yang biasa dihadapi dalam deteksi wajah. Dalam pengambilan gambar uji, diambil gambar dengan jumlah orang lebih dari satu dalam satu gambar. Beberapa tantangan yang diambil disesuaikan dengan tantangan umum penelitian dibidang klasifikasi yaitu *viewpoint variation, scale variation, deformation, occlusions, illumination, background clutter* [12].

Tantangan dalam klasifikasi yang diusulkan akan dievaluasi menggunakan tiga metode ini yaitu OpenCV haar cascade, OpenCV SSD dan Dlib CNN dilakukan dengan cara menganalisis jumlah wajah yang terdeteksi dalam satu gambar. Secara garis besar, metode ini membandingkan library OpenCV dan Dlib. Tahap berikutnya adalah menganalisis kegagalan pada masing-masing metode. Langkah terakhir adalah mengambil kesimpulan dari uji coba perbandingan metode.

## 3. Hasil dan Pembahasan

Metode deteksi wajah mengambil peran penting dalam kesuksesan implementasi aplikasi yang berhubungan dengan wajah sebagai pusat analisis. Hasil kinerja dari deteksi wajah akan dibawah kedalam proses selanjutnya seperti penentuan landmark wajah, pengenalan wajah, pengenalan ekspresi wajah. Kegagalan dalam deteksi wajah menjadi kegagalan awal menganalisis secara komprehensif.

Pada algoritma *Haar cascade*, Ada dua indikator penting yaitu *scale factor* dan *minNeighbours*. Pada *scale factor* menggunakan 1,05 dan *minNeighbours* 5. Deteksi wajah menggunakan *single shot multibox detector* memiliki dua indikator penting yaitu melatih model terlebih dahulu atau *pre-trained*. *Pre-trained* model merupakan salah satu langkah mengambil model yang sudah diuji sebelumnya oleh orang lain kedalam data yang cukup besar seperti *dataset ImageNet*. *Pre-trained* model merupakan *dataset* yang sudah diakui sehingga kualitas *pre-trained* model sudah sangat baik Pada percobaan ini menggunakan *framework caffe model*.

Selain itu, Pada SSD ada file untuk mendefinisikan arsitektur model. Metode SSD merupakan metode yang dianggap cukup efisien dimana lokalisasi dan deteksi dilakukan secara bersamaan[12]. algoritma yang menggunakan proses terpisah sangat tidak efisien karena meningkatkan waktu training dan berkurangnya akurasi. Adapun Tren perkembangan arsitektur *deep learning* untuk deteksi objek mengarah pada arsitektur yang lebih cepat dan sistem deteksi yang lebih efisien.

Deteksi wajah menggunakan Dlib CNN memiliki indikator penting yaitu *pre-trained*. Pada dlib

merepresentasikan kotak pembatas melalui objek persegi panjang dengan properti kiri, atas, kanan, dan bawah. Dlib CNN dianggap bisa mendeteksi dari segala sudut pandang[13]. Adapun beberapa indikator penting pada kode python masing-masing metode sebagai berikut:

---

#### Program Indikator Penting Pada Haar Cascade

---

```

""" # load the input image from disk, resize
it, and convert it to
# grayscale
image = cv2.imread(args["image"])
image = imutils.resize(image, width=500)
gray = cv2.cvtColor(image,
cv2.COLOR_BGR2GRAY)

# detect faces in the input image using the
haar cascade face
# detector
print("[INFO] performing face
detection...")
rects = detector.detectMultiScale(gray,
scaleFactor=1.05,
minNeighbors=5, minSize=(30, 30),
flags=cv2.CASCADE_SCALE_IMAGE)
print("[INFO] {} faces
detected...".format(len(rects)))

# loop over the bounding boxes
for (x, y, w, h) in rects:
# draw the face bounding box on the image
cv2.rectangle(image, (x, y), (x + w, y + h),
(0, 255, 0), 2)
...

```

---

#### Program Indikator Penting Pada SSD

---

```

""" cv2.dnn.readNetFromCaffe(args["prototxt"],
args["model"])

# load the input image and construct an
input blob for the image
# by resizing to a fixed 300x300 pixels and
then normalizing it
image = cv2.imread(args["image"])
(h, w) = image.shape[:2]
blob =
cv2.dnn.blobFromImage(cv2.resize(image,
(300, 300)), 1.0,
(300, 300), (104.0, 177.0, 123.0))

# pass the blob through the network and
obtain the detections and
# predictions
print("[INFO] computing object
detections...")
net.setInput(blob)
detections = net.forward()

# loop over the detections
for i in range(0, detections.shape[2]):
# extract the confidence (i.e.,
probability) associated with the
# prediction
confidence = detections[0, 0, i, 2]

# filter out weak detections by ensuring
the confidence is
# greater than the minimum confidence
if confidence > args["confidence"]:
# compute the (x, y)-coordinates of the
bounding box for the
# object

```

```

box = detections[0, 0, i, 3:7] *
np.array([w, h, w, h])
(startX, startY, endX, endY) =
box.astype("int")

# draw the bounding box of the face
along with the associated
# probability
text = "{:.2f}%".format(confidence *
100)
y = startY - 10 if startY - 10 > 10 else
startY + 10
cv2.rectangle(image, (startX, startY),
(endX, endY),
(0, 0, 255), 2)
cv2.putText(image, text, (startX, y),
cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0,
255), 2)...

```

---

#### Program Indikator Penting Pada Dlib CNN

---

```

""" # load dlib's CNN face detector
print("[INFO] loading CNN face
detector...")
detector =
dlib.cnn_face_detection_model_v1(args["model"])

# load the input image from disk, resize
it, and convert it from
# BGR to RGB channel ordering (which is what
dlib expects)
image = cv2.imread(args["image"])
image = imutils.resize(image, width=600)
rgb = cv2.cvtColor(image,
cv2.COLOR_BGR2RGB)

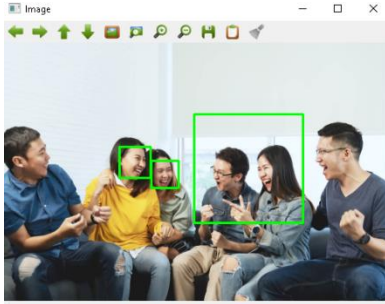
# perform face detection using dlib's face
detector
start = time.time()
print("[INFO] performing face detection
with dlib...")
results = detector(rgb, args["upsample"])
end = time.time()
print("[INFO] face detection took {:.4f}
seconds".format(end - start))

# convert the resulting dlib rectangle
objects to bounding boxes,
# then ensure the bounding boxes are all
within the bounds of the
# input image
boxes = [convert_and_trim_bb(image, r.rect)
for r in results]

# loop over the bounding boxes
for (x, y, w, h) in boxes:
# draw the bounding box on our image
cv2.rectangle(image, (x, y), (x + w, y + h),
(0, 255, 0), 2)...

```

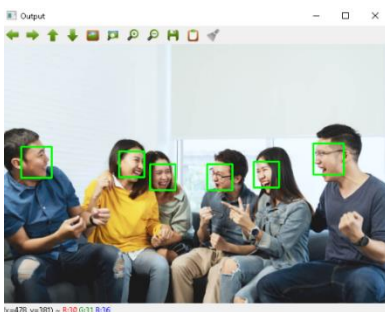
Hasil deteksi wajah dengan kondisi pertama yaitu posisi pose kepala yang bervariasi ditunjukkan pada Gambar 2- Gambar 4. Hasil menunjukkan bahwa pada metode haar cascade memiliki dua wajah yang tidak terdeteksi dengan baik dengan posisi pose kepala dengan menampakkan setengah wajah depan sekitar 90 derajat. Selain itu Ada dua wajah yang terdeteksi dalam satu kotak. Disisi lain, deteksi wajah dengan kondisi pertama menggunakan metode SSD dan Dlib CNN bisa mendeteksi semua wajah secara sempurna.



Gambar 2. Haar Cascade dengan perbedaan posisi kepala

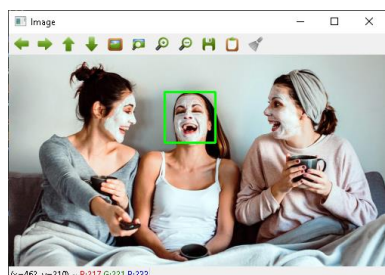


Gambar 3. SSD dengan perbedaan posisi kepala



Gambar 4. Dlib CNN dengan perbedaan posisi kepala

Hasil deteksi wajah dengan kondisi kedua yaitu pemakaian masker wajah ditunjukkan pada Gambar 5- Gambar 7. Hasil menunjukkan bahwa pada metode haar cascade menemui kendala yang cukup signifikan. Dari tiga wajah yang akan dideteksi, ada dua wajah yang tidak terdeteksi dengan baik. Dua wajah yang tidak terdeteksi dengan baik memiliki halangan pada pose kepala sekitar 90 derajat dari arah wajah depan. Disisi lain, deteksi wajah dengan menggunakan metode SSD dan Dlib CNN bisa mendeteksi semua wajah secara sempurna.



Gambar 5. Haar Cascade memakai masker wajah

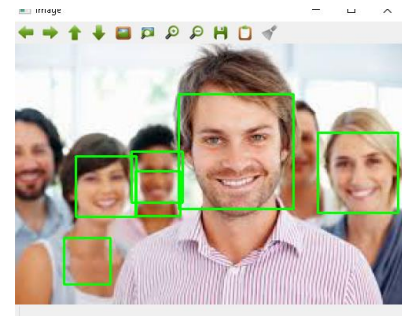


Gambar 6. SSD memakai masker wajah



Gambar 7. Dlib CNN memakai masker wajah

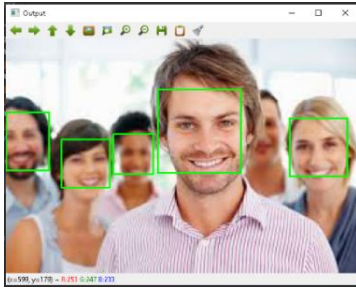
Hasil deteksi wajah dengan kondisi ketiga yaitu pencahayaan. Ada beberapa yang tidak jelas atau blur ditunjukkan pada Gambar 8 sampai Gambar 10. Hasil menunjukkan bahwa pada metode haar cascade menemui kendala yang cukup signifikan. ada dua wajah yang tidak terdeteksi dengan baik. Disisi lain, terjadi dua kesalahan kotak deteksi. Deteksi wajah dengan menggunakan metode SSD dan Dlib CNN bisa mendeteksi lima wajah dari 6 yang seharusnya. Satu wajah tidak terdeteksi karena tertutupi objek sehingga terlihat hanya sedikit atau tidak utuh.



Gambar 8. Haar Cascade dengan halangan pencahayaan blur



Gambar 9. SSD dengan halangan pencahayaan blur



Gambar 10. Dlib CNN dengan halangan pencahayaan blur

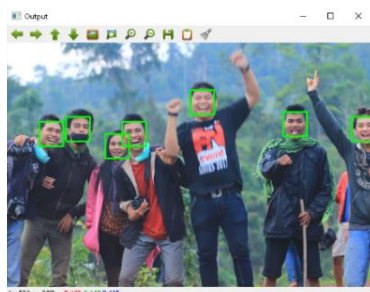
Hasil deteksi wajah dengan kondisi keempat yaitu memiliki kondisi latar yang ramai atau *noise*. ditunjukkan pada Gambar 11- Gambar 13. Hasil menunjukkan bahwa pada metode haar cascade ada satu wajah tidak terdeteksi dan ada kesalahan kotak deteksi.. Deteksi wajah dengan menggunakan metode SSD dan Dlib CNN bisa mendeteksi mendeteksi dengan sempurna



Gambar 11. Haar Cascade dengan noise background



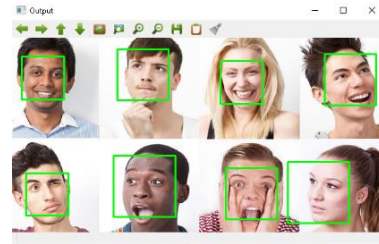
Gambar 12. SSD dengan noise background



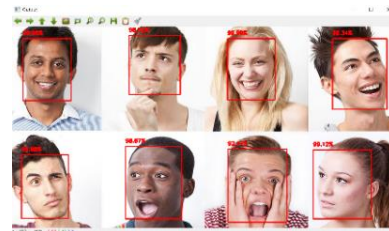
Gambar 13. Dlib CNN dengan noise background

Hasil deteksi wajah dengan kondisi kelima yaitu terdapat beragam ekspresi wajah ditunjukkan pada Gambar 14 - Gambar 16. Hasil analisis menunjukkan

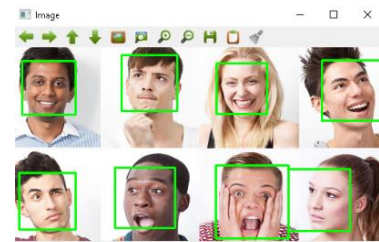
bahwa semua metode yang dibandingkan mampu mendeteksi secara sempurna dengan perbedaan ekspresi yang ada. Pada pengujian kelima ada dua kondisi yaitu perbedaan ekspresi dan pose kepala.



Gambar 14. Haar Cascade dengan beragam ekspresi wajah



Gambar 15. SSD dengan beragam ekspresi wajah



Gambar 16. Dlib CNN dengan beragam ekspresi wajah

Hasil Kinerja dari kelima kondisi dalam deteksi wajah pada kondisi liar atau *wild condition* ditunjukkan pada Tabel 1

Pada tabel 1, Hasil kinerja deteksi wajah menggunakan haar cascade dari lima data uji hanya satu yang terdeteksi dengan sempurna sehingga memiliki akurasi sekitar 20% . Pada deteksi wajah menggunakan SSD dan dlib CNN memiliki lima kondisi data uji, ada empat yang sesuai sehingga prosentase akurasi sekitar 80% dari lima kondisi uji. Hasil analisis deteksi wajah dari kelima kondisi yaitu deteksi wajah pada halangan posisi kepala, memakai masker wajah, pencahayaan, latar gambar yang memiliki banyak *noise*, perbedaan ekspresi memiliki hasil yang sangat rendah pada algoritma *haar cascade*. Algoritma SSD dan Dlib CNN memiliki hasil yang sama.

Algoritma *haar cascade* alam deteksi wajah memiliki banyak kelemahan. Pada lima percobaan, hanya satu percobaan pada variasi ekspresi wajah yang bisa dideteksi dengan baik. Pada Algoritma SSD dan Dlib

CNN dari lima percobaan kondisi, empat kondisi bisa bagus dan wajah tertutupi sehingga hanya tampak dideteksi dengan baik. sedikit.

Tabel 1 Hasil Kinerja Deteksi Wajah Dalam Berbagai Kondisi

| Kondisi Foto   | Har cascade   | SSD                       | Dlib CNN                  |
|--|---|---------------------------|---------------------------|
| Variasi posisi kepala atau sudut pandang                       | Terdeteksi 3 dari 6 orang dan ada dua orang terdeteksi dalam satu kotak | Terdeteksi semua          | Terdeteksi semua          |
| Halangan Benda lain atau masker ditambah variasi posisi kepala | Terdeteksi 1 dari 3 orang   | Terdeteksi semua          | Terdeteksi semua          |
| Pencahayaan  | Terdeteksi 6 dari 6 orang tetapi ada dua kesalahan deteksi              | Terdeteksi 5 dari 6 orang | Terdeteksi 5 dari 6 orang |
| Gangguan Background  | Terdeteksi 7 dari 7 orang tetapi ada satu yang salah deteksi            | Terdeteksi semua          | Terdeteksi semua          |
| Perubahan bentuk wajah karena variasi ekspresi                 | Terdeteksi semua  | Terdeteksi semua          | Terdeteksi semua          |

#### 4. Kesimpulan

Adapun simpulan dari penelitian ini adalah deteksi wajah menggunakan *haar cascade* memiliki lebih banyak kelemahan dari 5 data uji hanya 1 yang sesuai sehingga memiliki prosentase kinerja 20% dibandingkan deteksi wajah menggunakan SSD dan Dlib CNN. Hasil Kinerja SSD dan Dlib CNN memiliki kinerja yang sama dalam lima kondisi liar yang diuji, ada empat yang sesuai sehingga prosentase sekitar 80%. Deteksi banyak wajah dalam satu gambar memiliki kelemahan paling besar pada pencahayaan yang blur dan wajah tampak sebagian. Penelitian selanjutnya diarahkan untuk membandingkan algoritma lain dalam menangani deteksi wajah dengan kondisi pencahayaan yang kurang

#### Daftar Rujukan

- [1] D. Suprianto, "Sistem Pengenalan Wajah Secara Real-Time," *Sist. Pengenalan Wajah Secara Real-Time dengan Adab. Eig. PCA MySQL*, vol. 7, no. 2, pp. 179–184, 2013.
- [2] R. Wiryadinata, R. Sagita, S. Wardoyo, and Priswanto, "Face Recognition on Attendance System Using Method Of Dynamic Times, Principal Component Analysis, and Gabor Wavelet," *Issn 1858-3075*, vol. 12, no. 1, pp. 1–8, 2016.
- [3] L. Farokhah, "Implementasi Convolutional Neural Network untuk Klasifikasi Variasi Intensitas Emosi pada Dynamic Image Sequence," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 6, pp. 1070–1076, 2020, doi: 10.29207/resti.v4i6.2644.
- [4] S. Li and W. Deng, "Deep Facial Expression Recognition: A Survey," *IEEE Trans. Affect. Comput.*, vol. 3045, no. c, pp. 1–1, 2020, doi: 10.1109/taffc.2020.2981446.
- [5] K. Takayama, H. Johan, T. Nishita, and Takayama et al., "Face detection and face recognition of cartoon characters using feature extraction," *Image, Electron. Vis. Comput. Work.*, p. 48, 2012.
- [6] S. Zafeiriou, C. Zhang, and Z. Zhang, "A survey on face detection in the wild: Past, present and future," *Comput. Vis. Image Underst.*, vol. 138, no. March, pp. 1–24, 2015, doi: 10.1016/j.cviu.2015.03.015.
- [7] M. Fransiska Sisilia, F. Lia, and A. Nur Lailatul, "Pemodelan sistem deteksi wajah sebagai penghitung jumlah penumpang transportasi publik," *J. Resist. (Rekayasa Sist. Komputer)*, vol. 4, no. 1, pp. 67–77, 2021.
- [8] S. Abidin, "Deteksi Wajah Menggunakan Metode Haar Cascade Classifier Berbasis Webcam Pada Matlab," *J. Teknol. Elektroika*, vol. 15, no. 1, p. 21, 2018, doi: 10.31963/elektroika.v15i1.2102.
- [9] A. Priadana and M. Habibi, "Face detection using haar cascades to filter selfie face image on instagram," *Proceeding - 2019 Int. Conf. Artif. Intell. Inf. Technol. ICAIIT 2019*, pp. 6–9, 2019, doi: 10.1109/ICAIIIT.2019.8834526.
- [10] T. Mantoro, M. A. Ayu, and Suhendi, "Multi-Faces Recognition Process Using Haar Cascades and Eigenface Methods," *Int. Conf. Multimed. Comput. Syst. -Proceedings*, vol. 2018-May, pp. 1–5, 2018, doi: 10.1109/ICMCS.2018.8525935.
- [11] X. Ziwei et al., "Face Occlusion Detection Based on SSD Algorithm," *ICEIEC 2020 - Proc. 2020 IEEE 10th Int. Conf. Electron. Inf. Emerg. Commun.*, pp. 362–365, 2020, doi: 10.1109/ICEIEC49280.2020.9152335.
- [12] A. Rosebrock, *Deep Learning for Computer Vision With Python*, 3rd ed. United States of America: PyImageSearch, 2018.
- [13] P. Arun, "CNN based face detector from dlib," *Toward Data Science*, 2018. <https://towardsdatascience.com/cnn-based-face-detector-from-dlib-c3696195e01c> (accessed May 21, 2021).