

Identifikasi Wajah Asli dan Buatan Deepfake Menggunakan Metode Convolutional Neural Network

Jesselyn Mu¹, Muhammad Adrezo^{2*}, Ahmed Nizhan Haikal³

^{1,2,3}Program Studi Informatika, Universitas Pembangunan Nasional Veteran Jakarta, Jakarta Selatan, DKI Jakarta
Email: ¹2110511008@mahasiswa.upnvj.ac.id, ^{2*}muhammad.adrezo@upnvj.ac.id,
³2110511022@mahasiswa.upnvj.ac.id

(Naskah masuk: 13 Okt 2023, direvisi: 15 Jan 2024, diterima: 16 Jan 2024)

Abstrak

Di era teknologi saat ini, penggunaan dari *Artificial Intelligence* sangat berkembang pesat. Banyak teknologi yang sudah menerapkan AI untuk menghasilkan suatu informasi. Akan tetapi, tidak semua teknologi yang diciptakan menggunakan AI membawakan dampak positif. Salah satu teknologi AI yang dapat menimbulkan dampak negatif adalah *deepfake*. Salah satu dampak negatif yang dapat disebabkan oleh teknologi *deepfake* adalah membuat wajah manusia buatan yang menyerupai wajah aslinya. Oleh karena itu, *deepfake* merupakan salah satu kasus yang perlu diperhatikan lebih lanjut. Penggunaan *deepfake* yang salah ini apabila dibiarkan dapat merugikan banyak pihak, baik untuk orang lain maupun diri sendiri. Penelitian ini dibuat dengan tujuan agar dapat membuat sebuah model bersifat konseptual yang dapat melakukan identifikasi pada wajah asli maupun wajah buatan *deepfake*. Metode yang digunakan adalah metode CNN dengan pembuatan arsitektur 6 *convolutional layer* dan 3 *max pooling* serta tambahan layer *Batch Normalization*. Metode CNN dipilih karena dinilai baik dalam melakukan klasifikasi suatu citra. Setelah itu, metode CNN akan ditambahkan dengan arsitektur menyerupai VGG dan layer tambahan yaitu *Batch Normalization*. Hasil yang didapatkan pada penelitian ini adalah model yang dibuat dapat dinyatakan berhasil untuk melakukan identifikasi wajah buatan *deepfake* dengan wajah asli. Hal ini dibuktikan dengan tingkat akurasi, presisi, *recall*, dan *F1-Score* sebesar 91%.

Kata Kunci: *Deepfake, CNN, Deep Learning, Artificial Intelligence, Computer Vision.*

Identifying Deepfake Generated Faces and Real Faces Using Convolutional Neural Network

Abstract

In the current technological era, the use of *Artificial Intelligence* is growing rapidly. Many technologies have applied AI to produce information. However, not all technology created using AI has a positive impact. *Deepfake* is a great example. One of the negative impacts that can be caused by *deepfake* technology is its ability to create an artificial human face that resembles the original face. Therefore, *deepfakes* are a case that needs further attention. The wrong use of *deepfakes*, if left unchecked, can be detrimental to many parties, both to other people and to yourself. This research was created with the aim of being able to create a conceptual model that can identify real faces and *deepfake* artificial faces. The method used is the CNN method with an architecture of 6 *convolutional layers* and 3 *max pooling* as well as an additional *Batch Normalization* layer. The CNN method was chosen because it is considered good at classifying an image. After that, the CNN method will be added with an architecture similar to VGG and an additional layer, namely *Batch Normalization*. The results obtained in this research are that the model created can be declared successful in identifying between *deepfake* generated artificial faces and real faces. This is proven by the accuracy, precision, *recall* and *F1-Score* levels of 91%.

Keywords: *Deepfake, CNN, Deep Learning, Artificial Intelligence, Computer Vision.*

I. PENDAHULUAN

Pada perkembangan teknologi saat ini, *Artificial Intelligence* (AI) berkembang dengan sangat pesat. AI banyak diterapkan dalam teknologi pintar untuk menghasilkan suatu sistem otomatisasi [1]. Selain itu, AI juga dapat diterapkan dalam teknologi yang melakukan pengolahan citra dan video dengan tingkat yang sangat memuaskan. Akan tetapi, tidak semua teknologi yang menggunakan AI berdampak positif. AI juga dapat menimbulkan dampak negatif. Salah satu teknologi AI yang dapat menimbulkan dampak negatif adalah *deepfake*. *Deepfake* merupakan istilah yang digunakan untuk memanipulasi baik wajah maupun suara seseorang dari gambar atau video. Banyak dampak negatif yang dapat ditimbulkan dari penggunaan *deepfake* ini, seperti menyebarkan informasi palsu, penyalahgunaan identitas, dan pornografi [2].

Banyaknya dampak negatif yang disebabkan oleh *deepfake* tersebut tentunya membuat kekhawatiran di masyarakat. Terlebih, *deepfake* sangat berbahaya apabila diterapkan pada wajah manusia. Teknologi *deepfake* pada saat ini sudah dapat membuat gambar wajah manusia yang sulit dibedakan dengan aslinya. Oleh karena itu, diperlukan sebuah model yang dapat mengidentifikasi wajah asli dan wajah buatan *deepfake*. Pembuatan model ini diharapkan dapat mengurangi kekhawatiran masyarakat agar dapat dengan mudah mengidentifikasi wajah tersebut asli atau buatan *deepfake*. Salah satu metode yang banyak digunakan dalam melakukan identifikasi tersebut adalah *Convolutional Neural Network* (CNN) [3].

Sudah banyak penelitian yang melakukan klasifikasi terhadap wajah manusia menggunakan metode CNN. Seperti penelitian untuk melakukan klasifikasi kulit wajah yang mendapatkan akurasi sebesar 100% untuk pelatihan, 88% untuk pengujian, dan 90% untuk pengujian dengan data baru [4]. Lalu, pada penelitian untuk melakukan klasifikasi wajah bermasker mendapatkan akurasi sebesar 99,20% untuk data pelatihan dan 70,59% untuk data validasi pada 3 *layer* serta 98,20% untuk data pelatihan dan 82,35% untuk data validasi pada 5 *layer* [5]. Setelah itu, penelitian untuk sistem pengenalan wajah mendapatkan akurasi sebesar 99,84% [6]. Selanjutnya, penelitian untuk melakukan klasifikasi pada gambar bergerak untuk pengenalan wajah mendapatkan akurasi sebesar 90% untuk pelatihan dan 95% untuk validasi [7]. Terakhir, penelitian untuk melakukan klasifikasi jenis kulit wajah mendapatkan akurasi sebesar 99,51% [8]. Oleh karena itu, CNN dinilai baik dalam melakukan klasifikasi sebuah citra maupun gambar [9].

Dalam penggunaan CNN sebagai metode *deep learning* terdapat beberapa teknik untuk memaksimalkan metode *deep learning* salah satunya *Batch Normalization*. *Batch Normalization* merupakan teknik yang signifikan dalam pelatihan jaringan saraf mendalam, diperkenalkan oleh Ioffe dan Szegedy dalam penelitian berjudul "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift" [21]. Dalam penelitian ini, mereka mengatasi masalah pergeseran kovariat internal (*internal covariate shift*) dengan normalisasi *output* dari setiap lapisan dalam jaringan,

memastikan distribusi *input* relatif konstan selama pelatihan. *Batch Normalization* menghitung rata-rata dan deviasi standar dari setiap *batch* pelatihan, dan kemudian normalisasi dilakukan menggunakan parameter tersebut. Selain itu, diperkenalkan parameter penskalaan dan pergeseran yang memungkinkan model untuk mempelajari distribusi yang optimal. Teknik ini telah menjadi kontribusi berharga dalam percepatan pelatihan dan stabilisasi model *deep learning*, dan telah diadopsi secara luas dalam berbagai aplikasi.

Salah satu arsitektur yang sering digunakan yaitu arsitektur VGG (*Visual Geometry Group*) yang merupakan salah satu arsitektur dari CNN. Arsitektur ini dikenal karena pendekatan strukturnya yang mendalam dan homogen. Sehingga arsitektur ini sangat baik kinerjanya dalam tugas pengenalan gambar. Penelitian ini akan berfokus pada identifikasi wajah asli dan buatan *deepfake* menggunakan CNN dengan arsitektur VGG.

II. METODOLOGI PENELITIAN

Metode yang digunakan untuk penelitian ini adalah metode CNN (*Convolutional Neural Network*). Metode CNN bekerja dengan melakukan ekstraksi fitur dan klasifikasi citra dalam satu proses [10]. Sesuai dengan pendahuluan sebelumnya, metode CNN dipilih karena terbukti dapat menghasilkan akurasi dalam melakukan klasifikasi citra yang tinggi. Umumnya, metode CNN menggunakan 3 lapisan atau biasa disebut dengan *layer* [11]. 3 Lapisan tersebut adalah konvolusi *layer*, *pooling layer*, dan *fully-connected (dense layer) layer*. Konvolusi *layer* merupakan lapisan yang digunakan untuk mengekstrak fitur-fitur penting dari data yang dimasukkan. *Pooling layer* merupakan lapisan yang digunakan untuk mengurangi dimensi peta pada fitur yang dihasilkan dari konvolusi *layer*. Terakhir, *fully connected layer* merupakan lapisan akhir yang digunakan untuk klasifikasi. Hal ini dimaksudkan dengan, pada umumnya lapisan *fully connected* ini ditemukan di jaringan saraf tiruan biasa. Adapun pada penelitian yang dibuat, ada beberapa tahapan yang digunakan agar dapat menghasilkan sebuah sistem yang dapat melakukan identifikasi pada wajah manusia. Berikut merupakan tahapan yang dilakukan pada penelitian ini.



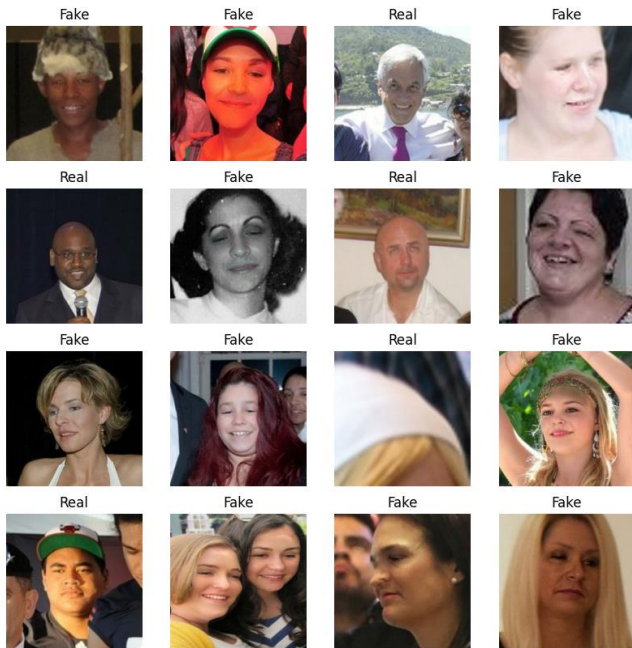
Gambar 1. Tahapan Penelitian

Sesuai dengan Gambar 1, berikut merupakan penjelasan dari masing-masing tahapan yang dilakukan:

1. Pengumpulan Data

Pada penelitian ini, data citra yang digunakan merupakan data yang didapatkan dari situs *open source* yaitu kaggle dataset *deepfake and real images* yang dikumpulkan oleh Manjil Karki pada tahun 2021 [11]. *Dataset* ini terbagi menjadi 3 bagian yaitu untuk *Test*, *Train*, dan *Validation* dengan masing-masing bagian memiliki 2 kategori yaitu *Fake* dan *Real*. *Data test* memiliki jumlah sebanyak 5.492 untuk

kategori *Fake* dan 5.413 untuk kategori *Real*. *Data train* memiliki jumlah sebanyak 70 ribu baik untuk kategori *Fake* maupun kategori *Real*. *Data validation* memiliki jumlah sebanyak 19,6 ribu untuk kategori *Fake* dan 19,8 ribu untuk kategori *Real*. Gambaran dari dataset yang digunakan dapat dilihat pada Gambar 2.



Gambar 2. Citra Penelitian

2. Rancangan Model

Rancangan model konsep pada penelitian ini akan dibuat dengan menggunakan arsitektur yang terinspirasi dari VGG dengan tambahan *layer Batch Normalization* yang nantinya akan diterapkan setelah masing-masing *layer* konvolusi. Adapun *Batch Normalization* merupakan sebuah langkah yang digunakan untuk menjaga kestabilan distribusi maupun mengurangi variasi *input* ke lapisan jaringan selama proses pelatihan. *Batch Normalization* digunakan karena dapat mempercepat proses pelatihan karena nilai *activation* yang digunakan stabil [12]. Persamaan *Batch Normalization* ditunjukkan pada Persamaan 1 dan 2.

$$x' = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{1}$$

$$y = \gamma x' + \beta \tag{2}$$

Pada Persamaan 1, x' merupakan aktivasi yang telah dinormalisasi. Dimana, nilai ini didapatkan dari x yang merupakan vektor aktivasi dari lapisan terdahulu, μ merupakan nilai rata-rata dari x di seluruh *batch* pelatihan, σ merupakan deviasi standar dari x di seluruh pelatihan, dan ϵ merupakan nilai kecil yang ditambahkan ke standar deviasi untuk menghindari pembagian dengan angka 0. Sementara, pada Persamaan 2, y adalah aktivasi yang telah ditransformasi kembali dengan parameter γ merupakan parameter

pembelajaran yang dapat membantu jaringan untuk melakukan pengukuran ulang pada aktivasi dan β adalah parameter pembelajaran yang dapat membantu jaringan untuk mempelajari pergeseran yang terjadi di aktivasi.

Adapun susunan dari model yang digunakan dapat dilihat pada Gambar 3.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 64)	1792
batch_normalization (Batch Normalization)	(None, 222, 222, 64)	256
conv2d_1 (Conv2D)	(None, 220, 220, 64)	36928
batch_normalization_1 (Batch Normalization)	(None, 220, 220, 64)	256
max_pooling2d (MaxPooling2D)	(None, 110, 110, 64)	0
conv2d_2 (Conv2D)	(None, 108, 108, 128)	73856
batch_normalization_2 (Batch Normalization)	(None, 108, 108, 128)	512
conv2d_3 (Conv2D)	(None, 106, 106, 128)	147584
batch_normalization_3 (Batch Normalization)	(None, 106, 106, 128)	512
max_pooling2d_1 (MaxPooling2D)	(None, 53, 53, 128)	0
conv2d_4 (Conv2D)	(None, 51, 51, 256)	295168
batch_normalization_4 (Batch Normalization)	(None, 51, 51, 256)	1024
conv2d_5 (Conv2D)	(None, 49, 49, 256)	590880
batch_normalization_5 (Batch Normalization)	(None, 49, 49, 256)	1024
max_pooling2d_2 (MaxPooling2D)	(None, 24, 24, 256)	0
flatten (Flatten)	(None, 147456)	0
dense (Dense)	(None, 1024)	150995968
dense_1 (Dense)	(None, 128)	131200
dense_2 (Dense)	(None, 1)	129

Total params: 152,276,289
Trainable params: 152,274,497
Non-trainable params: 1,792

Gambar 3. Rancangan Model

3. Implementasi dan Pengujian

Setelah pembuatan model selesai, model akan dikompilasi dengan menggunakan *optimizer* Adam dan fungsi *loss Binary Cross-Entropy* serta melihat kepada *metrics* akurasi. Adapun *optimizer* digunakan untuk mengurangi kesalahan serta memperbesar tingkat akurasi. Adam atau *Adaptive Moment Optimization* merupakan salah satu algoritma untuk *optimizer* ini. Adam bekerja dengan cara mengoptimalkan setiap *gradien* *oder* pertama secara komputasi dengan menggunakan memori yang rendah [12].

Adam *optimizer* mempunyai persamaan yang ditunjukkan pada Persamaan 3.

$$\theta_{t+1} = \theta_t - \alpha \frac{n}{\sqrt{vt} + e} mt \tag{3}$$

Dimana, θ merupakan parameter akan menjadi bobot atau penentu untuk perbaikan, α merupakan langkah yang akan digunakan dalam melakukan pelatihan, mt merupakan estimasi dari bias yang akan dikoreksi pada saat pelatihan, vt adalah estimasi *subgradien* yang akan dikoreksi dengan bias pada saat pelatihan.

Selain menggunakan *optimizer*, dibutuhkan *lost function* yang digunakan untuk mengubah parameter yang ada pada

model sehingga model yang dibuat lebih akurat dalam melakukan prediksi. Adapun *lost function* yang digunakan pada penelitian ini adalah *Binary Cross-Entropy*. Dimana, metode ini memberikan pengukuran yang sangat sesuai untuk masalah klasifikasi biner [13]. *Binary Cross-Entropy* memiliki persamaan yang ditunjukkan pada Persamaan 4.

$$L(y, y') = -(y \log(y') + (1 - y) \log(1 - y')) \quad (4)$$

Dimana, y adalah label sebenarnya yang bisa bernilai 0 atau 1 dan y' adalah label prediksi dari model yang merupakan model probabilitas sampel bisa masuk ke dalam kelas berbeda (positif 1).

Sebelum data diterapkan ke model, terlebih dahulu dilakukan *preprocessing* yang digunakan untuk melakukan normalisasi agar nilai seluruh data menjadi antara 0 - 1. Lalu, model akan di *training* dengan *epoch* maksimal 100 dan dilakukan optimasi *pipeline* data agar proses *training* lebih cepat dan efisien.

4. Evaluasi

Langkah terakhir setelah melakukan pengujian adalah evaluasi. Evaluasi pada model dilakukan untuk melihat seberapa baik hasil kinerja dari suatu model [12]. Terdapat 4 istilah yang digunakan dalam melakukan evaluasi [17]. Istilah pertama adalah *True Positive* (TP) yang diartikan sebagai jumlah data yang positif diklasifikasi benar oleh sistem. Istilah kedua adalah *True Negatif* (TN) yang diartikan sebagai jumlah data yang negatif diklasifikasi benar oleh sistem. Istilah ketiga adalah *True Negative* (TN) yang diartikan sebagai jumlah data negatif diklasifikasi salah oleh sistem. Terakhir, istilah keempat adalah *False Positif* (FP) yang diartikan sebagai jumlah data positif yang diklasifikasi salah oleh sistem. Adapun evaluasi yang digunakan pada penelitian ini adalah akurasi, presisi, *recall*, dan *F1-Score*. Adapun yang dimaksud dengan akurasi merupakan perbandingan dari data yang diprediksi benar dari jumlah seluruh data. Persamaan akurasi ditunjukkan dengan Persamaan 4.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Presisi merupakan perbandingan data positif yang diklasifikasi benar oleh sistem dibagi dengan total yang diklasifikasi benar oleh sistem. Persamaan presisi ditunjukkan dengan Persamaan 5.

$$Presisi = \frac{TP}{TP + FP} \quad (5)$$

Recall merupakan banyaknya total data positif yang berhasil diklasifikasi oleh sistem dibagi dengan total yang diklasifikasi benar dengan total yang diklasifikasi salah oleh sistem. Persamaan *recall* ditunjukkan dengan Persamaan 6.

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

F1-Score merupakan gabungan dari presisi dan *recall* yang dijadikan satu nilai tunggal. Persamaan *F1-Score* adalah ditunjukkan dengan Persamaan 7.

$$F1 - Score = \frac{2 \times Presisi \times Recall}{Presisi + Recall} \quad (7)$$

IV. HASIL DAN PEMBAHASAN

Citra yang sudah dikumpulkan akan dilakukan *resize* dengan ketentuan *batch size* 128, tinggi citra 224, dan lebar citra 224. Lalu, model akan dibuat sesuai dengan Gambar 3. Adapun susunan arsitektur yang dibuat adalah terdiri dari 6 *convolutional layer* dengan ukuran kernel 3x3 dan aktivasi ReLU yang dimana setelah setiap lapisan konvolusi akan diterapkan *layer Batch Normalization* dengan *max pooling* dengan ukuran *pool 2x2* setelah dua *layer* konvolusi. Lalu setelah lapisan *max pooling* terakhir, data akan di *flatten* agar dapat dijadikan *input* untuk *layer fully connected*. Terakhir, terdapat 3 *Dense layers* dengan aktivasi ReLU yang dimana lapisan pertama memiliki 1024 *neuron*, lapisan kedua memiliki 128 *neuron*, dan lapisan terakhir memiliki 1 *neuron* dengan aktivasi *sigmoid* sebagai output kelas prediksi dari model. Kemudian, model akan di *training* untuk menghasilkan nilai akurasi dan nilai *loss*.

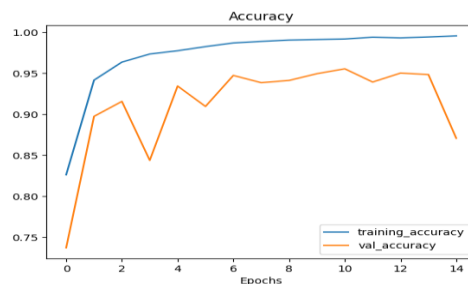
Dalam proses pembuatan model tersebut, dibutuhkan *code editor* yang dapat memproses dengan baik. Penelitian ini menggunakan *code editor* yang bersifat *open source* yaitu Kaggle. Kaggle dipilih karena dapat melakukan *training* model tanpa harus menunggu, bahkan dapat dilakukan saat perangkat yang digunakan dalam keadaan tidak digunakan. Berikut merupakan hasil dari pengujian dan evaluasi yang telah dilakukan dalam penelitian ini.

1. Hasil Pengujian

Hasil pengujian dilakukan untuk mengukur kinerja model yang telah dibuat dengan melihat dari pengukuran akurasi serta *loss*. Pengukuran tingkat akurasi dan *loss* ini digunakan untuk melihat tingkat keberhasilan model dalam melakukan klasifikasi. Pengukuran tingkat akurasi dan *loss* akan digambarkan melalui visualisasi hasil akurasi dan *loss* terhadap *epoch* atau iterasi sebanyak 15. Berikut merupakan penjelasan hasil untuk tingkat akurasi dan *loss*.

1.1 Hasil Akurasi Model

Berikut merupakan visualisasi untuk hasil pengukuran akurasi model pada saat *training* terhadap *epoch*.

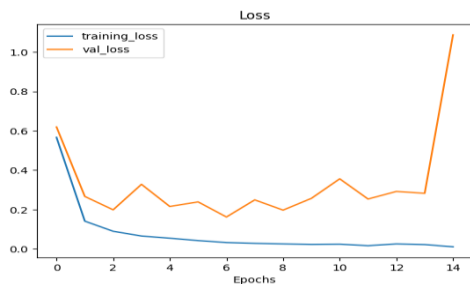


Gambar 4. Visualisasi Hasil Akurasi Model

Sesuai dengan Gambar 4, tingkat akurasi diukur untuk data *training* dan data validasi. Untuk data *training* terus mengalami peningkatan di setiap iterasinya dengan peningkatan yang hampir mendekati 100%. Sementara, untuk data validasi, akurasi yang didapatkan tidak konsisten naik ataupun turun. Akan tetapi, hampir setiap iterasi akurasi di data validasi mendekati 95%. Oleh karena itu, dapat disimpulkan model yang dibuat sudah baik dalam melakukan klasifikasi.

1.2 Hasil Loss Model

Berikut merupakan visualisasi untuk hasil pengukuran *loss* model dengan *epoch*.



Gambar 5. Visualisasi Hasil Loss Model

Sesuai dengan Gambar 5, tingkat *loss* diukur untuk data *training* dan data validasi. Untuk data *training* terus mengalami penurunan di setiap iterasinya dengan penurunan yang cukup signifikan hingga mendekati 0. Sementara, untuk data validasi, *loss* yang didapatkan tidak konsisten naik ataupun turun. Akan tetapi, hampir setiap iterasi *loss* di data validasi mendekati 0,4.

1.3 Hasil Pengujian Akurasi dan Loss Model

Berikut merupakan tabel hasil pengukuran akurasi dan *loss* model dengan *epoch* dengan iterasi sebanyak 15.

Tabel 1. Hasil Pengujian Akurasi dan Loss Model

Epoch	Akurasi	Loss	Val_Akurasi	Val_Loss
1	0,8260	0,5662	0,7369	0,6191
2	0,9415	0,1408	0,8973	0,2663
3	0,9636	0,0895	0,9156	0,1985
4	0,9734	0,0653	0,8436	0,3279
5	0,9775	0,0543	0,9343	0,2154
6	0,9825	0,0421	0,9095	0,2387
7	0,9870	0,0323	0,9473	0,1620
8	0,9888	0,0280	0,9384	0,2490
9	0,9905	0,0254	0,9412	0,1968
10	0,9911	0,0228	0,9494	0,2567
11	0,9918	0,0239	0,9554	0,3559
12	0,9940	0,0161	0,9392	0,2533
13	0,9931	0,0254	0,9502	0,2917
14	0,9942	0,0218	0,9484	0,2823
15	0,9956	0,0106	0,8706	1,0871

Sesuai dengan Tabel 1, tingkat akurasi yang dihasilkan model untuk data *training* terus mengalami peningkatan

dengan nilai akhir 0,9956 atau 99,56%. Lalu, untuk tingkat *loss* yang dihasilkan model untuk data *training* bersifat tidak konsisten dengan nilai akhir 0,0106 atau 1,06%. Untuk tingkat akurasi yang dihasilkan model untuk data validasi bersifat tidak konsisten dengan nilai akhir 0,8706 atau 87,06%. Untuk tingkat *loss* yang dihasilkan model untuk data validasi bersifat tidak konsisten dengan nilai akhir 1,0871 atau 10,87%.

2. Evaluasi

Dalam melakukan evaluasi, pada penelitian ini akan digunakan *Confusion Matrix*. *Confusion Matrix* merupakan salah satu metode untuk mempresentasikan hasil evaluasi dalam pembuatan suatu model [13]. Berikut merupakan hasil evaluasi yang telah dilakukan.

	precision	recall	f1-score	support
0	0.91	0.91	0.91	5492
1	0.91	0.90	0.91	5413
accuracy			0.91	10905
macro avg	0.91	0.91	0.91	10905
weighted avg	0.91	0.91	0.91	10905

Gambar 6. Classification Report

Dari Gambar 6, dapat dilihat bahwa untuk citra dengan kategori *Fake* (0) mendapatkan nilai *precision* sebesar 91%, *recall* sebesar 91%, dan *F1-Score* sebesar 91%. Sementara untuk citra dengan kategori *Real* (1) mendapatkan nilai *precision* sebesar 91%, *recall* sebesar 90%, dan *F1-Score* sebesar 91%. Baik kategori *Fake* maupun kategori *Real* memiliki nilai yang hampir mirip dikarenakan citra yang dimiliki hampir memiliki kemiripan antara masing-masing kategori. Akan tetapi, pembuatan model yang mendapatkan tingkat akurasi sebesar 91% dinyatakan berhasil dalam melakukan identifikasi antara wajah yang dibuat oleh *deepfake* dengan wajah asli.

V. KESIMPULAN

Kesimpulan yang didapatkan dari penelitian ini adalah pembuatan model menggunakan CNN dengan arsitektur 6 *convolutional layer* dengan tambahan *layer Batch Normalization* setelah setiap *layer* konvolusi dan *max pooling layer* setelah dua buah *layer* konvolusi dinyatakan berhasil dalam melakukan identifikasi antara wajah asli dengan wajah buatan *deepfake* dengan tingkat akurasi sebesar 91%. Disarankan untuk penelitian selanjutnya dapat menggunakan metode serta arsitektur yang lebih beragam agar mendapatkan tingkat akurasi yang lebih baik lagi serta menambahkan dataset karena aplikasi *deepfake* pada saat ini tidak terbatas hanya pada wajah saja.

REFERENSI

[1] R. A. S. Rahayu and H. Santoso, "Analisis Gambar Wajah Palsu: Mendeteksi Keaslian Gambar Yang Dimanipulasi Menggunakan Metode Variational Autoencoder Dan Forensics Deep Neural Network,"

- SIBATIK JOURNAL: Jurnal Ilmiah Bidang Sosial, Ekonomi, Budaya, Teknologi, dan Pendidikan*, vol. 2, no. 9, pp. 2701-2726, 2023.
- [2] I. D. Kasita, "Deepfake Pornografi: Tren Kekerasan Gender Berbasis Online (KGB0) Di Era Pandemi Covid-19," *Jurnal Wanita Dan Keluarga*, vol. 3, no. 1, pp. 16-26, 2022.
- [3] D. Marcella, Y. Yohannes and S. Davella, "Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network Dengan Arsitektur VGG-19," *Jurnal Algoritme*, vol. 3, no. 1, pp. 60-70, 2022.
- [4] N. Nurkhasanah and M. Murinto, "Klasifikasi Penyakit Kulit Wajah Menggunakan Metode Convolutional Neural Network," *Sainteks*, vol. 18, no. 2, pp. 183-190, 2022.
- [5] F. M. Qotrunnada and P. H. Utomo, "Metode Convolutional Neural Network untuk Klasifikasi Wajah Bermasker," *PRISMA, Prosiding Seminar Nasional Matematika*, vol. 5, pp. 799-807, 2022.
- [6] N. Dewi and F. Ismawan, "Implementasi Deep Learning Menggunakan Cnn Untuk Sistem Pengenalan Wajah," *Faktor Exacta*, vol. 14, no. 1, pp. 34-43, 2021.
- [7] A. Zarkasi, M. N. Athalaza, H. P. Satria, A. Primanita, S. Sutarno, A. Abdurahman and Y. Sazaki, "Klasifikasi Gambar Bergerak Pada Pengenalan Wajah Menggunakan Metode Convolutional Neural Network (Cnn)," *JSI: Jurnal Sistem Informasi (E-Journal)*, vol. 14, no. 2, 2022.
- [8] D. H. A. Sari, S. Sofia and N. K. C. Pratiwi, "Klasifikasi Jenis Kulit Wajah Menggunakan Modifikasi Convolutional Neural Network (CNN)," *eProceedings of Engineering*, vol. 9, no. 6, 2023.
- [9] C. N. Ihsan, "Klasifikasi Data Radar Menggunakan Algoritma Convolutional Neural Network (CNN)," *DoubleClick: Journal of Computer and Information Technology*, vol. 4, no. 2, pp. 115-121, 2021.
- [10] F. S. Putra, K. Kusriani and M. P. Kurniawan, "Deteksi Otomatis Jerawat Wajah Menggunakan Metode Convolutional Neural Network (CNN)," *Journal of Information Technology*, vol. 1, no. 2, pp. 30-34, 2021.
- [11] R. Indraswari, W. Herulambang and R. Rokhana, "Deteksi Penyakit Mata Pada Citra Fundus Menggunakan Convolutional Neural Network (CNN)," *Techno.com*, vol. 21, no. 2, pp. 378-389, 2022.
- [12] M. Karki, "deepfake and real images | Kaggle," [Online]. Available: <https://www.kaggle.com/datasets/manjilkarki/deepfake-and-real-images>. [Accessed 29 September 2023].
- [13] M. A. A. Fawwaz, K. N. Ramadhani and F. Sthevanie, "Klasifikasi Ras pada Kucing menggunakan Algoritma Convolutional Neural," *e-Proceeding of Engineering*, vol. 8, no. 1, pp. 715-730, 2021.
- [14] P. N. Zakiya, L. Novamizanti and S. Rizal, "Klasifikasi Patologi Makula Retina Melalui Citra Oct Menggunakan Convolutional Neural Network Dengan Arsitektur Mobilenet," *eProceedings of Engineering*, vol. 5, no. 8, pp. 5072-5082, 2021.
- [15] S. Wirapati and L. G. Astuti, "Sistem Pakar Untuk Membantu Diagnosis Diabetes Menggunakan Machine Learning Dengan Algoritma Jaringan Saraf Tiruan," *Jurnal Elektronik Ilmu Komputer Udayana*, vol. 11, no. 4, 2022.
- [16] A. A. Kurniawan and M. Mustikasari, "Implementasi Deep Learning Menggunakan Metode CNN dan LSTM untuk Menentukan Berita Palsu dalam Bahasa Indonesia," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 4, pp. 544-552, 2021.
- [17] B. P. Pratiwi, A. S. Handayani and Sarjana, "Pengukuran Kinerja Sistem Kualitas Udara Dengan Teknologi Wsn Menggunakan Confusion Matrix," *Jurnal Informatika Upgris*, vol. 6, no. 2, pp. 66-75, 2020.
- [18] A. Ridhovan and A. Suharso, "Penerapan Metode Residual Network (Resnet) Dalam Klasifikasi Penyakit Pada Daun Gandum," *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, vol. 7, no. 1, pp. 58-65, 2022.
- [19] T. Tinaliah and T. Elizabeth, "Penerapan Convolutional Neural Network Untuk Klasifikasi Citra Ekspresi Wajah Manusia Pada MMA Facial Expression Dataset," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 8, no. 4, pp. 2051-2059, 2021.
- [20] A. I. Pradana, "Deteksi Ketepatan Penggunaan Masker Wajah dengan Algoritma CNN dan Haar Cascade," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 9, no. 3, pp. 2305-2316, 2022.
- [21] S. Ioffe, and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448-456 2015.