

Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON

Real Time Sleepiness Detection Using OPENCV Library and PYTHON DLIB

Afrizal Zein

Program Studi Teknik Informatika, STMIK Eresha
Jl. Raya Puspitek Serpong No. 10 Tangerang Selatan Banten
e-mail : zeinafrizal@gmail.com

Abstrak--- *Tahukah Anda bahwa di Indonesia kecelakaan yang dipicu mengantuk porsinya melonjak drastis? Data Korlantas Polri menyebutkan, setidaknya setiap hari ada enam kasus kecelakaan akibat mengantuk. Artinya setiap empat jam terjadi satu kecelakaan akibat mengantuk. Banyak diantara kita yang merasakan bahwa rasa kantuk amat mengganggu konsentrasi. Fokus sang pengendara jadi buyar. Boro-boro mau mengantisipasi situasi mendadak, mau merespons yang biasa-biasa saja bisa ngawur.*

Ada sejumlah aspek yang bisa menyebabkan rasa kantuk terus mendera tubuh. Aspek utama tentu saja lantaran kurang tidur. Tubuh manusia normal membutuhkan tidur berkisar 6-8 jam setiap hari. Jika kurang dari itu dapat dipastikan rasa kantuk bakal hadir.

Aspek lain yang bikin mengantuk adalah obat-obatan. Pengendara yang mengonsumsi obat-obatan, seperti obat flu, sesaat sebelum mengemudi akan didera rasa kantuk. Setahu saya, satu jam setelah minum obat flu, rasa kantuk bakal hadir.

Dari masalah diatas penulis mengadakan penelitian sistem komputer vision yang secara otomatis dapat mendeteksi kantuk pengemudi dalam pantauan kamera web real-time dan kemudian membunyikan alarm serta menghubungi keluarga jika pengemudi kelihatannya mengantuk.. Dalam penelitian ini, pengujian deteksi kantuk maupun pemantauan kelopak mata dilakukan berdasarkan beberapa faktor yang memungkinkan dapat mempengaruhi akurasi dari pendeteksian kantuk maupun.. Faktor tersebut berupa pengaruh umur, gaya wajah, penambahan aksesoris dan pelatihan data training. Dari hasil pengujian menunjukkan perolehan tingkat akurasi pendeteksian kantuk mencapai sebesar diatas 90%.

Kata Kunci : *Deteksi Kantuk, OpenCV, Dlib*

Abstract--- *Did you know that in Indonesia accidents triggered drowsy portion jumped dramatically? Police Korlantas data says, at least every day there are six cases of accidents due to drowsiness. This means every four hours there is an accident due to drowsiness. Many of us feel that drowsiness interferes with concentration. The focus of the rider is dispersed. Boro-boro want to anticipate the sudden situation, would respond to the mediocre could be inconsequential.*

There are a number of aspects that can cause drowsiness continues to whack the body. The main aspect is of course because of lack of sleep. Normal human body requires sleep ranges from 6-8 hours every day. If less than that can be sure of drowsiness will be present. Another aspect that makes sleepiness is medication. Drivers who take drugs, such as cold medicine, just before driving will be drowsy. To my knowledge, an hour after taking the flu medication, drowsiness would be present. From the above problems the authors conducted a study of computer vision system that automatically can detect the drowsiness of the driver in real-time web camera monitoring and then sound the alarm and contact the family if the driver seems sleepy. In this research, drowsiness testing and eyelid monitoring performed based on some possible factors can affect the accuracy of the drowsiness detection as well as. The factor is the influence of age, facial style, the addition of accessories and training training data. From the test results showed the acquisition accuracy level of drowsiness reached above 90%.

Keywords : *Drowsiness detection, OpenCV, Dlib*

1. PENDAHULUAN

Ekspresi wajah adalah cara alami untuk menyampaikan sosial informasi tanpa kata-kata di antara manusia. Facial Pengakuan Ekspresi (FER) adalah area penelitian aktif di studi psikologi serta interaksi komputer-manusia. Sebuah ekspresi wajah

dapat dianggap sebagai pengesahan yang terlihat keadaan batin pikiran dan karenanya memberikan gagasan tentang niat, minat, dan psikologi orang itu. Sebagai hasil dari yang besar informasi yang dibawa oleh ekspresi wajah, mereka memainkan peran

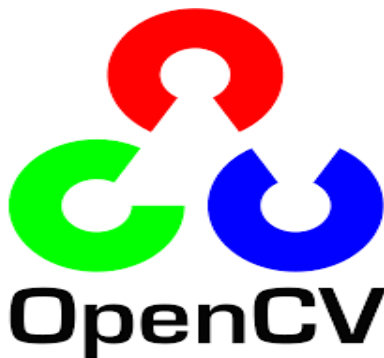
penting dalam interaksi manusia dengan manusia serta mesin [1].

Pengenalan ekspresi kantung dapat digunakan sebagai komponen vital dalam antarmuka alami antara manusia dan mesin. Perangkat pintar seperti webcam dapat mengenali ekspresi seseorang apakah dia dalam kondisi mengantuk. Ada dua pendekatan utama untuk analisis ekspresi kantung. Pertama pendekatan memanfaatkan seluruh citra wajah frontal dan memprosesnya agar berakhir dengan klasifikasi wajah ekspresi. Dalam pendekatan kedua, alih-alih menggunakan wajah gambar secara keseluruhan, itu dibagi menjadi sub-bagian dan diproses lebih lanjut. Manusia secara alami dan intuitif digunakan ekspresi kantung sebagai modalitas yang penting dan kuat untuk mengomunikasikan kondisi tubuh mereka dan berinteraksi secara sosial [2]. Saya telah melanjutkan minat penelitian dalam mengaktifkan komputer sistem untuk mengenali ekspresi kantung dan menggunakan citranya sebagai pemicu diaktifkannya alarm yang terpasang didalam kendaraan[3].

Dalam beberapa tahun terakhir, karena berbagai aplikasinya, deteksi kantung telah menjadi bidang penelitian yang penting dan masih ada upaya diambil untuk meningkatkan akurasi deteksinya. Perbedaannya pada nilai toleransi yang diberikan membedakan ekspresi kantung untuk memicu alarm.

2. TINJAUAN PUSTAKA

- a. **OpenCV** (Open Source Computer Vision Library) adalah salah satu *software* pustaka yang ditujukan untuk pengolahan citra dinamis secara real-time, yang dibuat oleh Intel, dan sekarang didukung oleh Willow Garage dan Itseez. OpenCV dirilis dibawah lisensi permisif BSD yang lebih bebas dari pada GPL, dan memberikan kebebasan sepenuhnya untuk dimanfaatkan secara komersil tanpa perlu mengungkapkan kode sumbernya. Ia juga memiliki antar muka yang mendukung bahasa pemrograman C++, C, Python dan Java, termasuk untuk sistem operasi Windows, Linux, Mac OS, iOS dan Android. OpenCV didisain untuk efisiensi dalam komputasi dan difokuskan pada aplikasi real-time.



Gambar 1. Logo OpenCV

Contoh penerapan OpenCV dengan Python adalah camera yang dipasang di parkir yang mampu membaca plat nomor. Plat nomor ini dikonversi dari analog ke digital lalu diolah menjadi karakter sehingga menjadi data yang bisa dijadikan sebagai informasi penting.

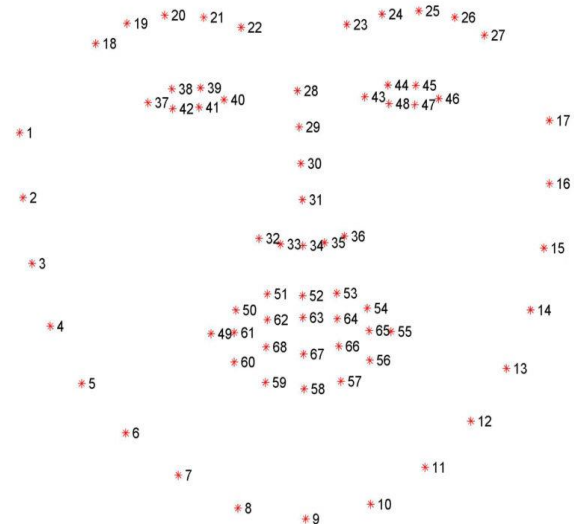
Intinya, OpenCV bersama Python dimanfaatkan untuk mengolah image atau video (tumpukan frame/image) sesuai dengan tujuan masing-masing yang melibatkan kamera untuk menangkap gambar lalu diolah di komputer.

Sebenarnya, OpenCV bisa digunakan di bahasa C++ dan Java, selain Python, tetapi bahasa yang paling mudah dari ketiga ini adalah Python karena sederhana. Saya sudah mempelajari C++ untuk OpenCV, ternyata cukup rumit kode-kodenya. Maka, Anda beruntung mempelajari Python dan juga beruntung mendapatkan modul ini.

b. Dlib

Detektor landmark wajah yang sudah dilatih sebelumnya di dalam perpustakaan dlib digunakan untuk memperkirakan lokasi 68 (x, y) -kolordinat yang memetakan struktur wajah di wajah.

Indeks dari 68 koordinat dapat divisualisasikan pada gambar di bawah ini:



Gambar 2. Memvisualisasikan 68 koordinat landmark wajah dari dataset IBUG 300-W (resolusi lebih tinggi)

Notasi ini adalah bagian dari 68 titik iBUG 300-W dataset yang prediktor landmark wajah dlib dilatih.

Penting untuk dicatat bahwa detektor pendeteksi wajah lainnya ada, termasuk model 194 titik yang dapat dilatih pada dataset HELEN.

Terlepas dari kumpulan data yang digunakan, kerangka kerja dlib yang sama dapat dimanfaatkan untuk melatih prediktor bentuk pada data pelatihan

masukannya - ini berguna jika Anda ingin melatih detektor landmark wajah atau prediktor bentuk khusus milik anda sendiri.

Detektor tengara wajah dlib akan mengembalikan objek bentuk yang mengandung 68 (x, y) -koordinat dari daerah landmark wajah.

Menggunakan fungsi `shape_to_np`, kami mengubah objek ini menjadi array NumPy, memungkinkannya untuk "bermain lebih baik" dengan kode Python kami.

Dengan adanya dua fungsi pembantu ini, kami sekarang siap mendeteksi penanda wajah di gambar.



Gambar 3. Dlib OpenCV Landmark Detection

3. METODE PENELITIAN

Suatu cara dilakukan dengan metode adalah dengan menggunakannya untuk menentukan berapa lama mata seseorang yang bersangkutan telah ditutup. Jika ada mata yang telah ditutup untuk jangka waktu tertentu, kami akan berasumsi bahwa mereka mulai tertidur dan memutar alarm untuk membangunkan mereka dan menarik perhatian mereka.

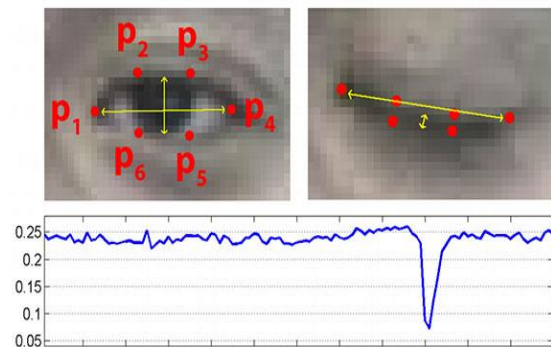
Untuk itu penulis membagi menjadi tiga tahap ;

- Penulis akan menunjukkan cara mengatur kamera di mobil sehingga penulis dapat dengan mudah mendeteksi wajah pengemudi dan menerapkan pelokalan wajah untuk memantau mata saya.
- Penulis kemudian akan menunjukkan bagaimana penulis dapat menerapkan detektor ngantuk kami sendiri menggunakan OpenCV, dlib, dan Python.
- Penulis meminta pengemudi masuk mobil dan berkendara seolah olah pengemudi mengantuk.

Akhirnya seperti yang telah diduga deteksi kantuk bekerja dengan baik dan dapat diandalkan setiap kali pengemudi mulai mengantuk/tertidur maka alarm akan berbunyi.

Algoritma Deteksi Kantuk

Setelah kita mendapatkan daerah mata, kita dapat menghitung aspek rasio mata untuk menentukan apakah mata tertutup atau tidak.



Gambar 4. Kiri-atas: Visualisasi landmark mata saat mata terbuka. Kanan atas: Tanda mata saat mata ditutup. Bawah: Memplot aspek rasio mata dari waktu ke waktu. Penurunan rasio aspek mata menunjukkan kapan

4. HASIL DAN PEMBAHASAN

Pengujian dilakukan menggunakan webcam m-tech quickcam ATW-800 dengan resolusi 5 megapixel yang dihubungkan dengan notebook Axioo Intel Core I7 RAM 8 menggunakan OS Linux.

Kita akan memerlukan paket SciPy sehingga kita dapat menghitung jarak Euclidean antara poin landmark wajah dalam perhitungan rasio aspek mata (bukan persyaratan ketat, tetapi Anda harus memiliki SciPy yang dipasang jika Anda berniat melakukan pekerjaan apa pun dalam visi komputer, pemrosesan gambar, atau ruang pembelajaran mesin).

Kita juga memerlukan paket imutils, rangkaian visi komputer dan fungsi pemrosesan gambar agar bekerja dengan OpenCV lebih mudah. Jika Anda belum menginstal imutils di sistem Anda, Anda dapat memasang / memutakhirkan imutils melalui:

```
$ pip install --upgrade imutils
```

Kita juga akan mengimpor kelas Thread sehingga kita dapat memutar alarm kita di utas terpisah dari utas utama untuk memastikan skrip kita tidak akan menjeda eksekusi saat alarm berbunyi.

Untuk benar-benar memainkan alarm WAV / MP3 kami, kami membutuhkan perpustakaan playsound, di Python, implementasi lintas platform untuk memainkan suara sederhana.

Perpustakaan playsound mudah diinstal melalui pip:

```
$ pip install playsound
```

Untuk mendeteksi dan melokalkan landmark wajah, kami memerlukan pustaka dlib yang diimpor.

Selanjutnya, kita perlu mendefinisikan fungsi `sound_alarm` kami yang menerima jalur ke file audio yang berada di disk dan kemudian memutar file:

```
def sound_alarm(path):
    # jalankan alarm sound
    playsound.playsound(path)
```

Kita juga perlu mendefinisikan fungsi `eye_aspect_ratio` yang digunakan untuk menghitung rasio jarak antara landmark mata vertikal dan jarak antara landmark mata horizontal:

```
def eye_aspect_ratio(eye):
    # menghitung jarak euclidean antara
    dua set
    # landmark mata vertikal (x, y) -
    kolordinat
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])

    # hitung jarak euclidean antara
    horisontal
    # tengara mata (x, y) -kolordinat
    C = dist.euclidean(eye[0], eye[3])

    # menghitung rasio aspek mata
    ear = (A + B) / (2.0 * C)

    # mengembalikan rasio aspek mata
```

Nilai kembalinya aspek rasio mata akan mendekati konstan ketika mata terbuka. Nilai tersebut kemudian akan berkurang dengan cepat menuju nol selama kedipan.

Jika mata tertutup, aspek rasio mata akan tetap hampir konstan, tetapi akan jauh lebih kecil daripada rasio ketika mata terbuka.

Seperti yang bisa kita lihat, aspek rasio mata adalah konstan (menunjukkan mata terbuka), kemudian dengan cepat turun menjadi nol, kemudian meningkat lagi, menunjukkan kedipan telah terjadi. Dalam kasus detektor kantuk kami, kami akan memantau aspek rasio mata untuk melihat apakah nilainya turun tetapi tidak meningkat lagi, sehingga menyiratkan bahwa orang tersebut telah menutup mata mereka.

Langkah selanjutnya adalah menerapkan deteksi landmark wajah untuk melokalisasi masing-masing daerah penting pada wajah:

Blok kode terakhir dalam detektor ngantuk kami menangani menampilkan bingkai output ke layar kami:

Menguji detektor kantuk OpenCV



Gambar 5. Kondisi pengendara saat normal tidak mengantuk



Gambar 6. Kondisi pengendara saat mulai mengantuk, peringatan dini timbul dan alarm berbunyi

5. SIMPULAN

- Detektor rasa kantuk kami bergantung pada dua teknik penglihatan komputer:
- Deteksi tengara wajah, aspek rasio mata, prediksi tengara wajah adalah proses pelokalan struktur wajah dan menjadi kunci pada wajah, termasuk mata, alis, hidung, mulut, dan garis rahang.
- Khususnya, dalam konteks deteksi kantuk, kami hanya membutuhkan daerah sekitar mata.
- Kita dapat menerapkan aspek rasio mata untuk menentukan apakah mata tertutup. Jika mata telah ditutup untuk jangka waktu yang cukup lama, kita dapat berasumsi bahwa pengguna berisiko tertidur dan membunyikan alarm untuk menarik perhatian mereka.

DAFTAR PUSTAKA

- [1] Ni Wayan Marti (2010), *Pemanfaatan GUI Dalam Pengembangan Perangkat Lunak Pengenalan Citra Wajah Manusia Menggunakan Metode Eigenfaces*, ISSN: 1907-5022, SNATI Juni 2010.
- [2] Dodit Suprianto, (2013), *Sistem Pengenalan Wajah Secara Real-Time dengan Adaboost, Eigenface PCA & MySQL* Jurnal EECCIS Vol. 7, No. 2, Desember 2013.
- [3] Shizhan Zhu, (2015), *Face Alignment by Coarse-to-Fine Shape Searching*, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, 2015.
- [4] Afrizal Zein (2016), *Pendeteksian Multi Wajah dan Recognition Secara Real Time Menggunakan Metoda Principal Component Analysis (PCA) dan Eigenface*, Jurnal ESIT STMIK ERESHA, 2016.
- [5] Afrizal Zein (2018), *Menggabungkan Dua Wajah Dengan Metoda Ensemble Regression Trees Menggunakan Pustaka Dlib Dan Opencv Python*, Jurnal ESIT STMIK ERESHA, 2018.
- [6] Davis E. King (2009), *Dlib-ml: A Machine Learning Toolkit*. Journal of Machine Learning Research 10 (2009) 1755-1758.
- [7] Vanita Jain (2017), *Emotion Detection from Facial Expression using Support Vector Machine*. *International Journal of Computer Applications (0975 – 8887)*, Volume 167, No. 8, June 2017.