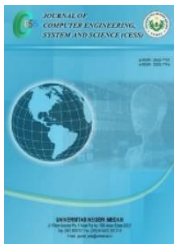
CESS e-ISSN: 2502-714x
(Journal of Computing Engineering, System and Science) 8(2) July 2023 329-338 p-ISSN: 2502-7131

Contents list available at www.jurnal.unimed.ac.id

CESS
(Journal of Computing Engineering, System and Science)

journal homepage: https://jurnal.unimed.ac.id/2012/index.php/cess



Sistem Deteksi Wajah Palsu Menggunakan Arsitektur MobileNets

Fake Face Detection System Using MobileNets Architecture

Gabriel Indra Widi Tamtama[1*], I Kadek Dendy Senapartha[2]

[1,2] Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana Yogyakarta
email: [1]gabriel@staff.ukdw.ac.id, [2]dendy.prtha@staff.ukdw.ac.id

A B S T R A K

Sistem pengenalan wajah merupakan salah satu metode dalam teknik *biometric* yang menggunakan wajah untuk proses identifikasi atau verifikasi seseorang. Teknologi ini tidak memerlukan kontak fisik seperti verifikasi sidik jari dan diklaim lebih aman karena wajah setiap orang memiliki karakter yang berbeda-beda. Terdapat dua fase utama dalam sistem biometrik wajah, yaitu deteksi wajah palsu *Presentation Attack (PA)* detektor dan pengenalan wajah (*face recognition*). Penelitian ini melakukan eksperimen dengan tujuan membangun sebuah model pembelajaran mesin (*machine learning*) berbasis *mobile* untuk melakukan deteksi wajah palsu ataupun memverifikasi keaslian wajah dengan menggunakan arsitektur Mobilenets. Verifikasi keaslian wajah diperlukan untuk meningkatkan sistem pengenalan wajah sehingga bisa membedakan wajah palsu dengan asli. Wajah palsu bisa dihadirkan dengan menunjukkan rekaman video atau gambar wajah seseorang sehingga bisa memanipulasi sistem. Dengan adanya metode verifikasi wajah asli, maka keamanan sistem bisa ditingkatkan dan meminimalisir penyalahgunaan. Kami menggunakan tiga jenis *dataset* publik, yaitu Replay-Mobile, Record-MPAD, dan LLC-FSAD untuk bahan *training* terhadap model *anti-spoof* yang dibangun. Model *anti-spoof* wajah dibangun dengan menggunakan arsitektur MobilenetV2 dengan menambahkan 3 *layer neural network* yang digunakan sebagai layer klasifikasi. Kemudian pengujian secara terkontrol dilakukan dengan menggunakan program komputer menghasilkan nilai HTER 0.17. Sedangkan hasil pengujian secara tidak terkontrol menggunakan aplikasi prototipe Android menghasilkan nilai HTER sebesar 0.21. Hasil pengujian ini menghasilkan selisih nilai HTER sebesar 0.04 yang mengindikasikan bahwa model *anti-spoof* wajah akan memiliki performa yang cenderung menurun bila digunakan secara real.

Kata Kunci: *biometrik, deteksi wajah palsu, machine learning, mobilenet, pengenalan wajah*

**Penulis Korespondensi:*
email: gabriel@staff.ukdw.ac.id

A B S T R A C T
The facial recognition system is a method in biometric techniques that use faces to identify or verify a person. This technology does not require physical contact such as fingerprint verification and is claimed to be safer because everyone's face has a different character. There are two main phases in the facial biometric system, namely fake face detection (Presentation Attack (PA) detector) and facial recognition. This study conducted experiments with the aim of building a mobile-based machine learning model to detect fake faces or verify facial authenticity using the MobileNets architecture. Verification of facial authenticity is needed to improve the facial recognition system so that it can distinguish fake faces from real ones. Fake faces can be presented by showing video recordings or pictures of someone's face so they can manipulate the system. The real-face verification method can improve system security and minimize misuse. We use three types of public datasets, namely Replay-Mobile, Record-MPAD, and LLC-FSAD for training materials for the built anti-spoof model. The facial anti-spoof model is built using the MobilenetV2 architecture by adding 3 neural network layers which are used as classification layers. Then controlled testing was carried out using a computer program to produce an HTER value of 0.17. While the results of uncontrolled testing using the Android prototype application produce an HTER value of 0.21. The results of this test produce a difference in the HTER value of 0.04, indicating that the facial anti-spoof model will have performance that tends to decrease when used in real terms.

Keywords: *biometrics, fake face detection, facial recognition, machine learning, mobilenet*

1. PENDAHULUAN

Sistem pengenalan diri merupakan salah satu cabang ilmu dari biometrika yang menggunakan bagian tubuh atau perilaku manusia. Sistem akan mencari dan mencocokkan identitas seseorang dengan basis data acuan yang sudah dibuat sebelumnya melalui proses pendaftaran. Sidik jari merupakan salah satu contoh dari biometrika untuk mengenali dan mengidentifikasi seseorang berdasarkan bagian dari tubuh manusia. Beberapa contoh manfaat penerapan biometrika dapat dijumpai dalam sistem presensi kehadiran maupun sistem keamanan. Sebagai contoh, saat ini seseorang dapat melakukan proses autentikasi biometri secara otomatis hanya dengan menunjukkan wajah di depan kamera atau sistem presensi. Sistem biometrik ini diadopsi pada berbagai sektor seperti pembayaran *on-line, e-commerce,* dan sistem kontrol keamanan. Autentikasi wajah secara otomatis adalah salah satu sistem biometric yang banyak diadopsi oleh perangkat bergerak seperti Android maupun IOS [1].

Walaupun sistem autentikasi wajah saat ini memiliki performa yang baik, sistem ini masih rentan terhadap manipulasi wajah palsu dengan menggunakan media gambar tercetak, foto/video digital atau penggunaan topeng. Oleh karena itu dibutuhkan sistem deteksi wajah palsu yang mumpuni untuk dapat menangani permasalahan tersebut khususnya pada perangkat bergerak [2]. Sistem pengenalan wajah memiliki dua fase utama, yaitu proses deteksi wajah untuk mengetahui terdapat wajah atau tidak dalam sebuah frame, dilanjutkan dengan proses pengenalan wajah asli atau palsu. Proses pengenalan wajah tersebut disebut sebagai deteksi *Presentation Attack* (PA). Kedua fase tersebut saat ini sedang banyak dikaji dengan pendekatan *machine learning* [3].

Metode *machine learning* adalah pendekatan yang banyak digunakan untuk membangun model untuk melakukan klasifikasi, namun memerlukan proses pelatihan yang membutuhkan

dataset yang banyak dan kapasitas komputasi yang besar [4]. Selain itu, untuk dapat membangun sebuah model *machine learning* yang dapat mendeteksi PA, pengembang harus mengumpulkan dan menggunakan dataset dalam jumlah banyak [1]. Penelitian deteksi PA pernah dilakukan dengan menggunakan *Convolutional Neural Network* (CNN) yang di-*training* dengan menggunakan set data NUAA dan CASIA V2 [5]. Eksperimen ini menghasilkan model yang dievaluasi secara terkontrol dengan nilai akurasi sebesar 91,23% dan skor F1 sebesar 92,01%. Namun penelitian ini belum melakukan eksperimentasi pada perangkat bergerak seperti Android atau IOS sehingga belum dapat diketahui bagaimana tingkat akurasi serta performa model dalam penggunaan yang rill.

Selain itu penelitian deteksi PA pada perangkat *Internet of Things* (IoT) pernah dilakukan dan diterapkan sebagai sistem keamanan untuk mengakses suatu ruangan khusus [6]. Penelitian ini hanya berfokus pada proses *deployment* model deteksi PA pada perangkat Raspberry PI, sehingga belum mengukur akurasi dan performa model saat sistem digunakan. Penelitian lain dengan pendekatan studi literatur yang telah dilakukan ialah untuk mengetahui akurasi berbagai sistem deteksi PA pada perangkat bergerak [7]. Studi literatur ini menghasilkan bahwa akurasi dari sistem deteksi PA sangat bergantung pada algoritma dan set data yang digunakan saat membangun model deteksi PA. Selain itu akurasi sistem deteksi PA juga dipengaruhi oleh berbagai sensor yang tersedia pada perangkat bergerak seperti sensor LiDar atau *Time of Flight* untuk mengetahui *depth* dari data visual.

Berdasarkan studi literatur yang telah dilakukan, penelitian ini bereksperimen untuk membangun model deteksi PA menggunakan arsitektur MobileNets dengan pendekatan *transfer learning* yang dapat diimplementasikan pada perangkat bergerak. Untuk mengetahui performa sistem deteksi PA saat diterapkan pada perangkat bergerak maka evaluasi dilakukan dengan menggunakan metrik *Half Total Error Rate* (HTER).

2. DASAR TEORI

Sistem pengenalan autentikasi wajah, adalah metode autentikasi biometrik yang memanfaatkan fitur wajah untuk memverifikasi dan mengautentikasi identitas individu. Pemanfaatan metode ini dalam perangkat bergerak bertujuan untuk mempermudah proses autentikasi pengguna dan *control* akses terhadap perangkat. Penerapan deteksi PA digunakan untuk meningkatkan keamanan akses perangkat saat autentikasi wajah dilakukan. Metode-metode deteksi PA yang pernah digunakan dapat dikategorikan menjadi 3 kelompok [2], yaitu:

- Metode traditional: Membangun model yang secara spesifik digunakan sebagai *feature extractor* pada deteksi PA (Contoh: LBP (*Local Binary Pattern*), SIFT (*scale-invariant feature transform*), HoG (*histograms of oriented gradients*), SURF (speeded-up robust features), and DoG (*difference of gaussian*).
- Metode learning-based: Membangun model deteksi PA menggunakan metode deep learning yang di-training dengan menggunakan banyak data dan dilakukan dengan beberapa iterasi.
- Metode Hybrid: Proses deteksi PA yang dilakukan dengan menggabungkan metode tradisional dan *learning based*. Metode tradisional diterapkan dengan menggunakan perangkat khusus (contoh: sensor *depth* atau sensor LiDar) agar dapat melakukan klasifikasi dengan lebih spesifik.

MobileNetV2 merupakan arsitektur *deep learning* yang didesain untuk perangkat dengan komputasi terbatas, seperti *smartphone* atau *embedded system* agar dapat melakukan inferensi dengan efisien dan akurat [8]. Arsitektur ini berbasis kepada struktur *inverted*

residual dimana terdapat jalur penghubung residual antara *bottleneck* setiap *layer*. Dalam MobileNet, *bottleneck* mengacu pada lapisan dengan dimensi yang lebih rendah di dalam arsitektur. Lapisan ini bertujuan untuk mengurangi beban komputasi dan jumlah parameter yang diperlukan dalam *network*. Bagian tengah *layer* dari arsitektur ini dibangun dengan teknik *depth-wise separable convolution* yang digunakan untuk memfilter *feature*.

Pada deep learning, semakin banyak layer yang ada pada sebuah model akan berdampak pada performa dan komputasi model saat melakukan inferensi data input. Dengan semakin banyaknya layer yang ada pada sebuah arsitektur, maka model akan berpotensi untuk dapat mengenali pola-pola yang lebih kompleks, mengekstraksi fitur-fitur data, dan meningkatkan kapasitas arsitektur untuk dapat belajar. Namun dengan memperbanyak layer juga akan menimbulkan masalah seperti *vanishing gradient problem*, *overfitting*, dan peningkatan kompleksitas komputasi [9]. Oleh karena itu untuk dapat menentukan jumlah layer yang tepat dalam model *deep learning* pada kasus tertentu, maka dibutuhkan eksperimen yang empiris dengan mempertimbangkan keseimbangan antara performa, sumber daya komputasi dan kompleksitas model.

Transfer learning merupakan teknik *machine learning* yang memanfaatkan pengetahuan yang sudah ada untuk meningkatkan pembelajaran dan performa pada kasus tertentu. Dengan teknik ini, model yang telah dilatih sebelumnya (*pre trained models*) digunakan kembali dan diadaptasikan untuk melakukan inferensi data yang spesifik. Dengan memanfaatkan *weight* yang sudah ada sebelumnya, maka keterbatasan data dan kebutuhan komputasi yang tinggi untuk *training* dapat diatasi. Ada dua skenario umum yang digunakan dalam *transfer learning* [4], yaitu:

1. Ekstraksi Fitur: *pre trained models* digunakan sebagai ekstraktor fitur tetap. Bobot model yang dihasilkan selama proses pelatihan disimpan dan digunakan untuk mengekstraksi fitur dari data yang baru.
2. Fine Tuning: *pre trained models* digunakan sebagai titik awal, dan seluruh model atau lapisan tertentu dilatih lebih lanjut pada kumpulan data set yang baru. Bobot *pre trained model* disesuaikan selama proses pelatihan untuk mengadaptasi model ke tugas baru.

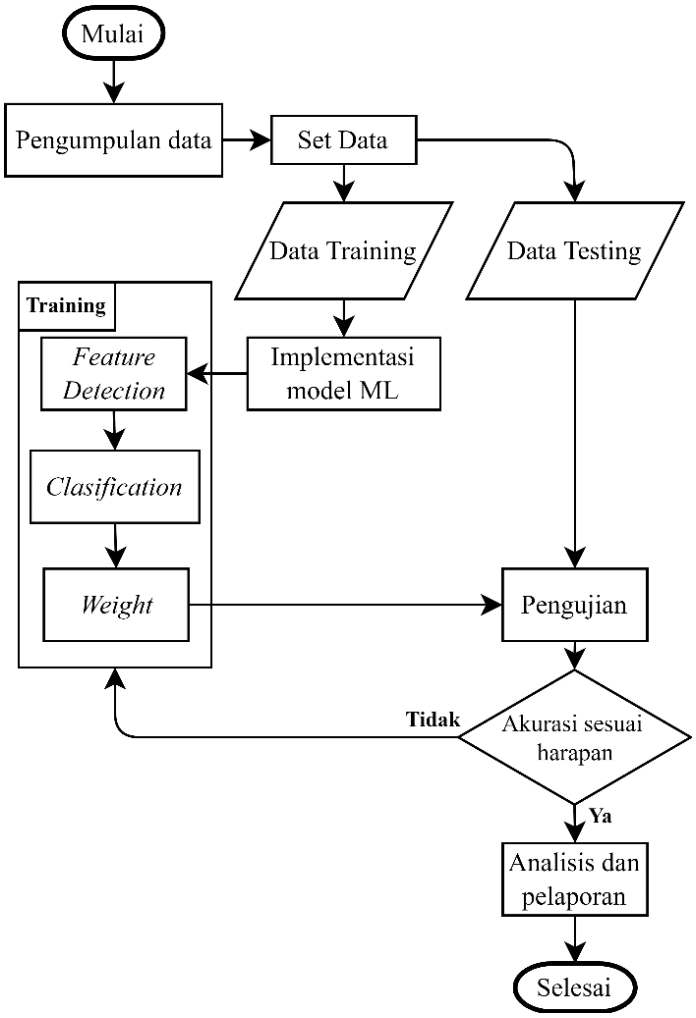
Penggunaan *transfer learning* memiliki beberapa keuntungan seperti meningkatkan performa, mengurangi waktu pelatihan, *generalization ability,* dan *robustness and regularization*. Metode *transfer learning* telah diterapkan dalam beberapa studi kasus seperti *computer vision, natural language processing,* dan pengenalan suara. Pada penelitian ini, skenario *fine tuning* digunakan untuk mempercepat proses pelatihan model *deep learning* untuk dapat mendeteksi kasus PA pada perangkat bergerak.

Untuk menguji performa model dalam mendeteksi PA, maka metrics Half Total Error Rate (HTER) digunakan. Matriks ini merupakan matriks evaluasi yang paling banyak digunakan dalam bidang *biometric authentication systems*, sehingga dapat digunakan untuk menguji performa sistem dalam mendeteksi wajah asli dan palsu [10]. Nilai HTER didapat dengan cara menghitung rata-rata dari *False Rejection Rate* (FRR) dan *False Acceptance Rate* (FAR). FRR adalah rata-rata kesalahan model melakukan kesalahan saat menolak akses wajah asli. Sedangkan FAR adalah rata-rata kesalahan model melakukan kesalahan saat menerima akses wajah palsu. Nilai HTER diformulasikan pada formula 1.

$$HTER = \frac{FRR+FAR}{2} \quad (1)$$

3. METODE

Gambar 1 merupakan metode penelitian yang dilakukan, yang dimulai dari pengumpulan data, pembangunan dan implementasi model deteksi PA, proses *training*, proses pengujian dan diakhiri dengan analisis. Dataset yang digunakan adalah dataset publik, yaitu Replay-Mobile [11], Record-MPAD [12], dan LLC-FSAD [13]. Dataset REPLAY-Mobile terdiri dari 1040 data video, RECOD-MPAD memiliki 2.250 data video, dan LLC-FSAD terdiri dari 16.885 data gambar. Tahap *pre-processing* dilakukan dengan mengubah data video menjadi data gambar dan juga menyeragamkan resolusi gambar menjadi 224 x 224 piksel agar sesuai dengan konfigurasi gambar input yang akan digunakan pada model deteksi PA. Dataset ini dibagi menjadi tiga bagian, yaitu data pelatihan (*data training*), data validasi (*validation*) dan pengujian (*data test*).



Gambar 1. Tahapan Penelitian

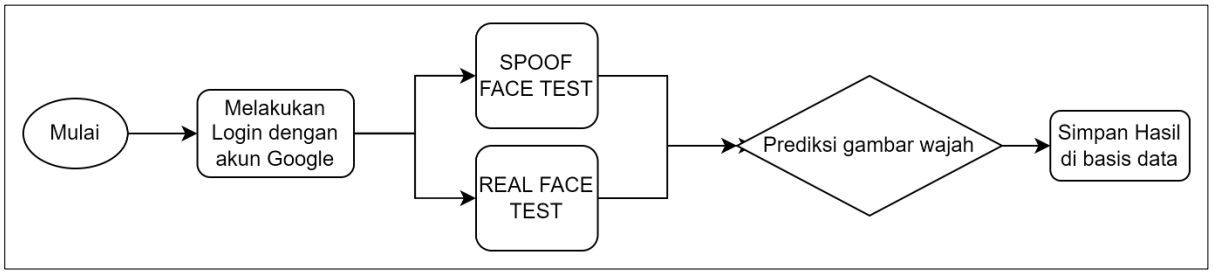
Implementasi model deteksi PA dilakukan dengan menggunakan arsitektur MobileNetsV2 sebagai fondasinya namun dengan penambahan 3-layer *neural network* yang digunakan sebagai layer klasifikasi. Konfigurasi arsitektur MobilenetV2 yang digunakan adalah input sebesar 224-pixel, *weight* dari Imagenet, dan menggunakan *global average pooling* untuk mengekstraksi fitur dari input. Tabel 1 adalah arsitektur dari model deteksi PA yang dibangun.

Pada *layer* 22 dan 23 ditambahkan konfigurasi berupa *batch normalization* untuk mempercepat proses training model, dan juga *dropouts* untuk mencegah *overfitting* [14]. Pada layer ketiga, digunakan sebuah *neuron* dengan fungsi aktivasi *sigmoid* yang digunakan sebagai layer *output* model. *Output* dari layer ini adalah bilangan pecahan yang akan digunakan untuk menterjemahkan prediksi model klasifikasi biner. Bila nilai *output* mendekati 0 berarti wajah asli, namun bila mendekati 1 berarti wajah palsu.

Tabel 1. Arsitektur model deteksi PA dengan fondasi MobileNetV2

Layer (Type)	Output Shape	Parameters
Mobilenetv2_1.00_224	(None, 1280)	2.257.984
Dense	(None, 64)	81.984
Batch_norm	(None, 64)	256
Activation	(None, 64)	0
Dropout	(None, 64)	0
Dense_1	(None, 32)	2080
Batch_norm_1	(None, 32)	128
Activation_1	(None, 32)	0
Dropout_1	(None, 32)	0
Dense_2	(None, 1)	33

Proses training model dilakukan dengan menggunakan 3 jenis dataset publik yaitu REPLAY-Mobile [10], RECOD-MPAD [11], dan LLC-FSAD [12]. Dataset REPLAY-Mobile terdiri dari 1040 data video, RECOD-MPAD memiliki 2.250 data video, dan LLC-FSAD terdiri dari 16885 data gambar. Tahap pre-processing dilakukan dengan mengubah data video menjadi data gambar dan juga menyeragamkan resolusi gambar menjadi 224 x 224 piksel.



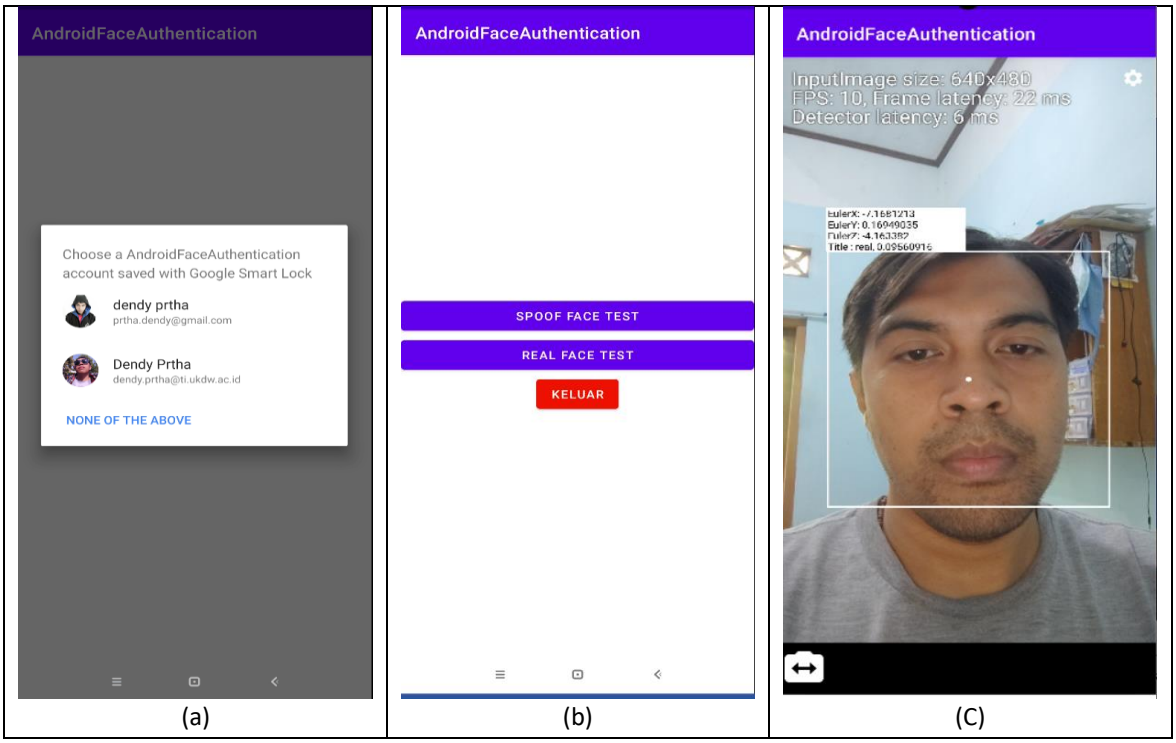
Gambar 2. Alur kerja deteksi PA pada perangkat bergerak

Untuk dapat mengintegrasikan model yang dicapai dengan aplikasi Android, maka model deteksi PA wajah yang telah training dengan framework Tensorflow [15]. Proses *login* terlebih dahulu harus dilakukan sebelum pengguna melakukan pengujian dengan menggunakan aplikasi prototipe. Ini dimaksudkan untuk membedakan hasil pengujian setiap user sehingga dapat dilakukan analisa data secara lebih mendetail. Gambar 2 merupakan desain alur kerja prototipe aplikasi deteksi PA pada perangkat bergerak.

4. HASIL DAN PEMBAHASAN

Arsitektur pada table 1 di-*training* dengan menggunakan 3 dataset publik dan menghasilkan model dengan jumlah parameter sebanyak 2,342,465. Prototipe aplikasi Android dibangun untuk mengevaluasi kinerja model deteksi PA wajah. Untuk mengintegrasikan model pada

aplikasi Android, maka model harus dikonversi menjadi format *.tflite*. Seperti diagram alur deteksi PA yang digambarkan pada Gambar 2, maka kerja sistem dimulai dari proses *login* yang dilanjutkan dengan memilih jenis pengujian yang dilakukan. Terdapat 2 pilihan jenis pengujian yaitu Spoof Face Test yang digunakan untuk menguji performa model terhadap input gambar PA dan Real Face Test untuk menguji performa model terhadap input gambar wajah asli. Setelah jenis pengujian dipilih pengguna memposisikan kamera depan/belakang menghadap wajah atau gambar untuk mendeteksi gambar wajah adalah asli atau palsu.



(a) (b) (C)

Gambar 3. (a) Tampilan *Login* aplikasi deteksi PA, (b) Tampilan pemilihan jenis pengujian dan (c) Tampilan proses inference gambar wajah.

Evaluasi dilakukan dengan dua cara yaitu pengujian terkontrol menggunakan komputer dan pengujian tidak terkontrol menggunakan aplikasi Android. Untuk mengevaluasi model secara terkontrol, set data khusus dibangun dengan cara merekam wajah-wajah secara *real*. Set data dibangun dengan merekam wajah 10 orang subyek dengan tingkat pencahayaan sebesar 250 *lux* dengan durasi 15-30 detik. Hasil rekaman video wajah kemudian di-*preprocessing* dan kemudian dievaluasi dengan metrik *Half Total Error Rate* (HTER). Tabel 2 merupakan hasil evaluasi yang dilakukan dengan menguji model terhadap 4 jenis data set yang telah dipersiapkan, yaitu, REPLAY-Mobile, LLC-FASD, RECOD-MPAD dan set data khusus yang telah dibangun. Tabel ini menunjukkan nilai HTER sebesar 0.17 saat digunakan untuk membedakan set data wajah khusus yang telah dibangun.

Tabel 2. Hasil pengujian model secara terkontrol terhadap 4 set data

No	Set Data Evaluasi	FAR	FRR	HTER
1	REPLAY-Mobile	0.41	0.24	0.33
2	LLC-FASD	0.11	0.4	0.25
3	RECOD-MPAD	0.01	0.04	0.03
4	Custom Dataset	0.22	0.11	0.17

Pengujian tidak terkontrol dilakukan dengan menggunakan perangkat Android sebanyak 1.936 kali pada berbagai pencahayaan yang berbeda. Prototipe aplikasi Android yang telah dilengkapi dengan model deteksi PA diimplementasikan pada beberapa responden yang kemudian akan melakukan pengujian secara bebas. Hasil pengujian yang telah terkumpul pada basis data Firebase dihitung dengan menggunakan metrik yang sama seperti pada pengujian terkontrol dan didapatkan nilai sebesar 0.21.

Tabel 3. Hasil pengujian model secara tidak terkontrol pada perangkat Android

No.	Pengujian	Hasil
1	Total TP	501
2	Total FP	192
3	Total TN	1063
4	Total FN	181
5	Total Pengujian	1937
6	Nilai FAR	0.15
7	Nilai FRR	0.27
8	Nilai HTER	0.21

Tabel 3 menunjukkan hasil pengujian tidak terkontrol yang dilakukan pada aplikasi Android. Arsitektur model deteksi PA menggunakan MobileNetV2 sebagai pondasinya dapat berjalan secara efisien dan ringan karena telah menggunakan *depthwise separable convolutions* dan *linear bottlenecks*. Hal ini dapat secara signifikan mengurangi kompleksitas komputasi dan jumlah parameter sehingga model yang dihasilkan juga cukup kecil yaitu berukuran 8,78 MB, memungkinkan untuk dapat dijalankan pada perangkat bergerak seperti Android.

Pada pengujian secara terkontrol dengan menggunakan set data khusus, menunjukan nilai HTER sebesar 0.17 dengan nilai FRR lebih kecil (0.11) daripada FAR (0.22). Ini berarti bahwa model deteksi PA cenderung lebih banyak melakukan kesalahan dengan mendeteksi wajah palsu sebagai wajah asli. Sedangkan pada pengujian secara tidak terkontrol menggunakan aplikasi Android menunjukan nilai HTER sebesar 0.21 dengan nilai FRR sebesar (0.27) dan FAR sebesar (0.15). Ini mengindikasikan bahwa model deteksi PA cenderung salah mendeteksi wajah asli sebagai wajah palsu. Sehingga pada kasus ini sistem deteksi PA akan dapat mengurangi tingkat kenyamanan pengguna karena sering kali ditolak saat menginputkan wajah asli untuk melakukan autentikasi.

5. KESIMPULAN

Model deteksi PA wajah yang dibangun dengan menggunakan arsitektur MobileNetV2 sebagai fondasi berhasil dibangun dan digunakan. Pengujian secara terkontrol yang dilakukan menggunakan set data khusus menghasilkan nilai HTER 0.17 sedangkan hasil pengujian secara tidak terkontrol pada aplikasi prototipe Android menghasilkan nilai HTER sebesar 0.21. Dari hasil pengujian pada dua environment yang berbeda ini dapat diketahui bahwa terdapat selisih nilai HTER sebesar 0.04. Namun perbandingan nilai FRR yang lebih besar dibandingkan FAR pada pengujian tidak terkontrol mengindikasikan bahwa saat proses inferensi, model lebih berhati-hati dalam menentukan wajah asli saat mendeteksi input gambar wajah yang diberikan. Karakteristik ini dapat mengurangi tingkat kenyamanan pengguna karena sering kali

ditolak saat menginputkan wajah asli untuk melakukan autentikasi. Selain itu selisih nilai HTER ini mengindikasikan bahwa model deteksi PA memiliki performa yang cenderung menurun bila digunakan pada kasus nyata. Hal ini mungkin dapat disebabkan oleh berbagai macam faktor, dari karakteristik kamera dan sensor perangkat, warna cahaya yang terpantul pada wajah, dan juga tingkat pencahayaan pada saat pengujian tidak terkontrol dilakukan, namun hal ini perlu dilakukan pengujian yang lebih lanjut.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Lembaga Penelitian dan Pengabdian Masyarakat dan Fakultas Teknologi Informasi Universitas Kristen Duta Wacana yang telah mendukung penelitian ini baik dari segi materiil maupun non materiil sehingga penelitian ini dapat berjalan dengan lancar.

REFERENSI

[1] I. K. Dendy Senapartha and G. Indra Widi Tamtama, "Studi Literatur Presentation Attack dan Set Data Anti-Spoof Wajah," *J. Tek. Elektro*, vol. 14, no. 1, p. 8, 2022, doi: https://doi.org/10.15294/jte.v14i1.36108.

[2] C. Kong, S. Wang, and H. Li, "Digital and Physical Face Attacks: Reviewing and One Step Further." arXiv, Sep. 29, 2022. Accessed: Oct. 26, 2022. [Online]. Available: http://arxiv.org/abs/2209.14692

[3] Z. Yu, Y. Qin, X. Li, C. Zhao, Z. Lei, and G. Zhao, "Deep Learning for Face Anti-Spoofing: A Survey." arXiv, May 16, 2022. doi: https://doi.org/10.48550/arXiv.2106.14948.

[4] F. Zhuang *et al.*, "A Comprehensive Survey on Transfer Learning." arXiv, Jun. 23, 2020. doi: 10.48550/arXiv.1911.02685.

[5] R. Hadiprakoso and I. Buana, "Deteksi Serangan Spoofing Wajah Menggunakan Convolutional Neural Network," *J. Tek. Inform. Dan Sist. …*, no. Query date: 2023-06-14 05:44:29, 2021, [Online]. Available: http://114.7.153.31/index.php/jutisi/article/view/4001

[6] G. Safri, D. Irawan, and R. Astutik, "Penerapan Liveness Sebagai Anti-Spoofing Citra Digital Pada Sistem Keamanan Akses Kontrol Ruang Server Berbasis Raspberry Pi," *E-Link J. Tek. Elektro Dan …*, no. Query date: 2023-06-14 05:44:29, 2021, [Online]. Available: http://journal.umg.ac.id/index.php/e-link/article/view/3333

[7] H. Agusti, "Pengenalan Wajah dengan Menggunakan Smartphone: Sistematik Review," *J. Forensik Dan Med. Indones.*, no. Query date: 2023-06-14 05:44:29, 2021, [Online]. Available: http://jos.unsoed.ac.id/index.php/jfmi/article/view/4528

[8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks." arXiv, Mar. 21, 2019. Accessed: Sep. 22, 2022. [Online]. Available: http://arxiv.org/abs/1801.04381

[9] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[10] Z. Ming, M. Visani, M. M. Luqman, and J.-C. Burie, "A Survey on Anti-Spoofing Methods for Face Recognition with RGB Cameras of Generic Consumer Devices." arXiv, Oct. 08, 2020. Accessed: Jul. 20, 2022. [Online]. Available: http://arxiv.org/abs/2010.04145

[11] A. Costa-Pazo, S. Bhattacharjee, E. Vazquez-Fernandez, and S. Marcel, "The Replay-Mobile Face Presentation-Attack Database," in *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*, Darmstadt: IEEE, Sep. 2016, pp. 1–7. doi: 10.1109/BIOSIG.2016.7736936.

[12] W. R. Almeida *et al.*, "Detecting face presentation attacks in mobile devices with a patch-based CNN and a sensor-aware loss function," *PLOS ONE*, vol. 15, no. 9, p. e0238058, Sep. 2020, doi: 10.1371/journal.pone.0238058.

[13] D. Timoshenko, K. Simonchik, V. Shutov, P. Zhelezneva, and V. Grishkin, "Large Crowdcollected Facial Anti-Spoofing Dataset," in *2019 Computer Science and Information Technologies (CSIT)*, Yerevan, Armenia: IEEE, Sep. 2019, pp. 123–126. doi: 10.1109/CSITechnol.2019.8895208.

[14] C. Garbin, "Dropout vs. batch normalization: an empirical study of their impact to deep learning," *Multimed. Tools Appl.*, vol. 79, no. 19, pp. 12777–12815, 2020, doi: 10.1007/s11042-019-08453-9.

[15] R. David *et al.*, "TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems." arXiv, Mar. 13, 2021. Accessed: Aug. 01, 2022. [Online]. Available: http://arxiv.org/abs/2010.08678