Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 5, October 2024, pp. 1423-1432

DOI: https://doi.org/10.52436/1.jutif.2024.5.5.2328
p-ISSN: 2723-3863
e-ISSN: 2723-3871

FACIAL PHOTO AUTHENTICITY DETECTION USING FACE RECOGNITION AND LIVENESS DETECTION

Bimo Vallentino Achmad[*1], Supatman[2]

[1,2]Informatics, Universitas Mercu Buana Yogyakarta, Indonesia
Email: [1]bimovallentino@gmail.com, [2]supatman@mercubuana-yogya.ac.id

(Article received: June 19, 2024; Revision: July 17, 2024; published: October 27, 2024)

Abstract

Facial recognition has been widely adopted by many systems as authentication. However, relying on facial photos for authentication is insufficient, as these can be manipulated using printed or digital photos. One method that can be used to prevent this is to integrate face recognition with liveness detection. In this research, face recognition and liveness detection are implemented using a Convolutional Neural Network (CNN) because CNN has the ability to process and extract features from photos effectively. There are two types of datasets used, namely CelebA-Spoof for liveness detection and lfw-deepfunneled for face recognition. The face recognition model achieved good accuracy and loss results of 0.9153 and 0.0514, very promising. Meanwhile, the liveness detection accuracy and loss were 0.8633 and 0.7166.

Keywords: *convolutional neural network, CNN, face recognition, liveness detection.*

DETEKSI KEASLIAN FOTO WAJAH MENGGUNAKAN FACE RECOGNITION DAN LIVENESS DETECTION

Abstrak

Penggunaan *face recognition* sudah banyak diadopsi oleh banyak sistem sebagai autentikasi. Akan tetapi, pendeteksian keaslian foto wajah seseorang tidak cukup hanya dengan *face recognition* dikarenakan *face recognition* dapat dimanipulasi menggunakan foto print atau foto digital. Salah satu metode yang dapat digunakan untuk mencegah hal tersebut adalah mengintegrasikan face *recognition* dengan *liveness detection*. Pada penelitian ini, *face recognition* dan *liveness detection* diterapkan menggunakan *Convolutional Neural Network* (CNN) karena CNN memiliki kemampuan dalam memproses dan mengekstrak ciri dari foto dengan baik. Terdapat dua jenis dataset yang digunakan yaitu CelebA-Spoof untuk *liveness detection* dan lfw-deepfunneled untuk *face recognition*. Hasil akurasi dan *loss* yang didapat dari model CNN *face recognition* sangat menjanjikan yaitu 0.9153 dan 0.0514. Sedangkan untuk *liveness detection* sebesar 0.8633 dan 0.7166.

Kata kunci: *convolutional neural network, CNN, face recognition, liveness detection.*

1. PENDAHULUAN

Kemudahan dan kenyamanan dalam mengakses suatu sistem merupakan sebuah kebutuhan yang penting pada saat ini. Penggunaan *password* sebagai proses untuk mengakses suatu sistem atau biasa disebut autentikasi, sudah banyak diimplementasikan pada sistem yang ada pada saat ini. Akan tetapi, proses autentikasi tersebut bisa dikatakan tidak mudah karena pengguna harus bisa mengingat *password* yang dimiliki dan bisa juga dikatakan tidak nyaman karena bisa diintip oleh orang lain jika pengguna tersebut tidak sadar ketika memasukkan *password*. Selain itu**,** penggunaan *password* dinilai mempunyai kelemahan karena *password* bisa ditebak oleh orang lain atau bisa diretas oleh seorang *hacker*. Sehingga diperlukan metode autentikasi lain yang bisa digunakan untuk menanggulangi kelemahan tersebut. Salah satu metode yang bisa digunakan adalah biometrik. Biometrik merupakan sebuah metode yang digunakan untuk mengenali seseorang berdasarkan fisik atau ciri — ciri orang tersebut [1]**.** Metode biometrik bekerja dengan cara memverifikasi data biometrik yang sudah disimpan dengan data yang diterima atau mengidentifikasi pengguna dengan cara membandingkan dengan seluruh data yang tersimpan [2]. Selain itu, biometrik memberikan kemudahan bagi pengguna, karena pengguna tidak perlu lagi untuk mengingat atau menyimpan data yang diperlukan untuk memasuki suatu sistem. Pengguna hanya perlu mengunakan fisik atau ciri — ciri unik pada pengguna tersebut, tergantung jenis biometrik mana yang digunakan.

Face recognition atau pengenalan wajah merupakan salah satu jenis dari biometrik yang digunakan sebagai proses validasi untuk masuk ke dalam sebuah sistem. Sudah banyak sistem yang mengadopsi *face recognition* sebagai proses validasi seperti pada sistem *lockscreen handphone*, sistem transaksi pada *m-Banking*, sistem presensi karyawan dan masih banyak lagi. Pada dasarnya, *face recognition* bekerja dengan cara mendeteksi wajah terlebih dahulu, mengekstraksi ciri dari foto yang diterima, lalu melakukan identifikasi dan verifikasi apakah foto yang diterima terdapat di dalam database yang tersimpan atau tidak [3]. Terdapat beberapa cara untuk mengimplementasikan *face recognition* seperti dengan menggunakan cara lokal, holistik dan *hybrid* [4]. Meskipun *face recognition* sudah mengalami banyak perkembangan dari tahun ke tahun, masih banyak masalah yang harus diperhatikan dalam penggunaan *face recognition*, seperti masalah keamanan. Saat ini, *hacker* atau penyerang memiliki kemampuan untuk meretas sistem dengan cerdik. Mereka dapat dengan mudah masuk ke dalam sebuah sistem yang memerlukan *face recognition* dengan cara menggunakan foto korban dari sosial media, foto wajah yang dicetak, atau dengan topeng wajah [5]. Untuk mengatasi masalah keamanan tersebut, salah satu cara yang bisa digunakan adalah dengan menerapkan *liveness detection*.

Liveness detection merupakan suatu teknik yang digunakan untuk memastikan bahwa sumber foto yang diterima merupakan foto asli dan bukan hasil tiruan oleh pihak lain yang bertujuan untuk memperoleh akses ke dalam sistem [6]. *Liveness detection* bekerja dengan cara mendeteksi terlebih dahulu keaslian foto wajah manusia. Jika *liveness detection* menganggap foto tersebut bukan wajah manusia asli, maka akan dianggap *error* dan tidak dapat melanjutkan ke tahap selanjutnya, yaitu tahap *face recognition* [7]. *Liveness detection* bisa diimplementasikan dengan beberapa teknik seperti mendeteksi berdasarkan kedipan mata [8], gerakan bibir [9], tekstur [10], menggunakan sensor khusus [11] dan sebagainya.

Penelitian yang dilakukan oleh Hadiprakoso dan rekannya yang berjudul "Deteksi Serangan Spoofing Wajah Menggunakan Convolutional Neural Network" [12] serangan *spoofing* dapat diatasi dengan menggunakan pendekatan respons mata seperti berkedip. Namun, menurut mereka pendekatan tersebut tidaklah cukup untuk mengantisipasi *spoofing* dalam bentuk gambar dinamis sehingga membutuhkan pendekatan lain. Dalam penelitian tersebut, mereka menggunakan pendekatan gerakan tangan sebagai verifikasi gambar tersebut asli. Mereka juga menambahkan perangkat berupa kamera 3-D atau *photoplethysmography*. Pada penelitian tersebut, jaringan dilatih untuk memahami kerumitan objek dan lapisan konvolusi akan mengenali atribut pada objek sehingga muncul matrik berisi angka-angka. Dari penelitian tersebut, Hadiprakoso dan rekannya menyimpulkan bahwa pendekatan *anti-spoofing* wajah efektif dilakukan dengan metode deep learning berdasarkan arsitektur CNN. Mereka mendapatkan hasil terbaik dengan parameter jumlah *epoch* 400, *learning rate* 0,01 dan tipe *optimizer* RMSProp dengan akurasi 91,23% dan skor F1 92,01%.

Penelitian lain yang dilakukan oleh Sari dan rekan-rekannya yang berjudul "Perancangan Sistem Absensi Berbasis Facial Recognition Menggunakan Algoritma CNN dan Liveness Detection (Studi Kasus: BPR Central Dana Mandiri)" [13] bertujuan untuk merancang sistem presensi berbasis *facial recognition* dengan penerapan algoritma *convolutional neural network* dan mengintegrasikan *liveness detector* agar sistem *facial recognition* dapat membedakan antara wajah asli seseorang dengan wajah yang berasal dari data *image*. Hasilnya sistem dapat mengenali seluruh wajah yang berasal dari database dengan tingkat akurasi sebesar 100%, sedangkan pengenalan wajah terhadap wajah-wajah acak yang didapatkan dari internet memiliki tingkat akurasi sebesar 73,33%.

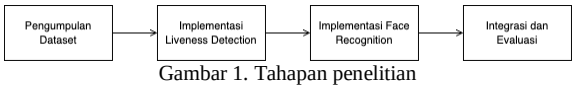
Penelitian lain yang dilakukan oleh Liu dan rekan-rekannya yang berjudul "An Identity Authentication Method Combining Liveness Detection and Face Recognition" [14] membahas penggunaan *liveness detection* dan *face recognition* untuk melakukan autentikasi identitas seseorang. Pada penelitian tersebut, sensor *kinect* digunakan untuk mengambil foto *infrared radiation* (IR) yang nantinya digunakan untuk *liveness detection*. Foto IR yang didapat tadi akan diekstrak cirinya dan diklasifikasikan menggunakan *Convolutional Neural Network* (CNN). Lalu, untuk model *face recognition* yang dipakai adalah model FaceNet yang ditingkatkan performanya sehingga bisa mendapatkan hasil yang lebih baik. Hasil dari model CNN yang dibuat untuk *liveness detection* dan *face recognition* mencapai tingkat akurasi sebesar 99,8% dan 99,7%.

Berdasarkan pemaparan di atas, model *liveness detection* dan *face recognition* akan dikembangkan dengan mengimplementasikan metode *Convolutional Neural Network* (CNN) pada penelitian ini. CNN merupakan salah satu bagian dari *deep learning* yang mempunyai kemampuan untuk memproses foto dengan baik. Selain itu, CNN juga dapat memberikan hasil yang luar biasa pada segmentasi, klasifikasi, dan pendeteksian foto [15]. CNN juga mempunyai kemampuan yang kuat dan efisien dalam melakukan ekstraksi dari sebuah foto beserta pengklasifikasiannya [16]. Keaslian wajah seseorang ditentukan dari foto wajah yang dikirim dan terdeteksi sebagai foto wajah yang nyata, diedit, maupun yang telah ditiru. Setelah dinyatakan asli atau nyata, akan dilakukan identifikasi dan verifikasi apakah foto tersebut dikenali atau tidak. Dengan demikian, metode CNN dipilih pada pengembangan *liveness detection* dan *face recognition* agar bisa

dengan akurat menentukan keaslian wajah dari foto yang diterima.

2. METODE PENELITIAN

Penelitian ini dimulai dari mengkaji studi serupa yang telah ada sebelumnya untuk mencari dasar teori yang kredibel. Kajian tersebut akan memberikan pemahaman terkait *face recognition*, *liveness detection*, dan CNN [12,13,14]. Secara garis besar, tahapan dari penelitian ini dimulai dari Pengumpulan Dataset, Implementasi *Liveness Detection*, Implementasi *Face Recognition* dan Integrasi serta Evaluasi yang ditunjukkan pada Gambar 1.



Gambar 1. Tahapan penelitian

2.1. Pengumpulan Dataset

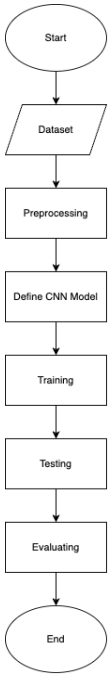
Pada proses *liveness detection*, dataset yang digunakan merupakan data sudah terpublikasi yaitu CelebA-Spoof [17]. CelebA-Spoof merupakan dataset yang mempunyai jumlah total 625.537 gambar dengan 10.177 subjek, meliputi 43 atribut pada wajah, pencahayaan, lingkungan, dan tipe *spoof* [17]. Dataset ini akan dibagi menjadi dua jenis, yaitu *live* dan *spoof*. *Live* merupakan gambar yang dianggap asli, sedangkan *spoof* merupakan gambar yang dianggap serangan atau palsu. Dikarenakan terdapat keterbatasan pada perangkat, maka dilakukan pemilahan dataset secara acak untuk menyeimbangkan proporsi dari spoof maupun live. Jumlah dataset yang digunakan sebesar 60.000, dengan jumlah spoof sebanyak 30.000 dan jumlah live sebanyak 30.000.

Sedangkan untuk *face recognition* dataset yang digunakan juga merupakan dataset yang sudah terpublikasi yaitu lfw-deepfunneled [18]. Dataset lfw-deepfunneled mempunyai jumlah total foto sebanyak 13.226 dengan label atau nama orang sebanyak 5.747. Data dipilih dari masing – masing label setidaknya berjumlah dua foto. Sehingga jumlah total foto yang digunakan sebanyak 9.131 dengan 1.674 label.

2.2. Implementasi Liveness Detection

2.2.1. Proses Pelatihan

Pada proses penggunaan *liveness detection*, terdapat beberapa proses yang harus dilakukan mulai dari proses pemilihan dataset hingga evaluasi. Untuk proses *liveness detection* ditunjukkan pada Gambar 2.



Gambar 2. Proses *liveness detection*

2.2.2. Arsitektur CNN

CNN mempunyai beberapa komponen yaitu lapisan masukan (input), lapisan konvolusi (*convolution layer*), lapisan pooling (*pooling layer*), fungsi aktivasi (*activation function*), lapisan *fully connected* (*fully connected layer*) dan lapisan keluaran (*output layer*) [19]. Arsitektur yang digunakan untuk membuat model CNN *liveness detection* terdiri dari lapisan konvolusi yang dilanjutkan dengan fungsi aktivasi ReLU, *batch normalization*, *max pooling* dan *dropout*. Masing – masing lapisan tersebut digunakan sebanyak empat kali sebelum dilanjutkan ke lapisan *fully connected*. Pada model ini, semua lapisan konvolusi menggunakan ukuran filter 3x3 dan jumlah filter yang diterapkan yaitu 32, 64, 128 & 128. Lalu, terdapat teknik regularisasi yang diaplikasikan agar performa model meningkatkan seperti *dropout* yang digunakan untuk menghilangkan *neuron* secara acak ketika proses pelatihan dan *batch normalization* yang digunakan untuk memastikan performa dari luaran fungsi aktivasi [20]. Selain itu, pada penelitain ini juga mengimplementasikan regularisasi L2 yang digunakan untuk meningkatkan performa model dengan cara memberikan penalti terhadap bobot yang besar [21]. Sehingga nilai bobot yang besar tersebut berubah menjadi nilai yang mendekati nol. Pada semua lapisan *max pooling* yang dipakai, ukuran dari *window* yang digunakan sebesar 2x2. Sebelum proses diteruskan ke lapisan *fully connected*, hasil luaran dari *max pooling* akan dilakukan proses *flatten* yang fungsinya untuk merubah dimensi luaran menjadi satu dimensi. Setelah proses *flatten* selesai, lapisan *fully connected* akan bertindak sebagai *classifier* pada model yang dibuat. Model *liveness detection* memerlukan luaran dari model berupa *binary*,

sehingga fungsi aktivasi *sigmoid* digunakan pada akhir lapisan *fully connected*. Untuk melihat lapisan secara lengkap, dapat dilihat pada Tabel 1.

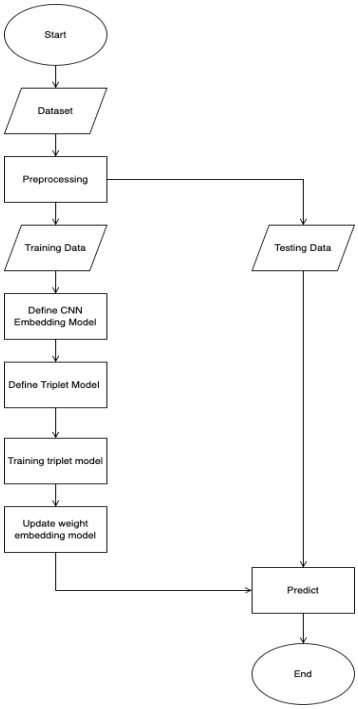
Tabel 1. Ringkasan arsitektur CNN *liveness detection*

Lapisan	Ukuran luaran	Parameter
Konvolusi	62x62x32	896
ReLU	62x62x32	
Batch Normalization	62x62x32	128
Max Pooling	31x31x32	
Dropout	31x31x32	
Konvolusi_2	29x29x64	18496
ReLU_2	29x29x64	
Batch Normalization_2	29x29x64	256
Max Pooling_2	14x14x64	
Dropout_2	14x14x64	
Konvolusi_3	12x12x128	73856
ReLU_3	12x12x128	
Batch Normalization_3	12x12x128	512
Max Pooling_3	6x6x128	
Dropout_3	32x32x32	
Konvolusi_4	4x4x256	295168
ReLU_4	4x4x256	
Batch Normalization_4	12x12x128	1024
Max Pooling_4	2x2x256	
Dropout_4	2x2x256	
Flatten	1024	
Dense	512	524800
Drouput_5	512	
Dense	1	

2.3. Implementasi *Face Recognition*

2.3.1. Proses Pelatihan

Pada proses penggunaan *face recognition*, terdapat beberapa proses yang harus dilakukan mulai dari proses pemilihan dataset, proses pendefinisian arsitektur CNN, pelatihan hingga evaluasi. Untuk proses *face recognition* bisa dilihat pada Gambar 3.



Gambar 3. Proses *face recognition*

2.3.2. Arsitektur CNN

Pada *face recognition*, arsitektur yang digunakan terdiri dari lapisan konvolusi, lalu dilanjutkan dengan fungsi aktivasi ReLU, *max pooling* dan *dropout*. Lapisan - lapisan tersebut akan digunakan sebanyak empat kali sebelum masuk ke lapisan *fully connected*. Pada lapisan *fully connected*, untuk luaran dari model CNN-nya menggunakan fungsi aktivasi ReLU sehingga menghasilkan vektor atau yang bisa disebut dengan *embedding model*, yang nantinya akan digunakan untuk kalkulasi pada proses penghitungan *triplet loss*. Model yang digunakan ini menerima foto dengan ukuran 64x64. Lalu, untuk semua ukuran dari lapisan konvolusi yang dipakai menggunakan ukuran 3x3. Akan tetapi, untuk jumlah filter yang digunakan berbeda pada di setiap lapisan konvolusi. Jumlah filter yang dipakai dimulai dari 64, 128, 256 & 512. Lalu, untuk semua lapisan max pooling menggunakan jumlah jendela yang sama yaitu 2x2. Untuk lebih jelasnya dapat dilihat pada Tabel 2.

Tabel 2. Ringkasan arsitektur CNN *face recognition*

Lapisan	Ukuran luaran	Parameter
Konvolusi	62x62x64	1792
ReLU	62x62x64	
Max Pooling	31x31x64	
Dropout	31x31x64	
Konvolusi_2	29x29x128	73856
ReLU_2	29x29x128	
Max Pooling_2	14x14x128	
Dropout_2	14x14x128	
Konvolusi_3	12x12x256	295168
ReLU_3	12x12x256	
Max Pooling_3	6x6x256	
Dropout_3	32x32x256	
Konvolusi_4	4x4x512	1180160
ReLU_4	4x4x512	
Max Pooling_4	2x2x512	
Dropout_4	2x2x512	
Flatten	2048	
Dense	512	1049088
Drouput_5	512	
Dense	128	65664

2.3.3. *Triplet Loss*

Untuk melakukan *face recognition*, ada fungsi tambahan yang digunakan untuk membantu mengenali wajah. Fungsi tersebut adalah fungsi *triplet loss*. *Triplet* dibentuk dengan cara membandingkan tiga foto dari dua *class* yang berbeda dan nantinya akan menghasilkan pasangan positif dan negatif [22]. Untuk foto *anchor*, positif, dan negatif ditunjukkan pada Gambar 4.

Gambar 4. Foto *anchor*, positif & negatif

Pada pasangan positif, foto pertama atau yang disebut dengan anchor (A) akan dipasangkan dengan foto kedua atau yang disebut dengan positif (P) untuk

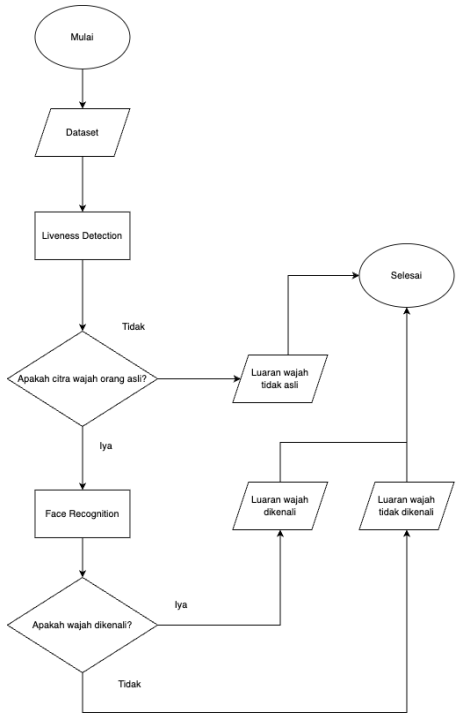
dicari jarak minimumnya. Sedangkan untuk pasangan negatif, *anchor* akan dipasangkan dengan foto ketiga atau yang disebut dengan negatif (N) untuk dicari jarak maksimumnya. Jarak tersebut akan dihitung menggunakan *euclidean distance*.

2.3.4. Penentuan Pengenalan Wajah

Untuk menentukan bagaimana masukan foto bisa dikenali, maka perlu dihitung perbandingan antara masukan foto dengan foto yang tersimpan. Setelah dilakukan prediksi terhadap foto baru dengan label yang sudah tersimpan, maka akan dihitung jaraknya menggunakan *euclidean distance*. Jika sudah mendapatkan jaraknya, maka akan dilakukan pengecekan apakah melebihi *threshold* yang ditentukan atau belum. Dalam penelitian ini *threshold* yang digunakan sebesar 0.4. Jika jarak melebihi *threshold*, maka foto tersebut akan dianggap sebagai foto yang tak dikenali.

2.4. Integrasi dan Evaluasi

Pada tahap ini, *liveness detection* dan *face recognition* yang sudah selesai diimplementasi akan diintegrasikan dan dievaluasi hasilnya, sehingga dapat mendeteksi keaslian foto wajah. Untuk proses integrasi dan evaluasi ditunjukkan pada Gambar 5.



Gambar 5. Proses integrasi dan evaluasi

3. HASIL DAN PEMBAHASAN

Pada bagian ini dapat diuraikan mengenai hasil dari penelitian beserta pengujian yang telah dilakukan. Selain itu, disampaikan juga mengenai pembahasan dari penelitian maupun pengujian yang telah dilakukan.

3.1. *Liveness Detection*

Sebelum memulai pelatihan model *liveness detection*, semua dataset *liveness detection* yang sudah dikumpulkan, ukurannya akan diubah semua menjadi 64x64. Setelah itu, dataset akan dibagi menjadi 80% untuk data pelatihan dan 20% untuk data pengujian.

Pada pelatihan ini, terdapat parameter - parameter atau biasa yang disebut *hyperparameter* yang digunakan untuk membuat model. *hyperparameter* tersebut yaitu *learning_rate*, *batch size* dan *epoch*. Untuk isi dari *hyperparameter* tersebut ditunjukkan pada Tabel 3.

Tabel 3. *Hyperparameter liveness detection*

Parameter	Nilai
Learning rate	0.00001
Batch size	32
Epoch	60

Setelah menentukan *hyperparameter*, selanjutnya adalah proses pelatihan. Pada proses pelatihan ini mengimplementasikan *optimizer* yang digunakan untuk melakukan kalkulasi dan update pada beberapa jaringan untuk mencapai nilai optimal secara bertahap, sehingga bisa meminimalisir *loss function* dan meningkatkan proses pelatihan secara iteratif [23]. *Optimizer* yang digunakan pada pelatihan model *liveness detection* ini yaitu *Adam optimizer*. Setelah menerapkan *optimizer* adam, maka hasil dari pelatihan dari data pelatihan ditujukkan pada Tabel 4.

Tabel 4. Pelatihan menggunakan data pelatihan

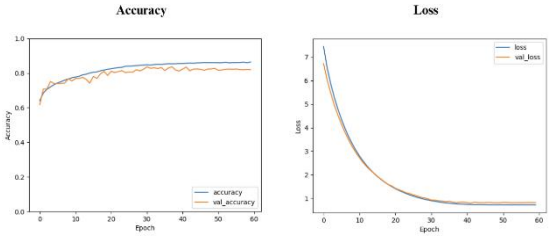
Epoch	Loss	Akurasi
1	7.4498	0.6406
2	6.5434	0.6848
3	5.8573	0.7061
…	…	…
59	0.7198	0.8599
60	0.7166	0.8633

Pada Tabel 4, akurasi yang didapat yaitu 0.7166 dan loss yang didapat yaitu 0.8633, sedangkan untuk hasil pelatihan dari data pengujian ditunjukkan pada Tabel 5.

Tabel 5. Pelatihan menggunakan data pengujian

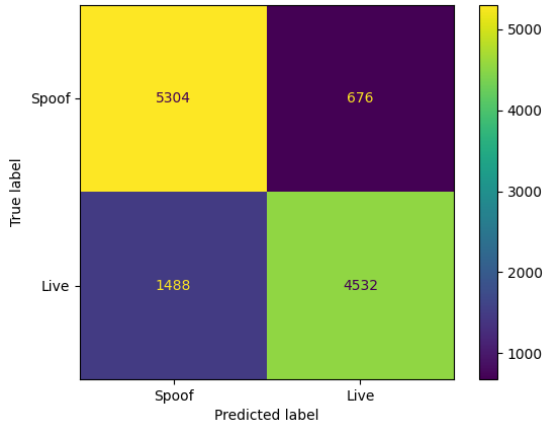
Epoch	Loss	Akurasi
1	6.7226	0.6176
2	6.0309	0.7082
3	5.4692	0.7072
…	...	...
59	0.8132	0.8207
60	0.8146	0.8197

Pada Tabel 5, akurasi yang didapat yaitu 0.8146 dan loss yang didapat yaitu 0.8197. Agar bisa mengamati data lebih mudah, maka dapat diamati pada Gambar 6.



Gambar 6. Hasil grafik pelatihan model *liveness detection*

Pada Gambar 6, ditunjukkan bahwa tingkat akurasi pelatihan semakin meningkat pada setiap *epoch*. Hal ini selaras dengan tingkat akurasi validasi yang mengikuti perkembangan dari akurasi pelatihan. Meskipun dari awal sampai pertengahan *epoch* mengalami ketidakstabilan. Namun, pada akhir *epoch*, validasi akurasi menjadi stabil. Diperkuat dengan visualisasi hasil *loss* dari pelatihan dan validasi yang bergerak turun seiring meningkatnya *epoch* serta tidak memiliki gap yang signifikan diantaranya, maka performa model dapat dikatakan bagus ketika dalam pelatihan. Hal ini dibuktikan dengan adanya pengujian model menggunakan dataset berjumlah 12000 yang tidak digunakan untuk proses pelatihan. Hasil pengujian ditunjukkan pada Gambar 7.



Gambar 7. Hasil grafik pelatihan model *face recognition*

Model dapat memprediksi dengan tepat bahwa foto tersebut merupakan *spoof* (*True Negative*) dan *live* (*True Positive*) berjumlah 5304 dan 4532 secara berurutan. Akan tetapi model memprediksi foto yang memiliki label *spoof* sebagai *live* (*false negative*) berjumlah 676. Sebaliknya label *live* sebagai *spoof* (*false positive*) berjumlah 1488. Dengan persamaan (1), maka akurasi yang didapat pada model ini sebesar 0.82.

$$acc = \frac{TN+TP}{TN+TP+FP+FN} \quad (1)$$

3.2. *Face Recognition*

Pada proses pelatihan pada *face recognition*, dataset yang sudah dikumpulkan akan dilakukan *preproceessing* terlebih dahulu. *Preprocessing* yang dilakukan adalah dengan cara mengubah ukuran foto menjadi 64x64 dan melakukan proses *crop* terhadap wajah yang terdeteksi. Setelah itu dataset dibagi menjadi dua bagian dengan data pelatihan sebesar 80% dan data pengujian sebesar 20%.

Pada saat melakukan pelatihan, data pelatihan dibagi sebanyak 20% sebagai data validasi. Selain itu terdapat *hyperparameter* yang diimplementasikan seperti *learning rate*, *batch size* dan *epoch*. *Hyperparameter face recognition* yang digunakan ditunjukkan pada Tabel 6.

Tabel 6. *Hyperparameter face recognition*

Parameter	Nilai
Learning rate	0.00001
Batch size	32
Epoch	50

Pembangunan model menggunakan data pelatihan yang telah divalidasi menghasilkan akurasi dan loss ditunjukkan pada Tabel 7 dan Tabel 8.

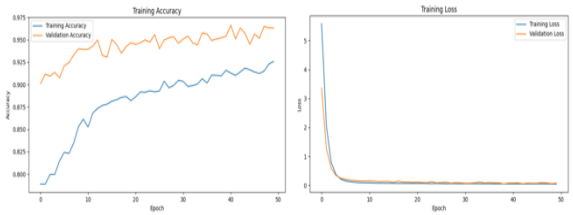
Tabel 7. Pelatihan menggunakan data pelatihan

Epoch	Loss	Akurasi
1	5.7607	0.6406
2	2.2660	0.6848
3	0.9251	0.7061
…	...	…
49	0.0473	0.8599
50	0.0514	0.8633

Tabel 8. Pelatihan menggunakan data pengujian

Epoch	Loss	Akurasi
1	3.5933	0.9112
2	1.4523	0.9105
3	0.6822	0.9032
…	...	...
59	0.0617	0.9553
60	0.0502	0.9608

Pada Tabel 7, tingkat akurasi yang didapatkan dari hasil pelatihan menggunakan data pelatihan adalah sebesar 0.8633. Lalu untuk tingkat *loss* yang didapatkan sebesar 0.0514. Sedangkan untuk pelatihan menggunakan data pengujian mendapatkan tingkat akurasi sebesar 0.9608 dan tingkat *loss* sebesar 0.0502. Hasil grafik pelatihan model *face recognition* ditunjukkan pada Gambar 8.

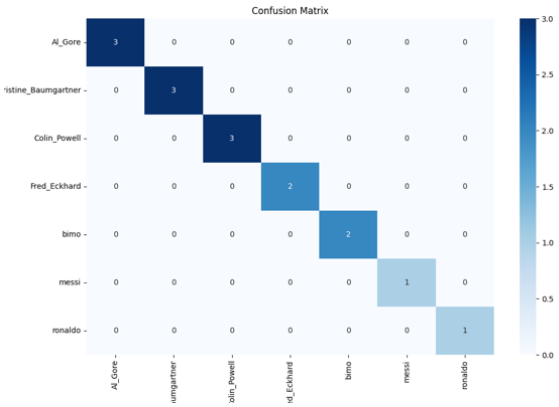


Gambar 8. Hasil grafik pelatihan model *face recognition*

Pada grafik Gambar 8, tingkat loss mengalami penurunan yang cukup drastis pada awal *epoch*, yaitu dari 5.7607 sampai 0.4387. Setelah itu, *loss* pelatihan dan validasi bergerak menurun dengan stabil hingga akhir *epoch* dengan nilai *loss* akhir sebesar 0.0514 untuk pelatihan dan 0.0502 untuk validasi. Ini merupakan pertanda yang baik dikarenakan model memiliki kesalahan yang minimal dalam mengenali data saat proses pelatihan. Tingkat akurasi mengalami

peningkatan pada setiap *epoch* dengan akurasi akhir sebesar 0.9153. Dengan kata lain, model memiliki performa yang baik dan menghasilkan *error* yang kecil.

Untuk membuktikan bahwa model memiliki performa yang baik, model diuji menggunakan sampel dari data pada dataset yang tidak digunakan dan data yang diambil secara acak dari internet. Hasilnya confusion matrix ditunjukkan pada Gambar 9.



Gambar 9. *confussion matrix* model

3.2. Integrasi dan Evaluasi

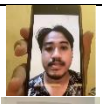
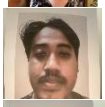
Untuk mendeteksi bahwa sebuah foto wajah itu asli atau tidak, diperlukan integrasi antar *liveness detection* dan *face recognition* seperti yang ada pada Gambar 5. Evaluasi ini nanti akan dilakukan dengan cara memberikan masukan foto yang mempunyai kategori sebagai *spoof* dan *live*. Terdapat juga foto dengan label atau nama orang tersebut yang nantinya digunakan untuk melakukan perbandingan antara foto yang tersimpan dengan masukan foto baru pada proses *face recognition*. Pada Gambar 10 terdapat tiga foto dengan indikasi *spoof* sedangkan pada Gambar 11 terdapat tiga foto dengan indikasi *live*.

Gambar 10. Foto indikasi *spoof*

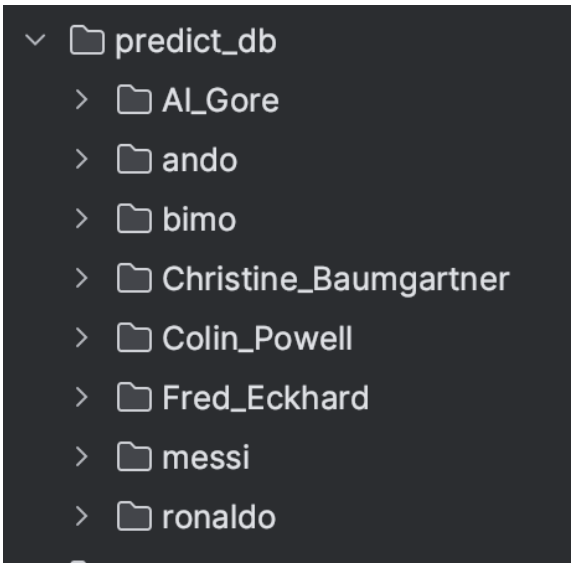
Gambar 11. Foto indikasi *live*

Foto - foto tersebut dilakukan proses liveness detection dan hasilnya dan hasilnya ditunjukkan pada Tabel 9.

Tabel 9. Hasil *liveness detection*

Foto	Hasil	Confidence level
	spoof	0.4063
	spoof	0.4566
	spoof	0.4124
	live	0.8946
	live	0.9289
	live	0.9870

Tabel 9. hasil *liveness detection* menunjukkan bahwa foto yang terindikasi sebagai *spoof* mengeluarkan hasil yang sesuai dengan kategorinya beserta *confidence* levelnya. Begitu juga yang terjadi pada foto yang terindikasi sebagai *live*. Untuk foto yang mengeluarkan hasil *live* akan lanjut ke proses selanjutnya yaitu proses *face recognition*. Pada *face recognition*, terdapat penyimpanan foto beserta labelnya yang nantinya kumpulan foto tersebut akan digunakan sebagai pembanding dari foto setelah proses *liveness detection* yang telah dilakukan. Untuk struktur penyimpanan ditunjukkan pada Gambar 12.



Gambar 12. Struktur penyimpanan *face recognition*

Hasil dari proses *face recognition* ditunjukkan pada Tabel 10.

Tabel 10. Hasil *face recognition*

Foto	Hasil	*euclidean distance*
	bimo	0.2025
	ando	0.0964
	bimo	0.2165

Pada Tabel 10, model secara tepat dapat mengklasifikasikan foto sesuai dengan label yang disimpan pada penyimpanan.

4. DISKUSI

Pada penelitian yang dilakukan ini, *custom* model untuk melakukan *liveness detection* telah berhasil dikembangkan dengan pengaturan parameter yang diterapkan seperti *learning rate* sebesar 0.00001, *batch size* sebesar 32 dan *epoch* sejumlah 60. Selain itu, untuk mengurangi generalisasi dalam mengenali foto *live* maupun *spoof* dalam data pelatihan, diimplementasikan teknik regularisasi seperti *dropout*, *batch normalization* dan L2. Lalu, dataset yang digunakan untuk pengembangan model berjumlah 48000 dan pengujian model sebesar 12000. Performa yang dihasilkan model ketika diuji menggunakan data uji mendapatkan akurasi sebesar 0.82 (82%). Sedangkan pada penelitian lain yang berjudul "Lightweight face recognition-based portable attendance system with liveness detection" [24] menggunakan dataset yang sama berjumlah 200 dan model transfer learning pada MobileNetV2, mendapatkan akurasi sebesar 92.2%. Perbandingan antara *custom* model yang dibuat dengan model yang ada pada penelitian sebelumnya membuktikan bahwa *custom* model yang dibuat mendapatkan hasil akurasi yang lebih rendah. Hal ini dapat disebabkan oleh beberapa faktor, seperti perbedaan jumlah dataset dan model yang digunakan. Model MobileNetV2 memiliki keunggulan dalam hal generalisasi karena sudah dilatih dengan dataset yang sangat besar sebelumnya, sehingga performa yang diberikan dapat melampaui *custom* model dengan dataset yang lebih kecil.

Lalu pada *face recognition*, *custom* model yang dikembangkan menggunakan parameter *learning rate* sebesar 0.00001, *batch size* sebesar 32 dan *epoch* sejumlah 50. Dataset yang digunakan terdiri dari 9131 foto dengan 1674 label. Hasil pelatihan dari *custom* model yang sudah mengimplementasikan *triplet loss* yang berupa vektor, akan digunakan untuk membandingkan foto yang diterima dengan foto yang sudah disimpan dalam database. Dua foto tersebut akan dicari jarak minimumnya menggunakan *euclidean distance* dengan *threshold* sebesar 0.4. Proses ini serupa dengan penelitian lain yang berjudul "Face Recognition Using Deep Convolutional Network and One-shot Learning"[25]. Hasil penelitian tersebut menunjukkan kemampuan untuk mengenali wajah baru dengan akurat tanpa perlu melatih ulang model, mirip dengan pendekatan yang digunakan dalam penelitian ini. Akan tetapi, penelitian ini menggunakan model dengan lapisan model CNN yang lebih sedikit dibanding penelitan sebelumnya yang menggunakan model VGG. Sehingga, model yang digunakan memilki kompleksitas yang lebih kecil yang mengakibatkan proses pelatihan pada penelitian ini bisa lebih cepat. Penggunaan model yang lebih sederhana namun cepat sangat bermanfaat untuk sistem yang menitikberatkan komputasi yang efisien.

Model dari *liveness detection* dan *face recognition* kemudian diintegrasikan untuk mengetahui keaslian foto wajah seseorang. Pada penelitian ini dilakukan pengujian enam foto wajah yang terdiri dari tiga foto *live* dan tiga foto *spoof*. Hasil dari *liveness detection* menunjukkan bahwa model dapat dengan benar mengklasifikasikan foto tersebut berupa *live* maupun *spoof*. Lalu, tiga foto *live* yang diprediksi dengan benar dilanjutkan ke tahap *face recognition*. Foto - foto tersebut akan dibandingkan dengan foto yang memiliki label dan sudah tersimpan sebelumnya di dalam database. Hasil akhir pada tahap face recognition mengindikasikan bahwa foto yang diuji dapat dikenali dengan tepat sesuai dengan labelnya. Hal ini menunjukkan bahwa model *face recognition* bekerja dengan baik dalam mengidentifikasi foto wajah yang telah ada dalam database. Kombinasi antara *liveness detection* dan *face recognition* ini memastikan bahwa sistem tidak hanya dapat mendeteksi apakah foto wajah tersebut asli, tetapi juga dapat mengidentifikasi label foto wajah tersebut.

5. KESIMPULAN

Dari hasil penelitian yang telah dilakukan untuk mendeteksi keaslian suatu foto wajah dengan cara mengintegrasikan *liveness detection* dan *face recognition* menggunakan model CNN, bisa disimpulkan bahwa model dapat melakukan deteksi secara optimal. Model CNN pada *liveness detection* memiliki akurasi sebesar 0.8633 dan *loss* sebesar 0.7166 pada akhir *epoch*. Lalu untuk model CNN pada *face recognition*, memiliki akurasi sebesar 0.9153 dan *loss* sebesar 0.0514 pada akhir *epoch*. Lebih jauh lagi, pengujian serta integrasi antara *liveness detection* dengan *face recognition* menghasilkan luaran yang sesuai.

Namun model CNN yang digunakan dapat dikembangkan lagi pada penelitian selanjutnya. Pengorganisasian dan penyeimbangan dataset, penanganan terhadap bias, pengaturan kompleksitas model dan *hyperparameter* dipertimbangkan untuk diterapkan agar model memberikan performa yang

lebih baik pada *liveness detection* maupun *face recognition*.

DAFTAR PUSTAKA

[1] A. K. Jain, D. Deb, and J. J. Engelsma, "Biometrics: Trust, But Verify," *IEEE Trans. Biom. Behav. Identity Sci.*, vol. 4, no. 3, pp. 303–323, Jul. 2022, doi: 10.1109/TBIOM.2021.3115465.

[2] P. Datta, S. Bhardwaj, S. N. Panda, S. Tanwar, and S. Badotra, "Survey of Security and Privacy Issues on Biometric System," in *Handbook of Computer Networks and Cyber Security*, B. B. Gupta, G. M. Perez, D. P. Agrawal, and D. Gupta, Eds., Cham: Springer International Publishing, 2020, pp. 763–776. doi: 10.1007/978-3-030-22277-2_30.

[3] I. Adjabi, A. Ouahabi, A. Benzaoui, and A. Taleb-Ahmed, "Past, Present, and Future of Face Recognition: A Review," *Electronics*, vol. 9, no. 8, p. 1188, Jul. 2020, doi: 10.3390/electronics9081188.

[4] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, "Face Recognition Systems: A Survey," *Sensors*, vol. 20, no. 2, p. 342, Jan. 2020, doi: 10.3390/s20020342.

[5] Y. Moon, I. Ryoo, and S. Kim, "Face Antispoofing Method Using Color Texture Segmentation on FPGA," *Security and Communication Networks*, vol. 2021, pp. 1–11, May 2021, doi: 10.1155/2021/9939232.

[6] E. A. Raheem, S. M. S. Ahmad, and W. A. W. Adnan, "Insight on face liveness detection: A systematic literature review," *IJECE*, vol. 9, no. 6, p. 5865, Dec. 2019, doi: 10.11591/ijece.v9i6.pp5865-5175.

[7] M. Basurah, W. Swastika, and O. H. Kelana, "IMPLEMENTATION OF FACE RECOGNITION AND LIVENESS DETECTION SYSTEM USING TENSORFLOW.JS," *JIP*, vol. 9, no. 4, pp. 509–516, Aug. 2023, doi: 10.33795/jip.v9i4.1332.

[8] S. R. Akhdan, R. Supriyanti, and A. S. Nugroho, "Face recognition with anti spoofing eye blink detection," Perlis, Malaysia, 2023, p. 020006. doi: 10.1063/5.0113512.

[9] M. Zhou *et al.*, "Securing Face Liveness Detection Using Unforgeable Lip Motion Patterns," 2021, doi: 10.48550/ARXIV.2106.08013.

[10] R. J. Raghavendra and R. S. Kunte, "A Novel Feature Descriptor for Face Anti-Spoofing Using Texture Based Method," *Cybernetics and Information Technologies*, vol. 20, no. 3, pp. 159–176, Sep. 2020, doi: 10.2478/cait-2020-0035.

[11] S. Rao, Y. Huang, K. Cui, and Y. Li, "Anti-spoofing face recognition using a metasurface-based snapshot hyperspectral image sensor," *Optica*, vol. 9, no. 11, p. 1253, Nov. 2022, doi: 10.1364/OPTICA.469653.

[12] X. Liu, Z. Deng, and Y. Yang, "Recent progress in semantic image segmentation," *Artif Intell Rev*, vol. 52, no. 2, pp. 1089–1106, Aug. 2019, doi: 10.1007/s10462-018-9641-3.

[13] Khatina Sari, Jasmir, and Y. Arvita, "Perancangan Sistem Absensi Facial Recognition Menggunakan CNN dan Liveness Detector pada BPR Central Dana Mandiri," *JAKAKOMUNAMA*, vol. 2, no. 1, pp. 70–80, Apr. 2022, doi: 10.33998/jakakom.2022.2.1.63.

[14] S. Liu, Y. Song, M. Zhang, J. Zhao, S. Yang, and K. Hou, "An Identity Authentication Method Combining Liveness Detection and Face Recognition," *Sensors*, vol. 19, no. 21, p. 4733, Oct. 2019, doi: 10.3390/s19214733.

[15] X. Liu, Z. Deng, and Y. Yang, "Recent progress in semantic image segmentation," *Artif Intell Rev*, vol. 52, no. 2, pp. 1089–1106, Aug. 2019, doi: 10.1007/s10462-018-9641-3.

[16] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of Image Classification Algorithms Based on Convolutional Neural Networks," *Remote Sensing*, vol. 13, no. 22, p. 4712, Nov. 2021, doi: 10.3390/rs13224712.

[17] Y. Zhang *et al.*, "CelebA-Spoof: Large-Scale Face Anti-Spoofing Dataset with Rich Annotations," 2020, doi: 10.48550/ARXIV.2007.12342.

[18] G. Huang, M. Mattar, H. Lee, and E. G. Learned-miller, "Learning to Align from Scratch," *neural information processing systems*, vol. 25, pp. 764–772, Dec. 2012.

[19] D. Bhatt *et al.*, "CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope," *Electronics*, vol. 10, no. 20, p. 2470, Oct. 2021, doi: 10.3390/electronics10202470.

[20] 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.

[21] B. I. Hairab, M. Said Elsayed, A. D. Jurcut, and M. A. Azer, "Anomaly Detection Based on CNN and Regularization Techniques Against Zero-Day Attacks in IoT Networks," *IEEE Access*, vol. 10, pp. 98427–

98440, 2022, doi: 10.1109/ACCESS.2022.3206367.

[22] S. Sharma and V. Kumar, "3D landmark-based face restoration for recognition using variational autoencoder and triplet loss," *IET Biometrics*, vol. 10, no. 1, pp. 87–98, Jan. 2021, doi: 10.1049/bme2.12005.

[23] Y. Wang, Z. Xiao, and G. Cao, "A convolutional neural network method based on Adam optimizer with power-exponential learning rate for bearing fault diagnosis," *J. vibroeng.*, vol. 24, no. 4, pp. 666–678, Jun. 2022, doi: 10.21595/jve.2022.22271.

[24] N. Surantha and B. Sugijakko, "Lightweight face recognition-based portable attendance system with liveness detection," *Internet of Things*, vol. 25, p. 101089, Apr. 2024, doi: 10.1016/j.iot.2024.101089

[25] J. Sen, B. Sarkar, A. Hena, and Md. H. Rahman, "Face Recognition Using Deep Convolutional Network and One-shot Learning," *SSRG - IJCSE*, vol. 7, no. 4, pp. 23–29, Apr. 2020, doi: 10.14445/23488387/IJCSE-V7I4P107.