

Deteksi Masker Menggunakan Algoritma *Support Vector Machine*

R. Ferdian Dita Nugraha^{#1}, Krisna Bayu Dharma Putra^{*2}, Eliezer Delpiero B.P.^{#3}, Muhammad Mahdi^{#4}, Wella Wineke Prachinta^{#5}, M. Farhan Dwi Rizqi^{#6}, Tubagus Naufal Fathurahman^{#7}

[#]Departemen Ilmu dan Elektronika, Universitas Gadjah Mada
Yogyakarta, Indonesia

¹r.ferdian.dita.nugraha@mail.ugm.ac.id

²krisnabayu01@mail.ugm.ac.id

³eliezerdelpiero@mail.ugm.ac.id

⁴muhammad.mahdi@mail.ugm.ac.id

⁵wella.wineke.prachinta@mail.ugm.ac.id

⁶mfarhandr@mail.ugm.ac.id

⁷tubagus.naufal.fathurahman@mail.ugm.ac.id

Abstrak— Jurnal ini membahas tentang deteksi masker menggunakan algoritma *Support Vector Machine*. Metode yang dilalui diantaranya akuisisi citra, pre-processing, ekstraksi fitur, normalisasi, pelatihan model, dan pengujian model. Hasil deteksi menunjukkan nilai akurasi sebesar 90% saat testing menggunakan dataset tersedia. Secara keseluruhan, prediksi yang menggunakan data real world berhasil melakukan pendeteksian, tetapi terdapat beberapa kekurangan yaitu ketika citra wajah menggunakan aksesoris atau posisi masker yang tidak tepat.

Kata kunci— masker, deteksi wajah, pengolahan citra digital, *support vector machine*, *machine learning*

I. PENDAHULUAN

Penggunaan masker menjadi sangat populer di era pandemi ini karena salah satu metode efektif yang dapat digunakan untuk menghambat penyebaran dari virus COVID-19 adalah dengan menggunakan masker. Hal ini dibuktikan dengan meningkatnya kasus penularan COVID-19 akibat masyarakat yang tidak mematuhi aturan 3M terutama untuk menggunakan masker. Tentu hal semacam ini dapat merugikan masyarakat lain. Oleh karena itu, perlu dilakukan pengecekan secara berkala kepada masyarakat perihal penggunaan masker.

Namun, fakta di lapangan membuktikan bahwa pengecekan yang dilakukan secara manual menuai banyak masalah dan kesulitan. Salah satunya adalah banyaknya masyarakat yang harus dicek satu persatu oleh petugas. Oleh karena itu, kami menawarkan sebuah solusi pengecekan otomatis bernama Face mask detection.

Face mask detection adalah salah satu permasalahan yang dapat diselesaikan dengan digital image processing. Pada permasalahan ini, kita perlu melakukan pemrosesan citra pada gambar, untuk mendapatkan fitur-fitur pada gambar wajah untuk kemudian melakukan pengecekan ada atau tidaknya

masker yang digunakan dengan baik dan benar pada tiap-tiap wajah.

Pada solusi ini, kami menitik beratkan pada ekstraksi fitur. Beberapa fitur yang kami ambil dari gambar adalah Gray Level Co-occurrence Matrix (GLCM), Hu Moments, dan histogram. Ketiga fitur tersebut dirasa dapat merepresentasikan gambar dengan baik, baik secara warna, ukuran, dan tekstur.

Kami menggunakan pendekatan *machine learning* untuk melakukan prediksi dengan menggunakan algoritma Support Vector Machine. Dengan adanya proses ekstraksi fitur yang mengubah matriks citra menjadi vektor-vektor fitur, pendekatan *machine learning* konvensional dapat diaplikasikan tanpa menggunakan *deep learning*.

Walau aplikasi ini bukanlah hal baru, besar harapan kami untuk menemukan pengetahuan baru selama pembuatan proyek ini, baik di bidang pengolahan citra digital secara praktis, maupun di sisi *machine learning*.

II. DATASET

Digunakan dataset berupa citra grayscale wajah menggunakan masker dan tidak menggunakan masker dan dibagi menjadi dua kelas berdasarkan keberadaan masker pada citra. Dataset yang dipakai memiliki wajah yang menghadap ke beberapa arah tidak hanya lurus ke depan tetapi juga sedikit miring baik ke kanan dan kiri untuk meningkatkan kemampuan model mendeteksi wajah yang tidak hanya lurus kedepan. Dataset juga menggunakan data yang di-augmentasi untuk memperbanyak jumlah dataset yang ada. dataset berisikan 1364 data dengan 685 data bermasker dan 679 data yang tidak memakai masker. lalu dataset dibagi menjadi data untuk training model dan data untuk testing model dengan pembagian 68 data testing memakai masker dan 617 data training memakai masker. dan 68 data testing tanpa masker dan 611 data training tanpa menggunakan masker.

Pembagian data menjadi training dan testing dipilih secara random tetapi tetap memperhatikan keseimbangan jumlah data masker dan non-masker pada data training dan testing nya. dihasilkan pembagian dengan rasio sekitar 1:9 dengan 1 adalah data testing dan 9 merupakan data training. Dengan ini diharapkan model dapat terlatih sama baiknya untuk mengidentifikasi citra digital wajah dengan menggunakan masker maupun tidak dengan memberikan dataset yang tidak berat hanya ke satu kelas tertentu saja.



Fig. 1. Sample dataset citra menggunakan masker maupun tidak

III. MODUL YANG DIGUNAKAN

Modul-modul atau library dalam bahasa pemrograman Python yang digunakan adalah sebagai berikut:

- Numpy: Modul yang digunakan untuk melakukan fungsi-fungsi matematis serta dapat melibatkan array dan matriks multidimensi
- OS: Modul yang digunakan dalam memanggil fungsionalitas yang bergantung pada sistem operasi.
- OpenCV: Modul yang memiliki fungsionalitas untuk mengolah citra dalam computer vision.
- Mahotas: Modul yang juga memiliki fungsionalitas pengolahan citra.
- SKlearn: Modul yang digunakan untuk melakukan data processing dan machine learning.
- PIL: Modul yang digunakan untuk berinteraksi dan memanipulasi citra.

- Pickle: Modul yang digunakan untuk membuat serialisasi maupun deserialisasi struktur objek pada Python
- Math: Modul yang digunakan untuk operasi matematika lanjut.
- SKimage: Modul yang digunakan untuk preprocessing citra.

IV. METODE

A. Akuisisi Citra

Tahap akuisisi citra dilakukan dengan melakukan scanning terhadap direktori file yang berada di dalam direktori folder sistem kita. Direktori folder ini berisi kumpulan citra set bawaan dan citra *real world*. Citra set bawaan adalah citra yang sudah disiapkan dalam paper rujukan kami dan berfungsi untuk melakukan *testing* keakuratan prediksi. Sedangkan citra *real world* adalah set citra yang dibuat sendiri untuk melakukan pengetesan. Sebelum diolah, citra akan diubah ke bentuk *grayscale*.

B. Pre-processing

- Peningkatan Kualitas Citra

Peningkatan kualitas citra dilakukan dengan menggunakan metode *contrast stretching*. Metode *contrast stretching* adalah metode peningkatan citra dimana pixel citra yang awalnya rapat akan dibuat lebih merata.

- *Smoothing*

Smoothing dilakukan dengan menggunakan kernel

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- *Filtering*

Filtering yang dilakukan adalah dengan menggunakan metode *median filtering*. Filter jenis ini bermanfaat untuk mengurangi kemungkinan *noise* dalam citra.

C. Ekstraksi Fitur

- GLCM

Ekstraksi fitur ini bermanfaat untuk mendapatkan relasi antar piksel-piksel dalam citra. Hasil ekstraksi yang didapatkan adalah perhitungan statistika dari penghitungan matriks.

- *Hu moments*

Ekstraksi ini bermanfaat untuk mendapatkan ekstraksi fitur berupa 7 buah hasil perhitungan matematis dari *central moments*. *Hu moments* ini

memiliki sifat *scale invariant*, *rotation invariant*, dan *translation invariant* pada 6 *moments* pertama, sedangkan *moments* ke-7 berubah tergantung refleksi citra[1].

- Histogram

Ekstraksi ini akan menghasilkan intensitas piksel dari citra grayscale dari rentang 0 sampai 255 pixel.

D. Normalisasi

Normalisasi dilakukan dengan mengubah nilai piksel yang memiliki nilai lebih dari 255 menjadi piksel dengan nilai piksel sebesar 255.

Proses lainnya adalah dengan mengubah rentang nilai fitur yang diekstrak menjadi di antara 0 dan 1 menggunakan normalisasi MinMax. Metode normalisasi ini membutuhkan proses yang analog dengan training model, sehingga *scaler* perlu dilatih pada data training untuk menentukan nilai maksimum dan minimum untuk menentukan nilai akhir normalisasi. Proses ini bertujuan supaya konvergensi tercapai dengan lebih cepat.

E. Pelatihan Model

Pelatihan model dilakukan dengan menggunakan *machine learning* dan melakukan pengujian terhadap dataset yang telah disiapkan sebelumnya. Hasil tes ini adalah besaran prediksi dalam rentang 0 sampai 1.

F. Pengujian Model

Pengujian model dilakukan setelah mendapatkan nilai prediksi yang baik. Setelah itu, akan dilakukan pengujian model terhadap dataset *real world*.

V. IMPLEMENTASI DAN HASIL

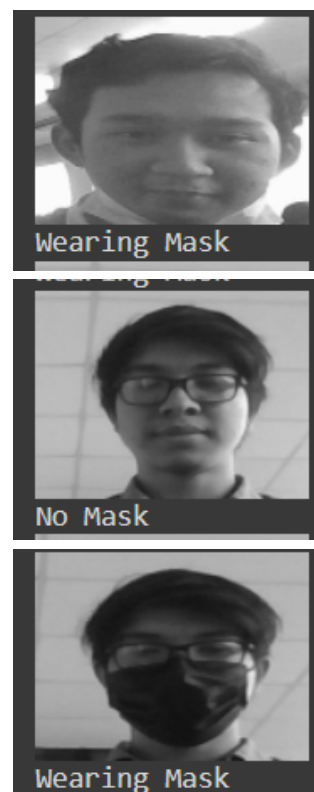
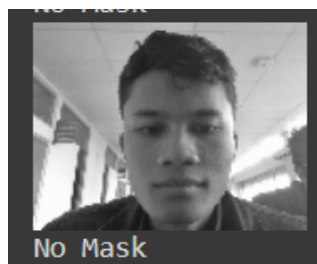


Fig. 2. Hasil prediksi menggunakan model

Implementasi dilakukan menggunakan *platform* Google Colaboratory. Untuk menyimpan dataset gambar, kita menggunakan Google Drive.

Terjadi penurunan akurasi pada model saat diuji menggunakan *real-world data*. Akurasi model saat dites menggunakan dataset bawaan adalah 90%, namun beberapa kali terjadi salah prediksi pada data *real world*.

Kami mendapati bahwa proses normalisasi sangat mempengaruhi hasil akhir prediksi. Untuk meningkatkan akurasi, diaplikasikan normalisasi fitur yang rentangnya disamakan dengan dataset pelatihan. Dengan menggunakan normalisasi MinMax, kita akan mendapatkan data fitur dengan rentang angka dari 0 hingga 1. Hal ini untuk mempermudah konvergensi. Implementasi dilakukan dengan algoritma Support Vector Machine pada library Sklearn.

Pada implementasi saat ini, input berupa gambar bukan video. Gambar tersebut kami ambil secara manual menggunakan *webcam* lalu kami simpan dan kami muat dari direktori sistem.

Implementasi lebih lanjut dapat dilakukan dengan memasang kamera pada program. Hal ini bertujuan supaya model dapat melakukan prediksi secara *realtime*. Namun beberapa kendala mungkin akan dijumpai, seperti kualitas gambar yang jauh berbeda dengan data pelatihan. Hal ini membuat kita perlu melakukan pemrosesan lebih lanjut untuk membuat gambar dari *live feed* semirip mungkin dengan gambar pelatihan. Program ini juga sangat bergantung pada pencahayaan dan posisi wajah terhadap kamera. Kami

mendapati bahwa posisi wajah yang relatif miring akan membuat model salah memprediksi.

VI. KESIMPULAN

Penggunaan masker merupakan salah satu cara dalam menghambat penyebaran COVID-19 yang paling efektif. Namun pengecekan penggunaan masker ini akan sulit dilakukan secara manual, oleh karena itu kami membuat sebuah program dalam pengecekan penggunaan masker secara otomatis agar pengecekan ini lebih efektif dan tidak memakan waktu yang banyak. Untuk mendapat suatu program yang akurat kita akan menggunakan pendekatan *machine learning*.

Dengan pendekatan ini program akan dilatih dengan dataset yang berupa citra grayscale wajah menggunakan masker dan tidak menggunakan masker. Lalu modul yang akan digunakan pada pembuatan program ini bermacam macam seperti Numpy, OS, OpenCV, Mahotas, SKlearn, dan PIL. Pada program yang kita buat, kita akan menerapkan beberapa metode mulai dari preprocessing yang bertujuan untuk meningkatkan kualitas citra yang berisi smoothing dan median filtering. Lalu ada metode ekstraksi fitur yaitu GLCM, Hu moments, dan Histogram. Setelah penerapan metode tersebut kita melakukan normalisasi dan dilanjutkan dengan pelatihan model pada program yang kami buat. Dan terakhir kami lakukan pengetesan model untuk mendapat nilai prediksi.

Dari implementasi program yang kami buat menggunakan metode metode di atas kami mendapat nilai akurasi sekitar 90%, namun masih ada beberapa prediksi yang salah. Jadi dari program dan pengetesan yang kami lakukan dapat disimpulkan bahwa program yang kami buat berhasil dalam mendeteksi penggunaan masker, namun tidak memiliki nilai akurasi yang sempurna atau 100%. Untuk mendapat hasil sempurna dapat dilakukan dengan memodifikasi metode normalisasi sehingga normalisasi MinMax akan mendapatkan data fitur dengan rentang angka dari 0 hingga. Jadi tujuan utama dari pembuatan program kami telah tercapai yaitu memudahkan dalam pengecekan penggunaan masker pada manusia, walaupun nilai keakuratannya belum 100%.

REFERENSI

- [1] S. Mallick, "Shape Matching using Hu Moments (C++/Python)," Shape Matching using Hu Moments (C++/Python), Dec. 10, 2018. <https://learnopencv.com/shape-matching-using-hu-moments-c-python/>