

PENGENALAN WAJAH DAN DETEKSI KANTUK MENGGUNAKAN METODE HAAR CASCADE DAN CONVOLUTIONAL NEURAL NETWORK

(Face Recognition and Drowsiness Detection using Haar Cascade and Convolutional Neural Network Method)

Farel Hariesugama^[1], Fitri Bimantoro^[1], Gibran Satya Nugraha^[1]

^[1]Dept Informatics Engineering, Mataram University

Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: hariesugama@gmail.com, [bimo, gibransn]@unram.ac.id

Abstract

Drowsiness is an action that is felt when you want to fall asleep. Drowsiness or the feeling of wanting to fall asleep is caused by fatigue or exhaustion felt by the body and this can ruin all the activities you want to undertake. Drowsiness is a psychological and physiological change that a person experiences when carrying out tasks that require a high level of concentration and special attention for a long period of time. This research aims to implement a machine learning algorithm in a facial recognition and drowsiness detection system. The stages carried out to develop the system in this research consisted of two stages. The first stage, data processing and model creation for open and closed eye conditions using Convolutional Neural Network (CNN). The second stage detects faces using Haar Cascade using a model that has been developed. The results of trials using dropout obtained a high score, namely an accuracy of 0.97 and no overfitting was found and the selected Convolutional Neural Network (CNN) model was able to detect sleepiness in real time with fairly good accuracy based on different facial conditions and facial characteristics, namely on detection of the first to fourth correspondents, the lowest accuracy value was obtained for the fourth correspondent at a distance of 50 cm with an accuracy value of 30% and the highest accuracy value was obtained for all correspondences at a distance of 30 cm with an accuracy value of 80%.

Keywords: *Drowsiness, Face Recognition, Drowsiness Detection, Haar Cascade, CNN.*

1. PENDAHULUAN

Mengantuk adalah suatu tindakan yang dirasakan saat ingin tertidur. Mengantuk atau rasa ingin tertidur disebabkan oleh kelelahan atau keletihan yang dirasakan oleh tubuh dan hal ini dapat merusak semua kegiatan yang ingin dijalani. Mengantuk merupakan perubahan psikologis dan fisiologis yang dialami seseorang saat melakukan tugas yang membutuhkan konsentrasi tingkat tinggi dan perhatian khusus untuk jangka waktu yang cukup lama[1].

Mengantuk dapat memiliki dampak negatif terutama saat seseorang melakukan tugas-tugas yang memerlukan tingkat konsentrasi yang tinggi seperti mengemudi, bekerja dengan mesin berat, atau menjalani pekerjaan berbahaya. Ketidakmampuan untuk tetap terjaga dan fokus saat mengantuk dapat meningkatkan risiko kecelakaan dan kesalahan serius. Statistik menunjukkan bahwa banyak kecelakaan lalu

lintas disebabkan oleh pengemudi yang mengantuk. Hanya dengan beberapa detik rasa kantuk dirasakan maka akan berakibat fatal, rasa kantuk beberapa detik ini disebut microsleep. Durasi microsleep sangat singkat yaitu 1 detik hingga 30 detik[2].

Untuk mengatasi masalah ini, deteksi kantuk menjadi penting. Dengan menggunakan teknologi seperti pemantauan visual dan pengenalan wajah dapat mendeteksi tanda-tanda fisik ketika seseorang mulai mengantuk seperti mata yang terpejam atau berkedip secara cepat. Pendeteksian ini dapat membantu memperingatkan pengemudi atau pekerja agar mereka bisa melakukan tindakan pencegahan seperti beristirahat sejenak atau minum segelas air. Dengan pencegahan dapat meningkatkan keselamatan dan produktivitas di lingkungan kerja dan dalam aktivitas sehari-hari. Mengantuk menjadi salah satu faktor dimana banyak kasus kecelakaan berkendara,

hal ini disebabkan rasa kantuk yang membuat pengemudi tidak bisa fokus dalam berkendara dan sepertiga kasus kecelakaan lalu lintas di dunia disebabkan oleh kesalahan pengemudi[3]. Kecelakaan berkendara hanyalah salah satu dari efek mengantuk, adapun hal lain yang menjadi efek dari mengantuk yaitu tidak dapat pada pekerjaan, pembelajaran menjadi tidak kondusif, dan lain-lain. Penelitian ini dirancang untuk membantu dalam mengenali kantuk pada seseorang dan dapat diaplikasikan pada kegiatan yang membutuhkan pendeteksian kantuk seperti mendeteksi kantuk pada pengendara, deteksi kantuk pada siswa saat pembelajaran, dan kegiatan yang membutuhkan konsentrasi tinggi, dan lain-lain. Deteksi kantuk yang dilakukan pada penelitian ini berfokus pada deteksi kantuk berdasarkan kondisi mata dengan menggunakan metode Haar Cascade dan Convolutional Neural Network (CNN) untuk pembuatan model yang berfokus pada penggunaan Dropout yang diharapkan mendapatkan hasil komputasi lebih cepat dan nilai akurasi yang tinggi.

Dalam kasus ini sudah banyak studi yang dilakukan mengenai keselamatan berkendara dan deteksi kantuk menjadi salah satu studi yang dikembangkan seperti pada penelitian deteksi kantuk untuk keselamatan berkendara menggunakan metode Convolutional Neural Networks (CNN). Pada penelitian tersebut menggunakan metode CNN untuk membuat model yang terdiri dari beberapa layer konvolusi dan pooling yang dilatih menggunakan dataset gambar pengemudi mengantuk dan tidak mengantuk yang diambil melalui video sepanjang 9,5 jam dengan resolusi 640x480 pada 30 fps. Model CNN diimplementasikan pada aplikasi android untuk mendeteksi kantuk. Hasil pengujian model mendapatkan hasil akurasi dengan rata-rata 83.33% dari semua kategori yaitu dengan kacamata, malam hari tanpa kacamata, malam hari dengan kacamata, tanpa kacamata, dan dengan kacamata hitam[4]. Penelitian terkait juga diterapkan pada sistem presensi menggunakan metode deteksi wajah. Pada penelitian tersebut menggunakan Haar Cascade untuk mendeteksi dan sebagai metode presensi siswa. Uji coba dilakukan dengan cara mengambil gambar wajah siswa beresolusi 96x96 px sebanyak 28 kelas dan dilakukan deteksi wajah sebagai tanda telah melakukan presensi untuk setiap siswa. Hasil uji coba dari 28 kelas dataset didapatkan nilai akurasi 95% dengan variasi pencahayaan dan jarak dari objek menuju kamera[5].

Pada penelitian uji coba deteksi kantuk dan alarm secara real-time berdasarkan kondisi mata menggunakan gabungan dari metode Haar Cascade

Dan Convolutional Neural Network (CNN). Pada penelitian tersebut metode yang digunakan adalah Haar Cascade untuk deteksi wajah dan Convolutional Neural Network (CNN) untuk mencari hasil rata-rata akurasi model Convolutional Neural Network (CNN) dalam mendeteksi kantuk. Uji coba dilakukan dengan dataset sebanyak 210 gambar, 110 gambar mata tertutup dan 110 gambar mata terbuka beresolusi 24x24 pixel. Hasil uji coba akurasi Haar Cascade dalam mendeteksi wajah adalah 100% dan rata-rata akurasi model Convolutional Neural Network (CNN) dalam mendeteksi kantuk mencapai 97,23% dengan waktu komputasi 0.2975 detik[6].

Metode untuk pengenalan wajah yang digunakan adalah Haar Cascade dikarenakan metode ini memiliki respon komputasi yang cepat daripada metode lain[7]. Hal ini dikuatkan dengan proses yang dilakukan pada metode Haar Cascade. Proses tersebut adalah proses preprocessing sebelum masuk kedalam fitur haar-like, proses preprocessing yang dilakukan yaitu dengan melakukan grayscale untuk mengubah citra menjadi gambar keabuan dan histogram equalization untuk memperbaiki kontras gambar pada citra yang akan diuji[8]. Metode yang digunakan untuk mendeteksi kantuk adalah Convolutional Neural Network, kelebihan metode ini adalah nilai hasil akurasi yang relatif tinggi. CNN memiliki potensi karena CNN memiliki keunggulan dalam masalah akurasi dan masalah dataset besar daripada metode yang lain untuk mendeteksi kantuk[8].

Perbedaan CNN dengan metode lain seperti metode Hidden Markov Model (HMM) dan Support Vector Machine (SVM). Pada metode HMM dengan parameter kondisi mata terbuka dan tertutup memiliki kelebihan dalam ukuran dataset dan kelemahan terbatas pada struktur dasar. Pada metode SVM dengan parameter kondisi mata terbuka dan tertutup memiliki kelebihan pada generalisasi objek menjadi bagus tetapi memiliki kelemahan pada dataset yaitu tidak cocok untuk dataset besar. Pada metode CNN dengan parameter kondisi mata terbuka dan tertutup memiliki kelebihan pada nilai akurasi yang tinggi pada pengenalan gambar tetapi memiliki kekurangan pada proses training yang lama dan biaya komputasi yang besar[8]. Oleh karena itu pada penelitian ini memakai Haar Cascade dan CNN untuk pengenalan wajah dan deteksi kantuk yang hanya berfokus pada objek mata terbuka dan tertutup, dan juga untuk mengurangi beban komputasi dan waktu saat training maka dilakukan modifikasi pada saat skema pengujian berupa modifikasi menggunakan dropout. Dropout

juga digunakan untuk mengurangi kegagalan pada saat testing yang dikenal dengan overfitting[9].

Pada penelitian ini bertujuan untuk mengimplementasikan algoritma Machine Learning dalam sistem pengenalan wajah dan deteksi kantuk dengan mendapatkan nilai akurasi yang akurat lebih tinggi dari penelitian sebelumnya dengan menambahkan pengenalan wajah serta minimal akurasi pengenalan wajah dan deteksi kantuk sebesar 97%. Tahapan yang dilakukan untuk mengembangkan sistem pada penelitian ini terdiri dari dua tahap. Pertama, pembuatan model menggunakan *Convolutional Neural Network (CNN)*. Model dibuat dengan melakukan modifikasi menggunakan Dropout, Dropout digunakan untuk mendapatkan waktu training yang lebih cepat pada saat pembuatan model dan mencegah overfitting. *Convolutional Neural Network (CNN)* adalah suatu model pendekatan Deep Learning untuk melakukan pemrosesan gambar menjadi sebuah data klasifikasi yang akan diuji[10]. Kedua, pendeteksian kantuk terdeteksi pada saat mata tertutup atau setengah terbuka karena Haar Cascade melakukan deteksi seluruh wajah terlebih dahulu untuk mendapatkan pendeteksian wajah lalu dilanjutkan deteksi pada area mata menggunakan fitur cascade classifier. Haar Cascade adalah algoritma klasifikasi wajah menggunakan OpenCV untuk pengenalan dan pendeteksian wajah[10]. Sistem pengenalan wajah pada penelitian ini bertitik tumpu pada kondisi mata yang terbuka dan tertutup untuk mendeteksi kantuk pada wajah.

Berdasarkan paparan tersebut, penulis mengajukan sebuah penelitian untuk pengenalan wajah dan deteksi kantuk menggunakan metode Haar Cascade dan Convolutional Neural Network (CNN). Hasil penelitian diharapkan dapat mendapatkan hasil yang sama baiknya dengan penelitian sebelumnya yaitu sebesar 97% dan diharapkan dengan memakai modifikasi arsitektur CNN dengan menggunakan Dropout akan menghasilkan nilai akurasi yang tetap tinggi serta memiliki waktu training yang lebih cepat.

2. TINJAUAN PUSTAKA

Pada penelitian asisten berkendara menggunakan Raspberry Pi dan Haar Cascade Classifier. Pada penelitian tersebut menggunakan metode Haar Cascade Classifier yang diterapkan pada Raspberry Pi dengan MQ3 sensor untuk mendeteksi hal berbau alkohol dan diletakan pada dashboard mobil. Hasil uji coba dilakukan dengan cara melakukan testing dan mendapatkan hasil waktu hanya 2 detik saja untuk menentukan apakah pengemudi terdeteksi

mengantuk atau tidak. Untuk pendeteksian hal berbau alkohol didapatkan waktu 1 detik saja[3].

Pada penelitian model deteksi kantuk pada pengemudi menggunakan Convolutional Neural Network (CNN) untuk aplikasi android. Pada penelitian tersebut metode yang digunakan adalah Facial Landmark Detection untuk mendeteksi wajah dan CNN untuk memproses gambar dataset sebagai model untuk diuji pada Convolutional Neural Network (CNN), hal ini dilakukan untuk mendapatkan hasil training dari dataset berupa nilai akurasi. Dataset memiliki 22 kelas dengan kondisi mengantuk, tertidur, dan berkedip dari orang yang berbeda suku pada situasi pagi dan malam. Hasil uji coba model didapatkan rata-rata nilai akurasi sebesar 83.33% dengan maksimal size model tidak melebihi 75KB[4].

Pada penelitian ulasan deep learning tentang deteksi kantuk pada pengemudi. Pada penelitian tersebut dilakukan pengambilan video yang berpusat pada area muka sebagai dataset penelitian. Metode yang dipilih adalah metode yang mempunyai efisiensi kuat pada penelitian deteksi kantuk pada pengemudi yaitu *support vector machine (SVM)*, *hidden markov model (HMM)*, dan *Convolutional Neural Network (CNN)*. Pengujian dilakukan dengan cara mengambil video dari kamera atau smartphone pengemudi dan diambil sedemikian rupa agar wajah pengemudi terlihat jelas. Pada metode SVM digunakan untuk melakukan deteksi kantuk dari dataset yang telah dikategorikan dan pengujian dilakukan dengan menggunakan Haar feature agar dapat mendeteksi bagian bagian wajah dan mata, selanjutnya SVM melakukan pengujian pada status mata terbuka dan tertutup untuk memicu alarm. Pada metode HMM digunakan untuk deteksi mata berdasarkan warna dan geometris dari mata, tetapi menggunakan metode HMM hanya bisa digunakan pada dalam ruangan dikarenakan deteksi yang harus cukup detail. Hasil uji coba menjelaskan bahwa untuk pendekatan *Deep Learning* terutama *Convolutional Neural Network (CNN)* dapat menyelesaikan masalah klasifikasi seperti masalah computer vision task, scene segmentation, emotion recognition, object detection, dan image classification. Convolutional Neural Network (CNN) memiliki akurasi dan performa yang sangat bagus dibandingkan dengan metode SVM dan HMM, metode ini dapat dipakai juga pada pengukuran kewaspadaan pengemudi, deteksi keaktifan, pengukuran konsentrasi, dan pengukuran perhatian[11].

Pada penelitian *Convolutional Neural Network* sebagai deteksi tulisan aksara jawa. Pada penelitian tersebut metode CNN digunakan untuk melakukan

klasifikasi dan training dataset gambar aksara bahasa jawa. Uji coba dilakukan dengan dataset gambar beresolusi 28x28 px sebanyak 2000 data dengan 20 kelas. Untuk evaluasi, dilakukan training dataset menggunakan model (MLP) dengan 2 *hidden layer*. *Dataset* Hasil uji coba didapatkan konklusi bahwa model CNN lebih baik daripada model MLP, tetapi CNN membutuhkan waktu lebih lama saat training dataset daripada MLP. Hasil akurasi dari metode CNN tidak mencapai 90% disebabkan oleh kurangnya dataset untuk training data[10].

Pada penelitian pengenalan ekspresi wajah menggunakan CNN dan LBP. Pada Penelitian tersebut metode yang digunakan adalah CNN dan LBP. Metode CNN digunakan untuk melakukan klasifikasi yang sudah tertanam pada CNN yaitu Soft Max untuk klasifikasi gambar dan metode LBP dilakukan untuk melakukan ekstraksi fitur dengan ditambahkan klasifikasi dari metode SVM untuk klasifikasi dari gambar yang sudah dilakukan ekstraksi fitur. Dataset yang digunakan pada penelitian diambil melalui JAFFE, CK+, dan YALE FACE untuk ekspresi wajah dan dataset dibagi menjadi dua pembagian yaitu 70% untuk training dan 30% untuk testing. Hasil uji coba setelah melakukan ekstraksi fitur menggunakan algoritma LBP dan SVM menunjukan metode CNN memiliki presisi yang lebih baik pada dataset CK+ adalah 97.32% dan pada dataset YALE FACE adalah 31.82% sebagai dataset yang memiliki akurasi paling rendah[12].

Pada penelitian uji coba deteksi kantuk dan alarm secara *real-time* berdasarkan kondisi mata menggunakan gabungan dari metode Haar Cascade Dan Convolutional Neural Network (CNN). Pada penelitian tersebut metode yang digunakan adalah Haar Cascade untuk deteksi wajah dan Convolutional Neural Network (CNN) untuk mencari hasil rata-rata akurasi model Convolutional Neural Network (CNN) dalam mendeteksi kantuk. Uji coba dilakukan dengan dataset sebanyak 210 gambar, 110 gambar mata tertutup dan 110 gambar mata terbuka beresolusi 24x24 pixel. Hasil uji coba akurasi Haar Cascade dalam mendeteksi wajah adalah 100% dan rata-rata akurasi model Convolutional Neural Network (CNN) dalam mendeteksi kantuk mencapai 97,23% dengan waktu komputasi 0.2975 detik[6].

Pada penelitian sistem pengenalan wajah dengan algoritma Haar Cascade dan Local Binary Pattern Histogram (LBPH). Pada penelitian tersebut metode yang digunakan adalah Haar Cascade untuk mendeteksi wajah dan Local Binary Pattern Histogram (LBPH) untuk pengenalan wajah. Dataset diambil dengan cara memasukan 20 gambar dari kamera dan

disimpan dalam satu folder untuk diekstraksi nilai histogram agar didapatkan data dalam bentuk array. Pengujian dilakukan dengan jarak 50-160cm untuk menentukan apakah penelitian dapat dikatakan berhasil. Hasil uji coba deteksi wajah memiliki efektifitas tergantung dari kecepatan kamera dalam memuat gambar yaitu berkisar 12-13 FPS dalam satu frame[13].

Pada penelitian penerapan metode Convolutional Neural Network pada pengenalan pola citra sandi rumput. Pada penelitian tersebut dilakukan pengenalan pola pada citra sandi rumput dan mencari tingkat akurasi dari CNN dalam mengenali pola citra sandi rumput. Metode CNN digunakan untuk klasifikasi jenis pola citra dari sandi rumput dan CNN akan melakukan training pada dataset sebanyak 260 data. Hasil uji coba menunjukan nilai akurasi sebesar 96,92% dan hasil ini dipengaruhi oleh kerapian dari data dan metode juga berpengaruh besar pada hasil yang didapat[14].

Pada penelitian pemantauan deteksi kantuk menggunakan Convolutional Neural Networks (CNN) metode yang digunakan adalah Convolutional Neural Network untuk mendeteksi kantuk pada pengendara apakah mengantuk atau tidak mengantuk dan Haar Cascade Classifier untuk mendeteksi wajah yang berfokus pada bagian mata. Dataset yang digunakan adalah gambar mata dengan jumlah 1000 data dengan resolusi 24x24 px. Selanjutnya setelah proses ekstraksi gambar mata, dilakukan . Hasil uji coba menggunakan Dataset sebanyak 1000 data dan hasil akurasi yang didapatkan setelah melakukan training sebesar 97% dan testing sebesar 67%[15].

Pada penelitian sistem presensi mahasiswa berdasarkan pengenalan wajah pada mini PC menggunakan metode Local Binary Pattern dan K-Nearest Neighbor. Pada penelitian tersebut digunakan metode LBP untuk pengenalan wajah dan K-nearest Neighbor untuk mencari nilai akurasi dan waktu komputasi. Proses pengujian dilakukan dengan meletakan webcam dengan holder setinggi 75cm dan jarak 8 mahasiswa dengan prototipe sistem berjarak 200cm. Hasil uji coba pendeteksian menggunakan Haar Cascade memiliki nilai rata-rata akurasi sebesar 88.88% dan hasil uji coba pengenalan wajah menggunakan LBP dan KNN memiliki nilai rata-rata akurasi sebesar 78.125% pada $k = 3$ dengan waktu komputasi selama 28 sampai 160 milisekon[16].

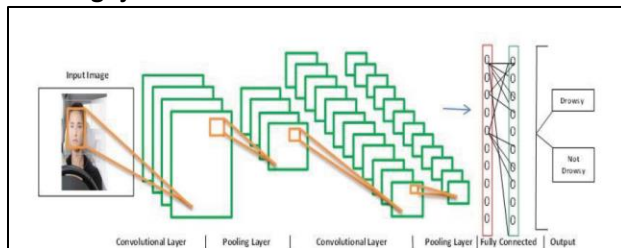
Berdasarkan tinjauan pustaka yang telah dipaparkan, penulis melakukan penelitian dengan pengenalan wajah dan deteksi kantuk menggunakan metode Haar Cascade dan Convolutional Neural

Network (CNN). Metode Haar Cascade digunakan untuk pengenalan wajah dan Convolutional Neural Network (CNN) digunakan untuk mendeteksi kantuk dan melakukan training untuk menentukan hasil akurasi dari pendeteksian kantuk yang berpusat pada mata yang terbuka dan tertutup. Metode Haar Cascade mempunyai rata-rata keberhasilan dalam mendeteksi wajah sebesar 100% berdasarkan objek wajah yang dimiliki sebagai dataset[17]. Bergitupula dengan CNN pada saat mengklasifikasikan citra mengantuk dan tidak mengantuk rata-rata sebesar 97% berdasarkan training model dan testing dengan cara pendeteksian wajah secara langsung menggunakan kamera webcam berdasarkan jarak 30 sampai 50 cm[18]. Penelitian ini merupakan uji coba untuk melihat apakah metode Haar Cascade dan CNN dapat melakukan pengenalan wajah dan deteksi kantuk pada satu objek wajah didalam satu buah citra.

3. METODE PENELITIAN

Pengujian memiliki 2 tahap yaitu pertama, pembuatan arsitektur model CNN untuk mendeteksi kantuk pada wajah berdasarkan mata terbuka dan tertutup. Pengujian kedua dilakukan dengan mendeteksi kantuk pada wajah berjarak 30-50 cm dari kamera, untuk mendapatkan akurasi dilakukan dengan cara menghitung menggunakan rumus $(\text{jumlah data benar} / \text{jumlah data total}) \times 100$ [6].

3.1 Pengujian Pertama



Gambar 3.1 Arsitektur model CNN

Pembuatan arsitektur model CNN terdiri dari convolution layer untuk filter pada citra input dengan filter yang bergeser secara bertahap untuk menghasilkan feature map. Activation ReLU digunakan untuk fungsi aktivasi yang menghasilkan output 0 jika inputnya negative dan mengurangi terjadinya vanishing gradient. Pooling layer digunakan untuk mengurangi dimensi dari feature map[9]. Classification layer berfungsi untuk mengklasifikasikan citra input kedalam kelas yang telah ditentukan dan layer ini menghasilkan probabilitas setiap kelas. Tahap pengujian merupakan tahap mempersiapkan arsitektur model CNN[9].

3.2 Pengujian Kedua

Pengujian kedua merupakan pengujian untuk mendeteksi kantuk pada seseorang secara real-time dan pengujian dilakukan dengan pengetasan sebanyak 20 koresponden untuk dilakukan pengujian dengan berbagai macam kondisi mata seperti mata sayu, mata sipit, dan mata yang memakai kacamata. Hasil akurasi didapatkan dengan cara mendeteksi wajah pada jarak 30-50 cm dan menggunakan rumus $(\text{jumlah data benar} / \text{jumlah data total}) \times 100$ [19].



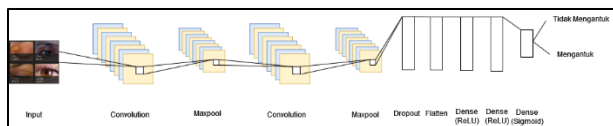
Gambar 3.1 Pengujian Kedua

4. HASIL DAN PEMBAHASAN

Pengujian dilakukan sebanyak 2 pengujian, pengujian pertama merupakan perancangan arsitektur CNN untuk proses yang menghasilkan model yang optimal. Untuk mendapatkan model yang optimal dilakukan training model dengan arsitektur CNN sederhana dengan menambahkan dropout untuk mencegah overfitting. Training model CNN dilakukan sebanyak 3 kali dengan nilai dropout yang berbeda yaitu dengan nilai 0.2, 0.3, dan 0.5 dengan epoch bernilai 100. Setelah mendapatkan model dengan akurasi yang paling baik, maka model akan digunakan untuk pengujian kedua. Pengujian kedua merupakan penerapan model CNN kedalam code menggunakan haar cascade untuk mendapatkan hasil pendeteksian mengantuk dan tidak mengantuk.

Dalam melakukan perancangan arsitektur CNN digunakan dataset dengan 2 kelas kondisi mata yaitu kondisi mata terbuka dan tertutup. Dataset yang akan digunakan mempunyai jumlah data gambar mata yang terbuka sebanyak 1452 gambar dan mata yang tertutup sebanyak 1452 gambar dengan total dataset

sebanyak 2904 gambar. Dataset yang akan digunakan untuk membuat model deteksi kantung dibagi menjadi training set dan testing set, training set memiliki 80% data dan testing set memiliki 20% data.



Gambar 4.1 Arsitektur CNN

Pengujian pertama merupakan perancangan arsitektur CNN untuk proses training yang menghasilkan model yang optimal. Arsitektur CNN pada penelitian ini terbagi dalam 2 tahapan, yaitu feature learning dan classification. Proses tahapan pembuatan arsitektur dijelaskan melalui penjelasan dibawah ini :

1. Proses konvolusi pertama digunakan kernel berukuran 3x3 dengan jumlah filter sebanyak 32 filter. Penggunaan stride juga digunakan untuk menunjukkan pergeseran 1 kernel terhadap matriks input. Setelah melakukan konvolusi, selanjutnya ditambahkan fungsi aktivasi ReLU (Retrified Linear Unit). Fungsi ini bertujuan untuk mengubah nilai negatif pada matriks setelah melakukan konvolusi menjadi nol. Hasil konvolusi pada tahap ini mempunyai ukuran yang sama dengan kernel input dikarenakan menggunakan nilai "padding = same".

2. Proses maxpooling dilakukan setiap setelah konvolusi untuk mengurangi dimensi data dan mempertahankan fitur-fitur penting dalam nilai maksimum 2x2. Penggunaan stride digunakan untuk menunjukkan pergeseran 2 kernel terhadap hasil matriks sebelumnya.

3. Proses konvolusi kedua digunakan kernel berukuran 3x3 dengan jumlah filter sebanyak 32 filter. Penggunaan stride juga dibutuhkan untuk menunjukkan pergeseran 1 kernel terhadap matriks input. Setelah melakukan konvolusi, selanjutnya ditambahkan fungsi aktivasi ReLU (Retrified Linear Unit). Fungsi ini bertujuan untuk mengubah nilai negatif pada matriks setelah melakukan konvolusi menjadi nol. Hasil konvolusi pada tahap ini mempunyai ukuran yang sama dengan kernel input dikarenakan menggunakan nilai "padding = same".

4. Proses Maxpooling dilakukan setiap setelah konvolusi untuk mengurangi dimensi data dan mempertahankan fitur- fitur penting dalam nilai maksimum 2x2. Penggunaan stride digunakan untuk menunjukkan pergeseran 2 kernel terhadap hasil matriks sebelumnya.

5. Proses Dropout dilakukan setelah beberapa layer untuk menghapus bagian neuron secara acak dan tidak akan digunakan dalam proses training. Hal ini bertujuan untuk mengurangi overfitting pada saat proses training.

6. Proses Flatten digunakan untuk menghasilkan output lapisan terakhir menjadi array satu dimensi sesuai dengan output lapisan terakhir dan lapisan flatten menjadi penghubung menuju lapisan dense yang membutuhkan input array satu dimensi.

7. Proses Dense pertama dengan nilai 128 neuron digunakan untuk menghubungkan output dari lapisan konvolusi ke lapisan output terakhir dengan menggunakan ReLU untuk meningkatkan kecepatan konvergensi model.

8. Proses Dense kedua dengan nilai 128 neuron dengan aktifasi ReLU yang sama seperti lapisan dense pertama digunakan untuk memepelajari fitur-fitur kompleks yang telah diekstrak dari lapisan dense sebelumnya.

9. Proses Dense terakhir dengan fungsi sigmoid berfungsi untuk melakukan klasifikasi masalah dengan kelas yang menghasilkan output antara 0 dan 1 atau sesuai dengan jumlah kelas yang diharapkan yaitu mengantuk dan tidak mengantuk.

Kemudian model yang dibentuk menggunakan arsitektur tersebut memiliki jumlah total parameter sebesar 560.578 dengan rincian sebagai berikut,

Tabel 4.1 Parameter CNN

No.	Nama	Size	Parameter
0	Input	32*32*3	0
1	Conv2d_1	32*32*32	896
2	MaxPool_1	16*16*32	0
3	Conv2d_2	16*16*64	18496
4	MaxPool_2	8*8*64	0
5	Dropout	8*8*64	0
6	Flatten	4096	0
7	Dense	128	524416
8	Dense	128	16512
9	Output	2	258
	Total		560.578

Parameter yang dilatih pada model ini berjumlah 560.578 parameter. Tabel 4.1 merupakan model yang terbentuk dari hasil training. Setiap proses konvolusi akan membuat ukuran citra semakin mengecil hingga 8x8 dan 64 fitur. Selanjutnya dilakukan perubahan data pada lapisan flatten menjadi data array satu dimensi agar dapat lanjut menuju lapisan dense dan didapatkan nilai 4096. Pada lapisan dense pertama terdiri dari 128 neuron dengan fungsi aktifasi ReLU yang menciptakan hidden layer dengan output 128 menuju lapisan dense kedua. Lapisan dense

kedua sama sama dengan lapisan dense pertama, selanjutnya pada output mempunyai nilai 2 dikarenakan kelas yang menghasilkan output nilai 0 dan 1 sesuai dengan kelas yang digunakan pada model yaitu mengantuk dan tidak mengantuk

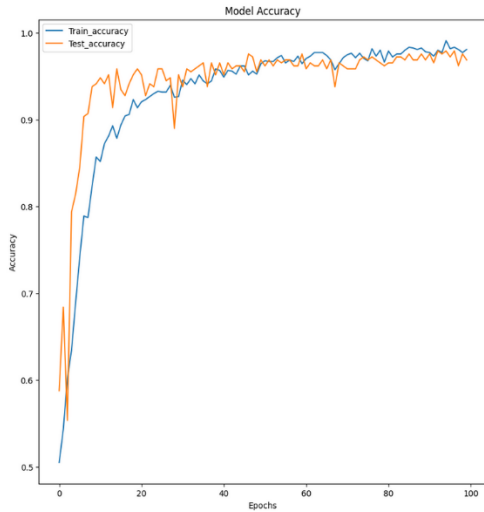
4.1 Perbandingan Hasil Training

Pembangunan model CNN dilakukan dengan cara seperti perancangan model tetapi untuk mendapatkan model yang optimal dan tidak overfitting dibutuhkan pemakaian nilai dropout yang berbeda setiap percobaan. Training model dilakukan sebanyak 3 kali dengan nilai yang berbeda untuk melihat hasil akurasi yang paling baik untuk digunakan pada pengujian kedua.

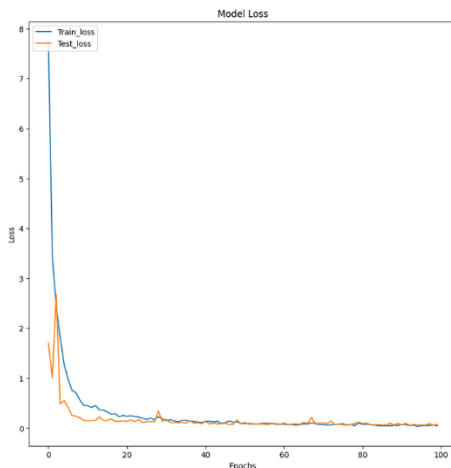
Tabel 4.2 Perbandingan hasil dropout

Nilai Dropout	Accuracy	Val_accuracy
Tanpa Dropout	0,98	0,96
0,2	0,98	0,96
0,4	0,96	0,96
0,5	0,98	0,98

Dengan hasil yang didapatkan, maka peneliti memilih nilai paling baik yaitu nilai dropout 0,2 pada pengujian training model untuk dilanjutkan menjadi model. Setelah itu dilakukan visualisasi nilai loss dan akurasi dari dataset yang digunakan berdasarkan jumlah epoch yang dilakukan.

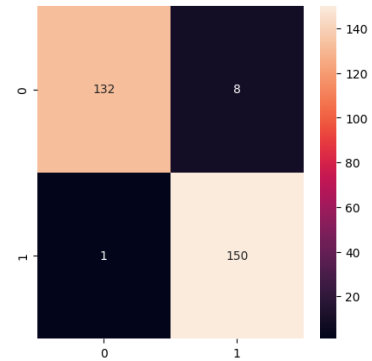


Gambar 4.2 Model akurasi



Gambar 4.3 Model loss

Berdasarkan hasil visualisasi model dataset tersebut tingkat accuracy-rate akhir yang didapat yaitu 98% dan loss-rate 0,5%. Kemudian dapat dilakukan evaluasi model pada dataset dan hasil klasifikasinya menggunakan confusion matrix.



Gambar 4.4 Confusion matrix

Selanjutnya menggunakan confusion matrix dilakukan untuk perhitungan informasi yang lebih detail untuk mengevaluasi model seperti nilai akurasi, precision, recall, dan f1-score.

```

10/10 [=====] - 0s 15ms/step
          precision    recall  f1-score   support

     0       0.99      0.94      0.97       140
     1       0.95      0.99      0.97       151

 accuracy: 0.97
 macro avg: 0.97      0.97      0.97
 weighted avg: 0.97      0.97      0.97
    
```

Gambar 4.5 Model akurasi

4.2 Penerapan Model

Penerapan model dimulai dari deteksi kantuk yang dilakukan secara realtime dengan menggunakan model klasifikasi yang sudah dibuat. Hasil deteksi kantuk dilakukan kepada beberapa orang untuk menguji apakah dengan menambahkan model CNN pada metode haar cascade classifier dapat mendeteksi kantuk secara realtime.. Pengujian dilakukan dengan mendeteksi kantuk pada wajah berjarak 30-50 cm dari kamera dan pada saat wajah kantuk dan tidak mengantuk terdeteksi maka gambar wajah akan tersimpan kedalam folder beserta file face_captured.xlsx yang didalamnya berisi tanggal dan deteksi kondisi wajah yang terambil. Selanjutnya untuk mendapatkan akurasi dilakukan dengan cara menghitung menggunakan rumus (jumlah data benar/jumlah data total) x 100. Berikut merupakan hasil dari deteksi kantuk pada wajah yang dilakukan pada empat koresponden.

Tabel 4.3 Deteksi kantung koresponden pertama

Jumlah Gambar	Jarak (cm)	Deteksi kantung	Akurasi
	30	Mengantuk	80%
		Tidak Mengantuk	70%
10	40	Mengantuk	70%
		Tidak Mengantuk	70%
	50	Mengantuk	70%
		Tidak Mengantuk	90%

Pada tabel 4.3 dapat dilihat deteksi kantung menyatakan wajah tersebut mengantuk disaat mata dari wajah yang dideteksi tersebut tertutup setengah maupun tertutup. Deteksi kantung mendapat akurasi nilai 80% dari jarak 30 cm, 70% dari jarak 40 cm, dan 70% dari jarak 50 cm dari kamera.

Tabel 4.4 Deteksi kantung koresponden kedua

Jumlah Gambar	Jarak (cm)	Deteksi kantung	Akurasi
	30	Mengantuk	80%
		Tidak Mengantuk	70%
10	40	Mengantuk	70%
		Tidak Mengantuk	70%
	50	Mengantuk	70%
		Tidak Mengantuk	90%

Pada tabel 4.4 dapat dilihat deteksi kantung menyatakan wajah tersebut mengantuk disaat mata dari wajah yang dideteksi tersebut tertutup setengah maupun tertutup. Deteksi kantung mendapat akurasi nilai 80% dari jarak 30 cm, 70% dari jarak 40 cm, dan 70% dari jarak 50 cm dari kamera.

Tabel 4.5 Deteksi kantung koresponden ketiga

Jumlah Gambar	Jarak (cm)	Deteksi kantung	Akurasi
	30	Mengantuk	80%
		Tidak Mengantuk	70%
10	40	Mengantuk	80%
		Tidak Mengantuk	70%
	50	Mengantuk	60%
		Tidak Mengantuk	90%

Pada tabel 4.5 dapat dilihat deteksi kantung menyatakan wajah tersebut mengantuk disaat mata dari wajah yang dideteksi tersebut tertutup setengah

maupun tertutup. Deteksi kantung mendapat akurasi nilai 80% dari jarak 30 cm, 80% dari jarak 40 cm, dan 60% dari jarak 50 cm dari kamera.

Tabel 4.6 Deteksi kantung koresponden keempat

Jumlah Gambar	Jarak (cm)	Deteksi kantung	Akurasi
	30	Mengantuk	80%
		Tidak Mengantuk	100%
10	40	Mengantuk	70%
		Tidak Mengantuk	70%
	50	Mengantuk	30%
		Tidak Mengantuk	90%

Pada tabel 4.6 dapat dilihat deteksi kantung menyatakan wajah tersebut mengantuk disaat mata dari wajah yang dideteksi tersebut tertutup setengah maupun tertutup. Deteksi kantung mendapat akurasi nilai 80% dari jarak 30 cm, 80% dari jarak 40 cm, dan 60% dari jarak 50 cm dari kamera.

5. KESIMPULAN DAN SARAN

Hasil ujicoba menggunakan *dropout* mendapatkan nilai yang tinggi yaitu 0,98 akurasi dan tidak ditemukan adanya *overfitting*. Deteksi kantung menggunakan model Convolutional Neural Network (CNN) yang dipilih mampu melakukan deteksi kantung secara realtime dengan akurasi yang cukup bagus berdasarkan kondisi wajah dan ciri wajah yang berbeda yaitu pada pendeteksian koresponden pertama hingga keempat, nilai akurasi untuk pendeteksian kantung paling rendah didapatkan pada koresponden keempat pada jarak 50 cm dengan nilai akurasi 30% dan nilai akurasi paling tinggi didapatkan pada semua koresponden pada jarak 30 cm dengan nilai akurasi 80%..

Hasil akurasi dapat meningkat jika pada saat ujicoba deteksi kantung secara langsung dilakukan pada keadaan yang terang. Hasil akurasi juga dapat meningkat jika pada saat ujicoba deteksi kantung secara langsung ditambahkan interval waktu pada saat pengambilan agar hasil menjadi optimal. Penambahan dataset pada penelitian sangat mungkin dilakukan agar menghasilkan kelas yang bervariasi seperti keadaan menguap. Penggantian dataset juga sangat mungkin dilakukan agar mendapatkan hasil yang lebih optimal, *dataset* yang disarankan yaitu dataset video kondisi mengantuk dan tidak mengantuk.

UCAPAN TERIMA KASIH

Ucapan terima kasih kepada bapak dosen dan bapak penguji yang sudah menyempatkan diri untuk membantu saya dalam membuat jurnal ini dan terimakasih kepada orang terdekat yang sudah membantu mau itu pikiran maupun jasa.

DAFTAR PUSTAKA

- [1] S. Soares, T. Monteiro, A. Lobo, A. Couto, L. Cunha, and S. Ferreira, "Analyzing driver drowsiness: From causes to effects," *Sustain.*, vol. 12, no. 5, pp. 1–12, 2020, doi: 10.3390/su12051971.
- [2] W. H. Sugiharto, M. I. Ghozali, and A. C. Murti, "Pemodelan Alat Pencegah Micro-Sleep Sebagai Upaya Mitigasi Kecelakaan Transportasi," *J-SAKTI (Jurnal Sains Komput. dan Inform.)*, vol. 3, no. 1, p. 11, 2019, doi: 10.30645/j-sakti.v3i1.92.
- [3] V. P. Kumar, P. Aravind, S. N. D. Pooja, S. Prathyush, S. Angeldeborah, and K. R. S. Chandran, "Driver assistance system using raspberry Pi and haar cascade classifiers," *Proc. - 5th Int. Conf. Intell. Comput. Control Syst. ICICCS 2021*, no. Iccics, pp. 1729–1735, 2021, doi: 10.1109/ICICCS51141.2021.9432361.
- [4] R. Jabbar, M. Shinoy, M. Kharbeche, K. Al-Khalifa, M. Krichen, and K. Barkaoui, "Driver Drowsiness Detection Model Using Convolutional Neural Networks Techniques for Android Application," *2020 IEEE Int. Conf. Informatics, IoT, Enabling Technol. ICloT 2020*, pp. 237–242, 2020, doi: 10.1109/ICloT48696.2020.9089484.
- [5] S. Mansoor, G. Sadineni, and S. H. Kausar, "Attendance Management System Using Face Recognition Method," *J. Phys. Conf. Ser.*, vol. 2089, no. 1, 2021, doi: 10.1088/1742-6596/2089/1/012078.
- [6] R. T. Puteri and F. Utaminigrum, "Micro-sleep detection using combination of haar cascade and convolutional neural network," *ACM Int. Conf. Proceeding Ser.*, pp. 130–135, 2020, doi: 10.1145/3427423.3427433.
- [7] M. F. Sitorus, R. Fatharani, N. Fadhillah, T. Informatika, F. Teknik, and U. Samudra, "Sistem Deteksi Multi Wajah Menggunakan Metode Haar Cascade Classifier," vol. 01, no. 01, pp. 1–5, 2020, [Online]. Available: <https://ejournalunsam.id/index.php/jitkom/>
- [8] A. Sada, "Histogram-Based Image Pre-processing for Machine Learning," *2018 IEEE 7th Glob. Conf. Consum. Electron.*, no. Gcce, pp. 272–275, 2018.
- [9] Alidin, "KLASIFIKASI PARU-PARU NORMAL, COVID-19, DAN PNEUMONIA BERDASARKAN CITRA RONTGEN MENGGUNAKAN METODE KLASIFIKASI CNN," no. November, 2022.
- [10] C. K. Dewa, A. L. Fadhilah, and A. Afiahayati, "Convolutional Neural Networks for Handwritten Javanese Character Recognition," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 12, no. 1, p. 83, 2018, doi: 10.22146/ijccs.31144.
- [11] C. C. Ukwuoma and C. Bo, "Deep Learning Review on Drivers Drowsiness Detection," *TIMES-iCON 2019 - 2019 4th Technol. Innov. Manag. Eng. Sci. Int. Conf.*, 2019, doi: 10.1109/TIMES-iCON47539.2019.9024642.
- [12] R. Ravi, S. V. Yadhukrishna, and R. Prithviraj, "A Face Expression Recognition Using CNN LBP," *Proc. 4th Int. Conf. Comput. Methodol. Commun. ICCMC 2020*, no. Iccmc, pp. 684–689, 2020, doi: 10.1109/ICCMC48092.2020.ICCMC-000127.
- [13] S. Al-Aidid and D. Pamungkas, "Sistem Pengenalan Wajah dengan Algoritma Haar Cascade dan Local Binary Pattern Histogram," *J. Rekayasa Elektr.*, vol. 14, no. 1, pp. 62–67, 2018, doi: 10.17529/jre.v14i1.9799.
- [14] A. Hibatullah and I. Maliki, "Penerapan Metode Convolutional Neural Network Pada Pengenalan Pola Citra Sandi Rumput," pp. 1–8, 2019.
- [15] D. R. S. Victoria and D. G. R. Mary, "Driver drowsiness monitoring using convolutional neural networks," *Proc. - IEEE 2021 Int. Conf. Comput. Commun. Intell. Syst. ICCIS 2021*, pp. 1055–1059, 2021, doi: 10.1109/ICCCIS51004.2021.9397070.
- [16] M. A. Prasanty and F. Utaminigrum, "Sistem Presensi Mahasiswa Berdasarkan Pengenalan Wajah Menggunakan Metode LBP dan K-Nearest Neighbor Berbasis Mini PC," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. Vol. 4, no. April 2020, pp. 1168–1171, 2020.
- [17] S. Yulina, "Implementation of Haar Cascade Classifier for Face Detection and Grayscale Image Transformation Using OpenCV," *J. Komput. Terap.*, vol. 7, no. Vol. 7 No. 1 (2021), pp. 100–109, 2021, doi: 10.35143/jkt.v7i1.3411.
- [18] R. T. Puteri and P. Utaminigrum, "Deteksi Kantuk Menggunakan Kombinasi Haar Cascade dan Convolutional Neural Network," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 3, pp. 816–821, 2020.
- [19] I. G. T. Suryawan and I. P. A. E. Darma Udayana, "Optimasi Convolutional Neural Network Untuk Deteksi Covid-19 pada X-ray Thorax Berbasis Dropout," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 3, p. 551, 2022, doi: 10.25126/jtiik.2022935143.