

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Terdapat beberapa penelitian terkait mengenai implementasi pada *face recognition*, yaitu:

2.1.1. *Masked Face Recognition Dataset and Application* [6]

Masalah utama pada studi ini adalah face recognition yang ada pada saat ini bekerja dengan tidak akurat, dikarenakan pada masa pandemi semua orang menggunakan masker. Hal ini disebabkan karena face recognition yang ada mengidentifikasi wajah seseorang dengan cara melihat komponen pada wajah seperti mata, hidung, dan mulut. Tujuan dari penelitian ini adalah untuk meningkatkan performa face recognition yang lebih baik sehingga dapat mengenali wajah pengguna yang menggunakan masker. Peningkatan ini membuat face recognition menjadi mask recognition. Metode yang digunakan pada penelitian ini adalah dengan mendesain face recognition yang ada berdasarkan algoritma deep learning. Algoritma deep learning akan diberikan dataset berupa Masked Face Detection Dataset (MFDD), Real-world Masked Face Recognition Dataset (RMFRD), dan Simulated Masked Face Recognition Dataset (SMFRD). Cara validasinya adalah dengan memberikan dataset kembali berupa hasil buatan peneliti kemudian akan dibandingkan antara jumlah dataset dengan jumlah error yang ada. Hasil dari penelitian ini memberikan akurasi sebesar 50%-95%. Berdasarkan pemahaman jurnal tersebut yang penulis dapatkan adalah pengembangan algoritma *deep learning* dapat digunakan untuk membangun *software* pengenalan wajah untuk mendeteksi wajah pengguna ketika menggunakan masker.

2.1.2. *Face Recognition with Occlusions in the Training and Testing Sets* [7]

Masalah utama pada studi ini adalah pada saat ini banyak algoritma yang tidak mampu mengenali wajah pengguna dikarenakan wajah pengguna yang tertutup sebagian. Hal ini disebabkan karena pada saat ini belum ditemukan metode

yang tepat untuk mencocokkan *dataset* yang ada dengan *dataset* dengan wajah yang tertutup sebagian. Tujuan dari penelitian ini untuk mencari teknik yang tepat untuk mengatasi masalah wajah yang tertutup sebagian. Penggunaan teknik yang tepat juga meningkatkan *face recognition* dalam hal pemfokusan dan pencocokan. Metode yang digunakan dengan menggunakan metode pendekatan rekonstruksi sehingga dapat direkonstruksi kembali dan diklasifikasi. Pada tahap rekonstruksi terdapat metode untuk perkiraan pembagian kelas, kemudian pada tahap klasifikasi digunakan metode *norms* dan *quasi-norms*. Hasil dari klasifikasi akan diberi label, kemudian akan dilakukan *face detection* dengan menggunakan *Principal Component Analysis* (PCA) dan pendekatan model menggunakan Gaussians untuk pemisahan *background* dengan objek yang ada. Langkah terakhir menggunakan metode morfologi seperti *erosion* dan *dilation* untuk mengisolasi segmen dan mengisi area yang kosong. Cara validasinya adalah dengan membagi ke dalam dua cara. Pertama dengan menggunakan *dataset* berupa *AR face database*, kedua dengan menggunakan kondisi nyata. Hasil keduanya akan dibandingkan dan akan dilakukan dengan beberapa kali eksperimental untuk mengoptimalkan hasil. Hasil dari penelitian ini sistem dapat mengenali gambar kurang dari satu detik. Dari jurnal diatas, yang penulis dapatkan adalah dengan memisahkan antara latar belakang dan subjek yang akan dideteksi sehingga proses pengenalan wajah lebih efektif untuk mengenali wajah pengguna.

2.1.3. Masked Face Recognition Using Convolutional Neural Network [8]

Masalah utama pada studi ini adalah penggunaan masker atau barang sejenis yang dapat menghalangi area wajah sehingga *face recognition* tidak dapat mengenali pengguna tersebut. Beberapa contoh barang yang membuat *face recognition* tidak dapat mengenali adalah masker medis, topi, kacamata hitam, dan syal yang menutupi area wajah. Adanya kumis, jenggot, dan beberapa makeup wajah juga memperburuk *face recognition* untuk mengenali pengguna. Tujuan dari penelitian ini adalah untuk peningkatan performa *face recognition* dalam mengenali pengguna. Tujuan lainnya adalah untuk mengidentifikasi wajah seseorang yang dianggap kriminal sehingga dapat dikenali walaupun wajahnya terhalang oleh beberapa benda. Metode yang digunakan dengan menggunakan *deep learning*. Pada

deep learning terdapat metode berupa *Convolutional Neural Network* (CNN) yang digunakan untuk klasifikasi pada *dataset*. Langkah selanjutnya dengan menggunakan *Multi-task Cascaded Convolutional Neural Network* (MTCNN). MTCNN adalah pengembangan dari CNN. Cara kerja dari MTCNN dengan mendeteksi garis wajah dari gambar yang disediakan kemudian membuat model jika wajah pengguna dihalangi. MTCNN bekerja secara paralel sehingga dapat memberikan hasil yang lebih cepat. Setelah menggunakan MTCNN, akan dilakukan *image post-processing*. *Image post processing* digunakan untuk memberikan *bounding box* serta memperkecil ukuran gambar yang sudah ada. Selanjutnya *feature extraction* menggunakan FaceNet. FaceNet berfungsi untuk identifikasi gambar yang sudah diolah lalu akan dilakukan verifikasi wajah dengan menggunakan *Support Vector Machine* (SVM). Cara validasinya adalah dengan menggunakan beberapa *database* seperti *AR face database*, *IIIT Disguise Face Database*, dan *database* buatan sendiri. *Database* akan dijalankan dengan beberapa skenario yang kemudian akan dilakukan percobaan untuk melihat bagaimana tingkat akurasi dari skenario tersebut. Hasil dari penelitian ini memberikan tingkat akurasi sebesar 63,52% sampai 98,10%. Dari penjelasan jurnal diatas, dapat disimpulkan bahwa algoritma SVM dapat digunakan untuk mengidentifikasi seseorang dalam *software* pengenalan wajah.

2.1.4. 3-D Face Recognition Under Occlusion Using Masked Projection [9]

Masalah utama pada studi ini adalah untuk mengidentifikasi pengguna dengan adanya halangan pada wajahnya membuat kerja *face recognition* menjadi terhambat. Tujuan dari penelitian ini membuat *3D face recognition* sehingga wajah pengguna dapat teridentifikasi walaupun ada penghalang. Metode yang digunakan adalah *masked projection*. *Masked projection* adalah sebuah metode untuk analisis sub-ruang dengan data yang tidak lengkap. Pada tahap ini akan dilakukan registrasi maupun penghapusan penghalang yang ada pada wajah pengguna. Pada tahap ini juga digunakan *database* dari Bosphorus dan the UMB-DB *databases*. *Masked projection* menggunakan algoritma *Iterative Closest Point* (ICP) untuk mendeteksi hidung pengguna. Hal ini disebabkan hidung merupakan area yang paling efektif untuk dideteksi. Langkah selanjutnya, peneliti membuat geometri untuk pemetaan

wajah pengguna. Kemudian, objek penghalang pada wajah yang sudah teridentifikasi akan dihilangkan. Terakhir mengklasifikasikan, memvalidasi, dan mengidentifikasi. Cara validasinya adalah dengan membandingkan hasil dari *database* yang ada. Hasil dari penelitian memberikan penggunaan *3D face recognition* memberikan hasil yang lebih baik dibandingkan *2D face recognition*. Dari penjelasan jurnal di atas dapat disimpulkan, dengan mendeteksi area hidung, maka proses deteksi lebih efektif. Hal ini disebabkan bentuk hidung yang lebih mudah diidentifikasi walaupun pengguna menggunakan masker

2.2. Literatur Terkait

Terdapat beberapa literatur terkait mengenai pembuatan pada *face recognition*. Dalam pembuatan tersebut, terdapat perbedaan mendasar dengan penelitian yang sudah ada dimana pada penelitian ini menggunakan algoritma HOG. Penggunaan algoritma HOG dipilih karena jika *software* dibangun menggunakan algoritma yang lebih akurat, sehingga yang terjadi *software* tersebut tidak dapat mengenali wajah seseorang dikarenakan ada bagian fitur pada wajah yang tertutupi. *Software* ini juga bekerja dengan menggunakan kamera dua dimensi, sehingga penelitian ini dapat diaplikasikan dengan menggunakan perangkat yang lebih terjangkau.:

2.2.1. Face Detection and Recognition [10]

Pengenalan wajah adalah sebuah tugas yang dikerjakan oleh manusia secara rutin dan mudah dalam kehidupan sehari-hari. Penelitian dan pengembangan ilmu pengenalan wajah berkembang secara otomatis atas dasar ketersediaan desktop kuat dan rendah biaya serta embedded system yang telah menciptakan minat yang sangat besar dalam pengolahan citra digital dan video. Motivasi penelitian dan pengembangan dari pengenalan wajah termasuk dalam lingkup biometric, pengawasan, interaksi manusia komputer dan manajemen multimedia.

Sistem face recognition pada umumnya mencakup empat modul utama, yaitu: deteksi, alignment, ekstraksi fitur dan pencocokan. Proses lokalisasi dan normalisasi (deteksi wajah dan alignment) adalah langkah-langkah sebelum proses pengenalan wajah (ekstraksi fitur wajah dan pencocokan) dilakukan.

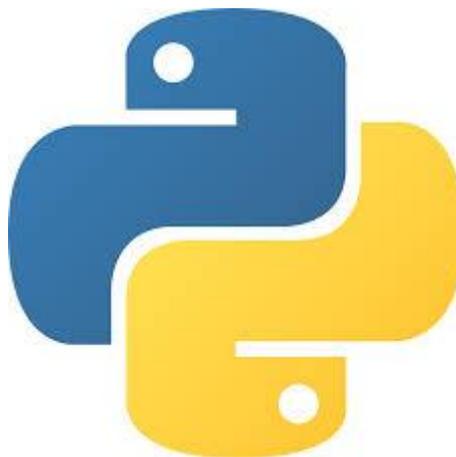
Deteksi wajah adalah langkah awal untuk melakukan identifikasi wajah atau face recognition. Sebuah pendeteksi wajah yang ideal seharusnya mampu mengidentifikasi dan menemukan lokasi dan luas semua wajah yang ada di dalam sebuah gambar tanpa memperhatikan pose, skala, orientasi, umur, dan ekspresi. Deteksi wajah melakukan segmentasi area citra wajah dengan bagian latar (background).

Proses alignment bertujuan untuk memperoleh akurasi yang lebih baik dan tinggi untuk lokalisasi dan normalisasi citra wajah sebab deteksi wajah menyediakan batas lokasi dan skala dari setiap citra wajah yang dapat terdeteksi.

Setelah sebuah wajah dilakukan normalisasi, feature extraction dilakukan untuk mengambil data yang efektif yang berguna untuk memisahkan antara citra citra wajah dan orang-orang yang berbeda satu sama lain dan cukup stabil untuk bermacam-macam geometric dan photometric. Pencocokan wajah dilakukan dengan cara melakukan pencocokan fitur yang telah diekstraksi dari citra wajah masukan dengan kumpulan data latihan dan uji coba citra wajah yang telah tersimpan sebagai database wajah.

Dari pengertian diatas, dapat dikatakan bahwa pada sistem pengenalan wajah adalah dengan menggunakan modul ini, proses pencarian wajah menjadi lebih mudah dan cepat..

2.2.2. Python [11]



Gambar 2.2.1 Logo Python

Python merupakan bahasa pemrograman tingkat tinggi (*high level language*) yang dikembangkan oleh Guido van Rossum pada tahun 1989 dan diperkenalkan untuk pertama kalinya pada tahun 1991. Python dirancang untuk memberikan kemudahan kepada *programmer* baik dari segi efisiensi waktu maupun kemudahan dalam pengembangan program dan dalam hal kompatibilitas dengan sistem. Python dapat digunakan untuk membuat program *standalone* dan pemrograman skrip (*scripting programming*).

Dalam bahasa Python sendiri terdapat bahasa tingkat rendah (*low level language*) yang berhubungan dengan bahasa mesin atau *assembly*. Dikutip menurut Sianipar dan Wadi [10], Python memiliki beberapa kelebihan dibandingkan dengan bahasa pemrograman lainnya:

1. pemrograman tingkat tinggi (*High level language*);
2. mudah dipelajari;
3. mudah digunakan;
4. mudah dalam pengembangan;
5. manajemen memori dinamis;
6. pemrograman berorientasi objek (*Object Oriented Programming*);
7. *platform independent*;
8. bersifat *open source*.

Dengan demikian, dapat disimpulkan adalah, dengan menggunakan bahasa python, pembuatan software dapat dilakukan dengan cepat. hal ini disebabkan penggunaan python yang lebih mudah dipelajari dan digunakan.

2.2.3. OpenCV [12]



Gambar 2.2.2. Logo OpenCv

OpenCV adalah singkatan dari *Open Computer Vision* yaitu *library open source* yang dikhususkan untuk melakukan *image processing*. OpenCV memiliki keuntungan menjadi kerangka *multi-platform*, mendukung Windows dan Linux dan baru-baru ini Mac OSX.

OpenCV menggunakan jenis detektor wajah yang disebut *classifier Haar Cascade*. Fungsi OpenCV yang digunakan untuk pengenalan wajah terkandung dalam beberapa modul:

1. *CalcCovarMatrix* : menghitung matriks kovarians untuk kelompok benda masukan;
2. *CalcEigenObjects* : menghitung basis ortonormal eigen dan rata-rata objek untuk kelompok benda masukan;
3. *CalcDecompCoeff* : menghitung koefisien dekomposisi objek masukan;
4. *EigenDecomposite*: menghitung semua koefisien dekomposisi untuk objek masukan;
5. *EigenProjection* : menghitung proyeksi benda ke sub-ruang eigen.

Dari penjelasan mengenai OpenCV, dapat disimpulkan bahwa dengan menggunakan modul OpenCV, pembuatan software dapat dilakukan dengan cepat. hal ini disebabkan modul OpenCV dibuat menggunakan bahasa python.

2.2.4. Tensorflow



Gambar 2.2.3 Logo Tensorflow

Dikutip pada laman developer.google.com [13] tensorflow adalah sebuah framework komputasional untuk membuat model machine learning. TensorFlow menyediakan berbagai toolkit yang memungkinkan Anda membuat model pada tingkat abstraksi yang Anda sukai.

Dikutip pada laman medium.com [14] Tensorflow dapat membantu membuat neural network (jaringan artifisial yang mirip otak manusia) dalam skala besar. Tensorflow telah membantu para ilmuwan dalam proyek-proyek seperti pencarian planet baru, membantu dokter mencegah kebutaan dalam pasien yang terkena diabetes dan lain-lain. Tensorflow juga adalah framework yang menopang proyek seperti AlphaGo dan Google Cloud Vision yang kamu dapat gunakan untuk coba.

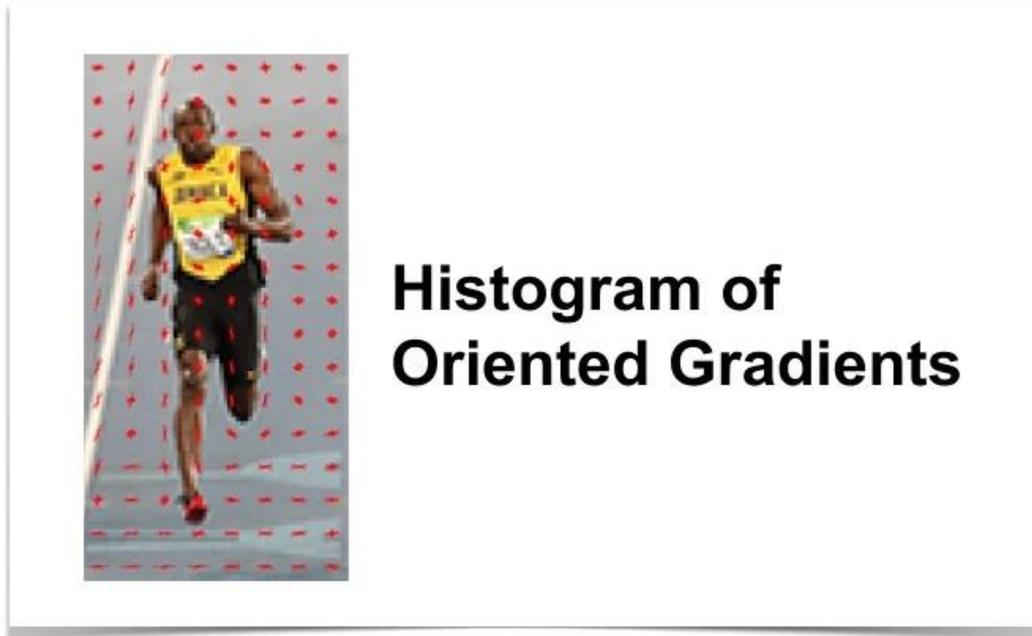
TensorFlow sendiri terdiri dari dua komponen yaitu:

1. buffer protokol grafik
2. waktu proses yang menjalankan grafik (terdistribusi)

Kedua komponen ini bersifat analog terhadap kode Python dan penafsir Python. Sama seperti penafsir Python yang diterapkan pada beberapa platform hardware untuk menjalankan kode Python, TensorFlow dapat menjalankan grafik pada beberapa platform hardware, termasuk CPU, GPU, dan TPU.

Sehingga dengan menggunakan tensorflow, pembuatan software dapat dilakukan dengan cepat. hal ini disebabkan penggunaan tensorflow yang lebih mudah dipelajari dan digunakan.

2.2.5. Histogram Of Gradient



Gambar 2.2.4 Histogram of Oriented Gradients

Dikutip pada laman OpenCV [15] Histogram of Gradient (HOG) merupakan sebuah algoritma yang terdapat di dalam artificial intelligence. HOG sendiri juga termasuk pada bagian machine learning dimana memiliki fungsi untuk mendeteksi suatu objek pada sebuah gambar. Dalam jurnal yang berjudul “Deteksi Objek Menggunakan Histogram of Oriented Gradient (HOG) untuk model Smart Room” yang ditulis oleh Robby Yuli Endra, Ahmad Cucus, [16] menjelaskan bahwa HOG merupakan konsep Histograms of Oriented Gradients ini digunakan untuk mengekstraksi fitur pada banyak objek gambar dengan menggunakan objek manusia dimana proses awal pada metode HOG dengan mengkonversi citra RGB (Red,Green,Blue) menjadi grayscale yang kemudian dilanjutkan dengan menghitung nilai gradien setiap piksel. Setelah membaginya ke dalam setiap piksel, perhitungan dilakukan dengan membagi kembali menjadi piksel yang lebih kecil untuk dilakukan perhitungan antara sumbu x dan sumbu y. Dari hasil perhitungan sumbu x dan sumbu y menghasilkan sebuah nilai gradien.

$$|G| = \sqrt{I_x^2 + I_y^2}$$

Gambar 2.2.5 Rumus Gradien

Setelah mendapatkan nilai gradien, proses selanjutnya menentukan jumlah bin orientasi yang akan digunakan dalam pembuatan histogram dimana pada proses ini disebut dengan spatial orientation binning. Sebelum pada proses gradient compute gambar dibagi menjadi beberapa cell dan dikelompokkan menjadi ukuran lebih besar yang dinamakan block. Sedang pada proses normalisasi block digunakan perhitungan geometri R-HOG. Proses ini dilakukan karena terdapat block yang saling tumpang tindih. Berbeda dengan proses pembuatan histogram citra yang menggunakan nilai-nilai intensitas piksel dari suatu citra atau bagian tertentu dari citra untuk pembuatan histogram.