

One-Shot Learning Face Recognition untuk Presensi Akademik Menggunakan Deep Convolutional Neural Network

Wailan Thom Tirajoh¹, B. Yudi Dwiandiyanta², Martinus Maslim³

Program Studi Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta Jl. Babarsari No. 43, Janti, Caturtunggal, Kec Depok, Kabupaten Sleman, 55281, Daerah Istimewa Yogyakarta, Indonesia

Email: wailantirajoh@gmail.com, yudi.dwiandiyanta@uajy.ac.id, martinus.maslim@uajy.ac.id

Abstrak. Untuk pandemi seperti sekarang ini, masyarakat lebih berhati-hati untuk kontak langsung dengan sebuah benda. Sehingga metode presensi dengan menggunakan fingerprint yang mayoritas diterapkan jadi tidak optimal. Agar tidak kontak langsung dengan mesin, pengenalan wajah dapat diterapkan sebagai pengganti proses presensi biometrik. Metode yang digunakan adalah multi-task cascaded convolutional network (MTCNN) untuk deteksi wajah dan Deep Convolutional Neural Network untuk identifikasi wajah. Perancangan aplikasi menggunakan python sebagai bahasa pemrograman, Sqlite3 sebagai basis data, Tkinter sebagai antarmuka, OpenCV & Tensorflow sebagai library pendukung, dan FaceNet & DLib sebagai framework tambahan. Aplikasi pengenalan wajah untuk presensi dapat meningkatkan proses presensi dengan cepat dan tepat karena model yang digunakan memiliki akurasi yang hampir sempurna (Labeled Faces in the Wild 99.63% & Youtube Faces DB 95.12%) sehingga presensi yang tercatat akurat.

Kata Kunci: Presensi akademik, Deep Convolutional Neural Network, Multi-Task Convolutional Neural Network, Deteksi dan Pengenalan Wajah.

1. Pendahuluan

Computer vision merupakan cabang ilmu pengetahuan yang mempelajari tentang pengenalan terhadap suatu objek yang dilakukan oleh komputer. pengembangan ilmu komputer ini didasari dengan bagaimana cara manusia melihat, dimana manusia melihat kemudian objek yang dilihat diterjemahkan otak untuk dikenali [1]. Deteksi wajah merupakan bagian dari *computer vision* yang berfungsi untuk mendapatkan koordinat wajah pada citra. Pendeteksian wajah memiliki beberapa algoritma yang dapat digunakan seperti deteksi wajah kaskade yang diusulkan oleh Viola dan Jones [2] dengan memanfaatkan *Haar-Like features* dan *AdaBoost* untuk melatih klasifikasi kaskadennya, yang mana mempunyai performa yang baik dalam penggunaan *real-time*. Deteksi wajah memiliki masalah pada variasi atau pose yang berbeda, tetapi beberapa peneliti [3][4][5] menunjukkan bahwa jenis detektor ini dapat menurunkan secara signifikan variasi wajah manusia yang berbeda. Selain menggunakan struktur kaskade, beberapa peneliti [6][7] memperkenalkan *Deformable Part Models (DPM)* untuk deteksi wajah dan memiliki performa yang luar biasa. Namun membutuhkan komputasi dan anotasi yang tinggi pada tahap pelatihan.

Lalu kemudian Convolutional Neural Network (CNN) muncul dan memiliki kemajuan yang luar biasa pada computer vision, terinspirasi dari metode deep learning dalam deteksi wajah. Beberapa pelatihan menggunakan CNN untuk deteksi wajah memiliki tumpang tindih [8][9] sampai akhirnya Multi-Task Convolutional Neural Network diusulkan yang mana ini merupakan pelatihan deteksi wajah yang lengkap dengan landmark wajah, disempurnakan dengan menggunakan teknik-teknik yang sudah dikerjakan oleh beberapa peneliti sebelumnya sehingga menghasilkan koordinat pada wajah yang pas bahkan dalam ukuran kecil pada citra [10].

Setelah mendapatkan koordinat wajah pada citra, untuk mengetahui identitas dari wajah tersebut adalah dengan melakukan pengenalan wajah. Pengenalan wajah juga mempunyai sejarah seperti deteksi wajah, mulai dari pengembangan algoritma *Local Binary Pattern*

Histogram [11], *Fisherface*, kemudian muncul *SIFT* dan *SURF*, dan yang sekarang memiliki akurasi yang baik adalah dengan menggunakan teknik *Deep Learning*, pengembangan pengembangan algoritma pengenalan wajah ini bertujuan untuk meningkatkan akurasi identifikasi wajah pada citra. Pengenalan wajah pada aplikasi ini akan menggunakan metode *Deep Convolutional Neural Network*, dan keluaran dari metode ini adalah nilai pengukuran dari wajah. Dalam proses pelatihan model, *Triplet Loss* dari *Triplet Network Deep Metric Learning* digunakan untuk mendapatkan hasil pengelompokan wajah yang baik [12]. Nilai dari pengukuran wajah ini kemudian akan diukur menggunakan perhitungan jarak euclidean untuk menentukan kemiripan wajah masukan dengan wajah yang ada dalam *dataset* [13].

Model pengenalan wajah ini dilatih oleh FaceNet dengan metode pembelajaran Euclidean Embedding menggunakan Deep Convolutional Network. Model ini dilatih menggunakan Stochastic Gradient Descent dengan standart backprop dan AdaGrad. Model ini menggunakan rectified linear units sebagai fungsi non-linear activation dan menggunakan dua arsitektur, yaitu arsitektur convolutional layers dari Zeiler & Fergus dan GoogLeNet style Inception models. Pelatihan ini berjalan dalam 275.000 langkah dan memakan waktu sekitar 30 jam dan berjalan menggunakan Nvidia Pascal Titan X GPU, Tensorflow r1.7, CUDA 8.0 dan CuDNN 6.0. Memakan waktu yang lama dan *dataset* yang besar, tetapi setelah pelatihan selesai model ini diuji pada Labeled Faces in the Wild (LFW) dan mencapai akurasi 99,63%. Dan pada YouTube Faces DB mencapai 95,12% [13].

Dengan melihat akurasi yang tinggi untuk pengelompokan wajah, sehingga metode ini sangat cocok untuk diterapkan dalam konteks pengenalan wajah untuk sistem presensi.

2. Tinjauan Pustaka

Pada tinjauan pustaka, penulis akan menerangkan penelitian yang telah dilakukan sebelumnya dan akan dijadikan referensi penulis pada penelitian ini. Penulis juga akan mempertimbangkan beberapa masukan oleh peneliti terdahulu untuk memperbaiki lagi topik yang dikerjakan. Tujuannya adalah untuk melakukan perbandingan dalam bentuk tabel, agar menemukan apa saja perbedaan dan persamaan dari penelitian ini. Dengan demikian dapat diketahui keunggulan dari penelitian saat ini dibandingkan dengan penelitian sebelumnya. Penulis belum menemukan metode *MTCNN* dan *Deep Convolutional Neural Network* yang diterapkan pada pengenalan wajah oleh beberapa peneliti sebelumnya, tetapi memiliki topik yang sama, yaitu deteksi wajah dan pengenalan wajah.

Peneliti sebelumnya dilakukan oleh Panji Purwanto, Burhanudin Dirgantoro, dan Agung Nugroho Jati pada tahun 2015 [14]. Para peneliti membuat aplikasi deteksi dan pengenalan wajah pada kamera pengawas sebagai pendeteksi bahaya, metode deteksi wajah yang dipakai pada aplikasi ini adalah Haar Cascade Classifier Like Feature yang diusulkan oleh Viola dan Jones. Metode pengenalan wajah yang digunakan penulis adalah Fisherface, terdapat tiga langkah setelah deteksi wajah dalam metode ini, antara lain adalah Principal Component Analysis (PCA), kemudian Fisher Linear Discriminant (FLD) dan klasifikasi. Modul perhitungan PCA dan FLD digunakan untuk membentuk set fisherface yang nantinya akan memberikan bobot yang berbeda-beda pada wajah. Setelah mendapatkan bobot wajah, wajah masukan akan dibandingkan dengan bobot wajah didalam *dataset* dengan menggunakan perhitungan jarak euclidean, Para peneliti ini telah melakukan pengujian pada 66 citra masukan, dan memiliki akurasi pengenalan wajah sebesar 81.82%.

Penelitian selanjutnya adalah Wibowo Joko Nuryanto pada Tahun 2017 [15]. Peneliti membuat aplikasi pengenalan wajah dengan menggunakan metode Speeded Up Robust Features. Proses pendeteksian wajah pada citra yang dilakukan adalah mulai dari menentukan parameter, kemudian memproses Fitur Haar Detect Object, lalu mengkonversi gambar kedalam bentuk citra abu-abu dan memotong gambar. *Dataset* yang digunakan memiliki jumlah 30 citra per orang, dan diuji pada 50 jumlah citra dengan tingkat ketinggian, jarak, pencahayaan yang berbeda. Menurut tabel match statistic deteksi dan recognition. Nilai persamaan tertinggi (1.0) terdapat pada gambar yang sama, dan ketika gambar yang berbeda dengan pemilik wajah yang sama memiliki nilai 0.4, bahkan ada yang nilainya hanya 0.1. Dari nilai yang ditampilkan akurasi dari pengenalan wajah menurun drastis ketika gambar yang didapatkan berbeda.

Penelitian selanjutnya dilakukan oleh Sinar Monika, Adul Rakhman, Lindawati pada tahun (2017) [16]. Para peneliti ini membuat aplikasi pengenalan wajah untuk pengaman rumah secara real-time menggunakan metode Principal Component Analysis / Eigenface. Peneliti tidak menyertakan metode deteksi wajah pada paper, dan menggunakan metode Eigenfaces untuk pengenalan wajah. Peneliti mendapatkan masalah pada faktor pencahayaan, dimana pencahayaan ini dapat menjadi faktor pembeda dalam Eigenface. Beberapa faktor lain yaitu blur, stretch, perubahan ekspresi wajah, dan pengambilan dari sudut yang berbeda. Proses pelatihan database menggunakan masing-masing 30 citra wajah per orang dari total 6 sample. Pengujian dilakukan dan mendapatkan hasil akurasi 88%. Beberapa faktor yang menjadi pokok permasalahan pada deteksi wajah menggunakan metode eigenface ini antara lain Pencahayaan, sudut pengambilan citra wajah, kemiripan nilai eigenface antar wajah, dan ekspresi wajah.

Penelitian selanjutnya dilakukan oleh Sayeed Al-Aidid, Daniel S. Pamungkas pada tahun 2018 [17]. Para peneliti membuat aplikasi pengenalan wajah dengan menggunakan algoritma Haar Cascade dan Local Binary Pattern Histogram. Metode pendeteksi wajah yang digunakan adalah Haar Cascade Classifier Like Feature yang diusulkan oleh Viola dan Jones. Metode yang digunakan untuk pengenalan wajah yaitu Local Binary Pattern Histogram, Nantinya setelah mendapatkan nilai LBPH, nilai itu akan membandingkan tiap histogram pada *dataset* dengan wajah yang akan dideteksi menggunakan perhitungan jarak euclidean. Pada proses pembuatan *dataset*, wajah per orang akan diisi dengan setidaknya 20 citra wajah dengan berbagai pose dan sudut yang berbeda. Kemudian *dataset* wajah itu akan dikonversikan menjadi citra abu-abu, setelah itu gambar dari wajah tersebut akan diekstrak untuk mendapatkan nilai histogramnya dalam bentuk array untuk masing masing orang. Disini penulis tidak menyertakan hasil analisis tingkat akurasi pengenalan wajah, penulis hanya membandingkan deteksi wajah manusia dan yang bukan wajah manusia. Wajah manusia tidak terdeteksi pada jarak kejauhan 160 cm.

3. Dataset dan Pengembangan Model

3.1. Dataset

Pada penelitian ini, penulis mengumpulkan data dari media *twitter* pada tanggal 1 Januari 2015 sampai dengan pada tanggal 01 April 2020 dengan menggunakan dua *keyword*. Dari gabungan *keyword* tersebut didapatkan data sebanyak lebih dari 14.000 data. Pengumpulan data dilakukan menggunakan *Python 3.7 (64-bit)* dengan hasil yang akan tersimpan sebagai data mentah dalam bentuk file *Microsoft Excel*. Data yang diperoleh berupa data yang belum bisa gunakan untuk dilakukan analisis sentiment sehingga perlu dilakukan pembersihan data yang disebut sebagai *Preprocessing*.

Dataset yang digunakan Google FaceNet untuk melatih network agar mendapatkan model yang baik sebanyak mulai dari 2,600,000 hingga 260,000,000 gambar dengan menggunakan The Zeiler & Fergus dan GoogleNet style Inception arsitektur untuk menghasilkan model embedding yang nantinya akan digunakan sebagai model embedding untuk melatih model pengenalan wajah.

Tabel 1. Perbandingan kinerja setelah 700 jam pelatihan

<i>Training Images</i>	<i>VAL</i>
2,600,000	76.3%
26,000,000	85.1%
52,000,000	85.1%
260,000,000	86.2%

Dataset yang digunakan untuk verifikasi wajah asli dan palsu dengan menggunakan 5,317 gambar wajah asli dan 5,255 gambar wajah palsu. *Dataset* wajah ini didapat dari ekstraksi *frame* dari video yang berisi wajah asli dan video yang berisi wajah palsu dengan menggunakan *MTCNN* dan untuk mendapatkan koordinat dan memotong bagian wajah yang terdeteksi perframe dan disimpan kedalam folder wajah asli dan wajah palsu. *Dataset* yang digunakan untuk model presensi sebanyak satu gambar wajah per *class*.

3.2. Analisis Data

Untuk *dataset* foto asli dan foto palsu, pada saat pengambilan data masih terdapat beberapa *false positive* untuk deteksi wajah sehingga diperlukan koreksi manual untuk menghapus data yang bukan wajah pada *dataset*. Juga untuk data yang salah masuk *class*, seperti data yang harusnya asli masuk ke *class* palsu kemudian harus manual dipindahkan ke *class* asli dan sebaliknya.

Untuk *dataset* yang digunakan untuk model presensi, masih terdapat wajah yang tidak tegak lurus pada kamera sehingga dapat menyebabkan model yang kurang akurat, dan dimensi foto yang berbeda sehingga dapat mempersulit model untuk mendapatkan hasil yang baik.

3.3. Preprocessing Data

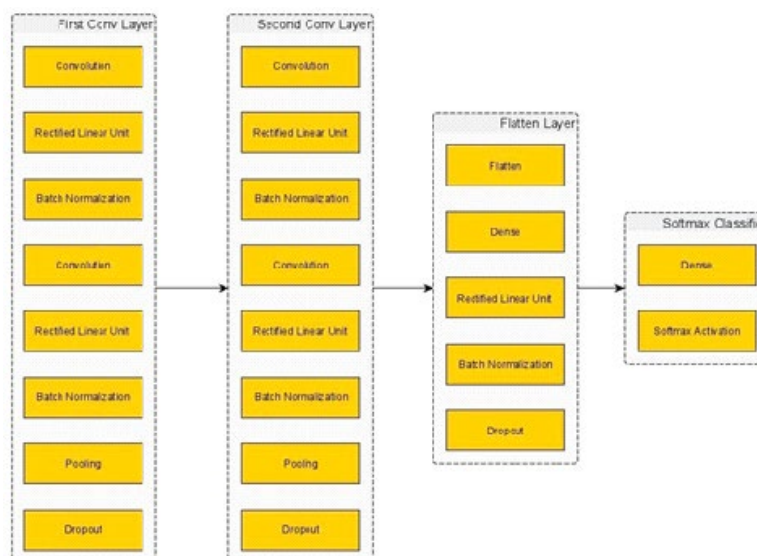
Tahap *Preprocessing* merupakan tahap awal dalam penelitian, yang bertujuan untuk memproses data citra wajah sehingga siap untuk masuk dalam tahap-tahap selanjutnya. Tahap *Preprocessing* pada penelitian ini terdiri dari meluruskan bagian wajah, redimensi citra (640x640). Tahap awal *Preprocessing* adalah meluruskan bagian wajah yang telah dipotong. Meluruskan wajah menggunakan fungsi *face align* dari *Dlib* yang mana ini adalah *face align* 2d. Tahap final *Preprocessing* adalah redimensi citra sehingga semua citra masukan memiliki ukuran yang sama untuk dilatih. Redimensi citra ini dibuat menjadi 640x640 pixel.

3.4. Pengembangan Model

Model pengenalan wajah palsu diterapkan dengan metode *Deep Convolutional Neural Network*. Model dibuat dengan menggunakan *Tensorflow Keras*. Jaringan yang dipakai mulai dari *Convolutional layer*, disusul dengan *Rectified Linear Unit* sebagai *activation function*, dan kemudian masuk lagi ke *convolutional layer*, dan terakhir adalah *POOL Layer set*. Kumpulan jaringan ini akan diulangi sebanyak dua kali dan kemudian akan diterapkan *softmax classifier*. Pelatihan ini menggunakan *one-hot encode* yang mana tingkat keyakinan dari tiap *class* yang akan menentukan keluaran model ini.

Model ini dilatih sebanyak 50 *Epoch*, menggunakan *batch_size* 8, dan 512 *dense*. 512 mungkin sangat tinggi, tapi dalam kasus ini sudah dicoba menggunakan mulai dari 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024. Dan angka yang paling cocok dan akurat dalam deteksi wajah palsu adalah dengan melakukan banyak perulangan latih jaringan sebanyak 2 kali, dan memakai 512 *dense*, untuk menjadi *memory* model yang lebih baik.

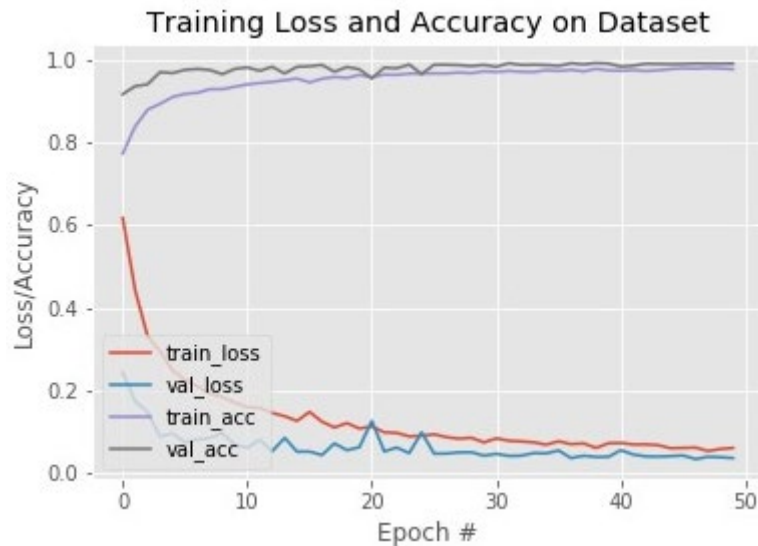
Pelatihan model menggunakan *dataset* wajah asli sebanyak 5317 gambar dan wajah palsu sebanyak 5255. Wajah palsu terdiri dari foto dan video dari *smartphone*, dan pas foto berukuran 3R.



Gambar 1. Arsitektur Deep Convolutional Neural Network untuk verifikasi wajah asli

3.4.1. Pelatihan dan Evaluasi Model

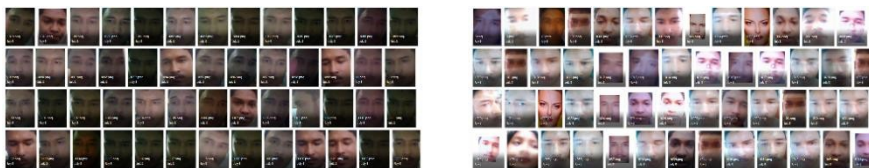
Model verifikasi wajah asli dan palsu dilatih dengan menggunakan GPU *Tesla K80* pada *Google Colab*, berjalan sekitar 20 sampai 30 menit. Evaluasi model yang diberikan adalah mencoba nilai yang berbeda pada *epoch*, *optimizer*, dan pada *dense layer*. *Epoch* dicoba mulai dari 10, 20, 30, dan 50. Dan *dense layer* dicoba mulai dari 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024. Dari percobaan tersebut dan mendapatkan akurasi yang baik terletak pada 50 *epoch* dan 512 *dense layers*. *Optimizer* yang dicoba adalah *Adam* dan *SGD*, dan yang dipakai adalah *Adam Optimizer*. Dapat dilihat di *chart* dibawah ini untuk *epoch* 30 dan 50 tidak memiliki perbedaan yang signifikan.



Gambar 2. Hasil *training* model verifikasi wajah asli dan palsu

3.4.2. Pengujian Model

Untuk pengujian model verifikasi wajah palsu dan asli, model dievaluasi dengan menggunakan video yang direkam dengan wajah asli dan wajah palsu (video / foto dari *smartphone*) dan menggunakan model untuk menentukan foto untuk dimasukkan ke dalam folder foto asli dan foto palsu. Disini jika model salah menempatkan tidak sesuai pada *classnya*, saya akan memindahkan foto yang salah tersebut ke *folder class* yang benar. Setelah menguji beberapa video pengujian dan membenarkan *class* secara manual, *dataset* yang sudah bertambah akan di latih ulang untuk memperbaiki akurasi.



Gambar 3. *Dataset* wajah asli (kiri) dan *dataset* wajah palsu (kanan)

4. Implementasi Model dan Pengujian Sistem

4.1. Implementasi Model

Proses implementasi model kedalam aplikasi adalah dengan cara memanggil model dengan *function load_model*. Disini model yang ada adalah model *FaceNet*, model *Liveness*, dan model *Classifier*. Model *FaceNet* nantinya digunakan untuk menghitung nilai pengukuran wajah pada *frame* secara *real time* untuk dihitung nilai *Euclidean*-nya dengan model pada *classifier*. Dan sebelum menghitung jarak *Euclidean*-nya, model *Liveness* terlebih dahulu dipakai untuk mengetahui wajah asli dan wajah palsu. *Feed* untuk model *liveness* adalah gambar wajah pada *frame*, setelah mendapatkan bagian wajah pada *frame*, wajah di

preprocessing terlebih dahulu secara cepat dengan menggunakan fungsi *resize* dari *opencv* dengan nilai 32x32, dan kemudian konversi gambar ke *array* menggunakan fungsi pada *opencv* juga dan selanjutnya menggunakan *expand dims* untuk mengetahui dimensi dari gambar tersebut. Setelah itu wajah dapat bias diprediksi dengan menggunakan fungsi *predict* dengan model yang sudah dibuat. Kemudian menggunakan fungsi *argmax* untuk mendapatkan posisi nilai *max* apakah itu asli atau palsu, dan kemudian untuk menampilkan asli dan palsu dengan menggunakan fungsi *classes* dengan parameter nilai dari *argmax* tersebut.

Kemudian untuk model *classifier*, gambar wajah masukkan juga terlebih dahulu akan dibersihkan setelah *dicrop* dari *frame* yang didapat, gambar akan *dirsize* 182x182, kemudian menggunakan fungsi *prewhitten* untuk mengurangi rata-rata dan menormalisasikan kisaran nilai piksel dari gambar wajah untuk membuat pelatihan menjadi lebih mudah, kemudian setelah *direshape* gambar ini yang nanti akan menjadi *placeholder* gambar masukkan untuk menghitung nilai pengukuran dengan model *FaceNet* dan kemudian nilai nilainya dihitung jarak *Euclidean distance*-nya dengan model *classifier*.

4.2. Pengujian Sistem

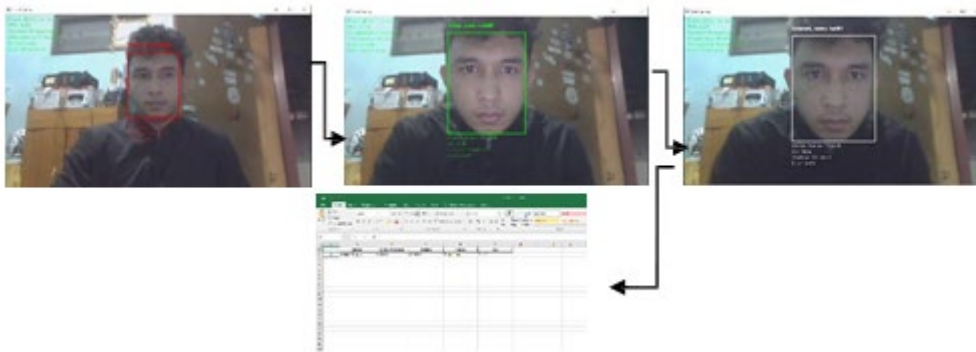
Tahap selanjutnya adalah pengujian data, dimana pengujian data ini berisi video bervariasi berdasarkan faktor faktor permasalahan yang diungkapkan peneliti sebelumnya yaitu ekspresi wajah, posisi wajah pada citra, serta pencahayaan yang dapat membuat akurasi tidak akurat. Pengujian wajah terbagi atas beberapa video yaitu: (1) wajah melihat ke atas dan kebawah, (2) jarak wajah dari kamera, (3) wajah melihat ke kanan dan ke kiri, (4) ekspresi wajah *random*, (5) Pencahayaan yang redup, (6) Deteksi wajah palsu dan asli (video), (7) Deteksi wajah palsu (Foto), (8) Wajah lurus ke kamera dan (9) Wajah Ganda.

Pengujian data dilakukan dengan *database* berisi 40 orang dan menggunakan 5 orang untuk menguji akurasi pemilik wajah, foto asli dan palsu, serta nilai *euclidean distance* wajah terhadap jarak. Pengujian data ini diambil berdasarkan hasil yang telah disimpan kedalam bentuk *chart* menggunakan *matplotlib*. Setelah proses pengujian data dilakukan akan ditampilkan hasil dalam bentuk presensi, dan hasil dari presensi berupa *excel*.



Gambar 4. Contoh pengambilan wajah beserta hasil analisis

Pada proses presensi, jika jarak wajah dari kamera melebihi dari 45 cm, maka wajah tidak akan diproses dan diberi peringatan tolong mendekat diatas wajah dengan tulisan dan *frame* warna merah. Setelah wajah memasuki jarak dibawah 45 cm dari kamera, wajah akan mulai diproses. Pemrosesan wajah ini mengambil dari perulangan *frame* yang didapat. Pada saat *frame* mencapai sepuluh maka wajah yang paling banyak terdeteksi akan dimasukkan kedalam presensi dan akan diberi warna hijau. Wajah yang sudah hadir akan diberi warna putih tidak akan masuk kedalam list presensi lagi sehingga dapat mempercepat proses presensi jika ada banyak orang yang mengantri dibelakang.



Gambar 5. Proses presensi

5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan analisis, dan hasil pengujian yang dilakukan, maka diperoleh kesimpulan yaitu: (1) Perancangan dan Pembuatan aplikasi pengenalan wajah untuk sistem presensi menggunakan metode *Deep Convolutional Neural Network* dan *MTCNN* telah dilakukan. Aplikasi ini dapat menjadi penunjang untuk dilakukan presensi biometrik dengan menggunakan wajah agar dapat menghasilkan data yang akurat dan efisien. (2) Aplikasi pengenalan wajah sudah akurat untuk mengenal pemilik wajah dengan *dataset* sebanyak 50 orang saat diuji dengan lima orang. Diuji dengan kondisi kecerahan redup hingga terang, posisi wajah yang berbeda, mimik wajah yang berubah, jarak wajah dari kamera. (3) Pendeteksi wajah palsu belum akurat dikarenakan beberapa faktor, yaitu kecerahan yang berlebihan pada wajah asli, dan kontras pada *handphone* yang redup saat ditunjukkan ke kamera.

5.2. Saran

Saran maupun masukkan yang dapat penulis sampaikan terhadap pengembangan perangkat lunak ini adalah: (1) Menambahkan *dataset* untuk melatih model pendeteksi wajah palsu, *dataset* yang perlu ditambahkan untuk minimal lima kamera yang berbeda dan orang yang bervariasi. (2) Membuat *API* sehingga aplikasi dapat terhubung dengan aplikasi lainnya sehingga dapat lebih efisien dalam presensi.

Referensi

- [1] W. S. Pambudi, B. Maria, N. Simorangkir, J. T. Elektro, U. I. Batam, and D. Obyek, "Facetracker Menggunakan Metode Haar Like Feature," vol. 2, no. 2, pp. 142–154, 2012.
- [2] P. Viola and Michael Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," pp. 1193–1197, 2001.
- [3] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Aggregate channel features for multi-view face detection," IJCB 2014 - 2014 IEEE/IAPR Int. Jt. Conf. Biometrics, 2014.
- [4] M. T. Pham, Y. Gao, V. D. D. Hoang, and T. J. Cham, "Fast polygonal integration and its application in extending haar-like features to improve object detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., no. May 2014, pp. 942–949, 2010.
- [5] Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng, "Fast human detection using a cascade of histograms of oriented gradients," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2, no. May 2014, pp. 1491–1498, 2006.
- [6] D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, "Computer Vision - ECCV 2014: 13th European Conference Zurich, Switzerland, September 6-12, 2014 Proceedings, Part IV," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Note Bioinformatics), vol. 8692, no. November, 2014.

- [7] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 2497-2504, 2014.
- [8] S. Yang, P. Luo, C. C. Loy, and X. Tang, "From facial parts responses to face detection: A deep learning approach," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, no. 3, pp. 3676-3684, 2015.
- [9] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A Convolutional Neural Network Cascade for Face Detection," *Adv. Exp. Med. Biol.*, vol. 201, pp. 17-27, 1986.
- [10] K. Zhang, Z. Zhang, Z. Li, S. Member, Y. Qiao, and S. Member, "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks Kaipeng," pp. 1-5.
- [11] A. Rahim, N. Hossain, T. Wahid, and S. Azam, "Face Recognition using Local Binary Patterns (LBP)," *Glob. J. Comput. Science Technol. Graph. Vis.*, vol. 13, no. 4, pp. 469-481, 2013.
- [12] Kaya and Bilge, "Deep Metric Learning: A Survey," *Symmetry (Basel)*, vol. 11, no. 9, p. 1066, 2019.
- [13] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 815-823, 2015.
- [14] P. Purwanto, B. Dirgantoro, and A. N. Jati, "Implementasi Face Identification Dan Face Recognition Pada Kamera Pengawas Sebagai Pendeteksi Bahaya," *eProceedings Eng.*, vol. 2, no. 1, pp. 718-724, 2015.
- [15] W. J. Nuryanto, "Pengenalan Wajah (Face Recognition) Dengan Menggunakan Metode SURF (Speeded Up Robust Features)," p. 200, 2017.
- [16] S. Monika, A. Rakhman, and Lindawati, "Pengaman Rumah dengan Sistem Face Recognition secara Real Time Menggunakan Metode Principal Component Analysis," *Pros. SNATIF Ke-4 Tahun 2017*, pp. 35-40, 2017.
- [17] 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.