PENERAPAN *LIVENESS* DETECTION DENGAN METODE NONLINEAR DIFFUSION DAN CONVOLUTIONAL NEURAL NETWORK

Alvin Christ Ardiansyah[1], Hendric Yulian[2], Simon[3], Gunawan[4], Sunaryo Winardi[5]
[1,2,3,4,5] Teknik Informatika, Universitas Mikroskil, Medan, Indonesia
[1]alvinardiansyah2002@gmail.com, [2]hendricyulian37@gmail.com, [3]yongxinmon@gmail.com,
[4]gunawan@mikroskil.ac.id, [5]sunaryo.winardi@mikroskil.ac.id

ABSTRACT

With the advancement of increasingly affordable camera technology and the ease of image capture processes, facial recognition has become the biometric authentication method most vulnerable to spoofing attacks. This vulnerability significantly undermines the data integrity and reliability of facial recognition-based attendance systems. Therefore, a liveness detection method is required to help mitigate these spoofing attacks. This study aims to enhance the reliability of existing attendance systems in the teacher and staff attendance application developed for Universitas Mikroskil. In this research, the reliability of the liveness detection system will be tested using a nonlinear diffusion algorithm and Convolutional Neural Network (CNN) against spoofing attacks, employing a confusion matrix method. Based on the conducted tests, the designed liveness detection system achieved an accuracy rate of 87.76%.

***Keywords*: liveness detection, nonlinear diffusion, spoofing, facial recognition**

I. PENDAHULUAN

Spoofing pada pengenalan wajah adalah suatu tindakan untuk mendapatkan akses ilegal ke dalam sistem dengan menggunakan citra wajah tiruan dari pengguna yang sah seperti cetak foto wajah pengguna, foto digital yang ditampilkan melalui layar perangkat elektronik seperti handphone, atau melalui pemutaran ulang video [1]. *Spoofing* melalui cetak foto wajah pengguna adalah metode yang paling mudah dikarenakan foto wajah pengguna biasanya tersedia secara publik dan mudah didapatkan melalui dunia maya seperti melalui media sosial. Cetak foto wajah pengguna juga sering dipakai karena cetak foto wajah dapat diputar, digeser dan ditekuk untuk mencurangi sistem pengenalan wajah [2].

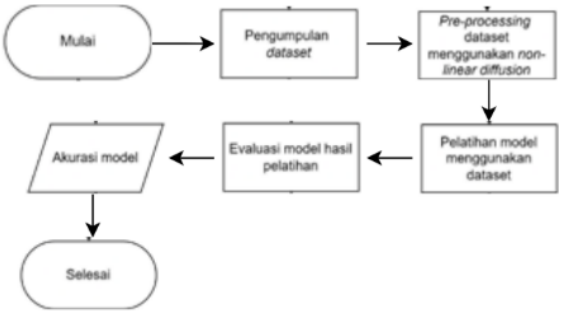
Oleh karena itu diperlukan pengembangan untuk dapat memastikan bahwa data yang didapatkan dari pengguna adalah data yang sebenar-benarnya dan mencegah *spoof*ing. Penerapan *liveness detection* pun hadir sebagai solusi dari permasalahan *spoofing* yang ada [3-6]. Terdapat beberapa metode yang sudah diajukan untuk menyelesaikan masalah serangan *spoofing* untuk deteksi *liveness* pada citra wajah. Liu et al. [7] mengajukan ekstraksi pola biner lokal yang di-*enhanced* pada peta fitur wajah sebagai fitur klasifikasi. Fitur-fitur ini kemudian dimasukkan ke dalam SVM (*Support Vector Machine*) dapat mengidentifikasi apakah citra wajah tersebut asli atau *spoof*. Das et al. [8] mengajukan pendekatan berdasarkan analisis frekuensi dan tekstur untuk membedakan antara wajah asli dan wajah palsu. Analisis frekuensi dilakukan dengan mentransformasikan citra ke dalam domain frekuensi menggunakan *Fourier Transform*, kemudian menghitung deskriptor frekuensi untuk mengetahui perubahan temporal pada wajah. Analisis tekstur dilakukan menggunakan LBP (*Local Binary Pattern*), dan vektor fitur yang dihasilkan dimasukkan ke pengklasifikasi SVM. Kim et al. [9], mengajukan perhitungan kecepatan difusi berdasarkan perbedaan sifat permukaan antara wajah asli dan wajah *spoof* dan ekstraksi fitur anti-*spoofing* berdasarkan pola lokal kecepatan difusi. Fitur-fitur ini kemudian dimasukkan ke dalam SVM untuk klasifikasi. Peneliti Ranjana Koshky dan Ausif Mahmood [10] menyelesaikan permasalahan *spoofing* pada sistem presensi digital berbasis pengenalan wajah dengan mengusulkan penggunaan non-linear diffusion yang didasarkan pada skema *Additive Operator Splitting (*AOS) dan *Tridiagonal Matrix Algorithm* (TDMA) [11] terhadap gambar yang diambil untuk memperjelas tepi dan mempertahankan batasan kontur pada gambar yang kemudian menghasilkan diffused image. *Nonlinear diffusion* membantu membedakan citra *spoof* dengan citra asli dengan melakukan difusi secara cepat, dengan demikian, tepian (edges) yang diperoleh dari permukaan yang datar akan memudar, sedangkan tepian dari citra asli akan bertahan [12]. *Diffused image* ini kemudian dimasukkan ke dalam beberapa model CNN seperti CNN-5, ResNet50, dan Inception v4 untuk mengekstrak fitur-fitur yang kemudian digunakan untuk mengklasifikasikan apakah gambar tersebut asli atau palsu.

Deteksi serangan *spoofing* pada sistem pengenalan wajah menjadi sangat penting karena ancaman keamanan yang ditimbulkan dapat merusak integritas sistem, terutama dalam aplikasi yang melibatkan otentikasi pengguna, seperti akses ke perangkat pribadi, sistem presensi, atau bahkan transaksi finansial. Keberhasilan *spoofing* dapat membuka akses ilegal bagi pihak yang tidak berwenang dan menyebabkan kerugian material maupun reputasi bagi individu atau organisasi yang terkena dampak. Oleh karena itu, pengembangan metode untuk mendeteksi *liveness* , yaitu kemampuan sistem untuk membedakan antara wajah asli dan tiruan, menjadi krusial dalam memastikan keamanan dan keandalan sistem pengenalan wajah. Peningkatan keamanan ini menjadi semakin relevan mengingat penggunaan teknologi pengenalan wajah yang terus berkembang dalam berbagai aspek kehidupan sehari-hari, seperti *smartphone*, keamanan perbankan, dan kontrol akses pada tempat-tempat sensitif [13].

Sejalan dengan urgensi tersebut, tujuan utama dari penelitian ini adalah untuk mengembangkan metode yang lebih efektif dalam mendeteksi serangan *spoofing* pada sistem pengenalan wajah, dengan memanfaatkan pendekatan berbasis analisis tekstur dan frekuensi, serta teknik *deep learning*. Kontribusi signifikan dari penelitian ini adalah pengusulan dan evaluasi model deteksi *liveness* berbasis CNN yang mampu membedakan citra wajah asli dan *spoof* dengan akurasi tinggi. Selain itu, penelitian ini mengintegrasikan teknik non-linear diffusion untuk meningkatkan kemampuan sistem dalam mengenali perbedaan antara wajah asli dan tiruan. Dengan pendekatan ini, penelitian diharapkan dapat menawarkan solusi yang dapat diimplementasikan pada sistem otentikasi berbasis wajah secara real-time, sehingga memperkuat perlindungan terhadap serangan *spoofing* dalam berbagai aplikasi yang semakin banyak diadopsi oleh masyarakat.

II. METODE PENELITIAN

Penelitian ini bertujuan untuk membangun sebuah model Inception v4 yang mampu digunakan untuk mendeteksi gambar wajah hasil *spoofing*. Adapun ringkasan proses pembangunan model dapat dilihat pada Gambar 1.



Gambar 1. Proses pembangunan model *Inception* v4

A. Pengumpulan Dataset

Penelitian ini menggunakan 4 jenis dataset di dalam proses pembangunan model Inception v4 yaitu:

1. Dataset MSU MFSD (Michigan State University Mobile Face *Spoofing* Database) [14] berisi 280 rekaman video wajah asli dan wajah *spoof* dengan jumlah partisipan sebanyak 35 orang. Dua jenis kamera dengan resolusi berbeda (720×480 dan 640×480) digunakan untuk merekam video dari 35 orang. Untuk wajah asli, setiap individu memiliki dua rekaman video yang diambil masing-masing dengan kamera laptop dan smartphone android. Untuk wajah *spoof*, dua jenis kamera, kamera iPhone dan Canon digunakan untuk merekam video definisi tinggi pada setiap subjek. Video yang diambil dengan kamera Canon kemudian diputar ulang di layar iPad Air untuk menghasilkan serangan replay HD, sementara video yang direkam oleh ponsel iPhone diputar ulang untuk menghasilkan serangan replay seluler. Serangan foto dilakukan dengan mencetak foto 35 subjek pada kertas A3 menggunakan printer berwarna. Rekaman video terhadap 35 orang dibagi menjadi kumpulan data training (15 subjek dengan 120 video) dan testing (40 subjek dengan 160 video).
2. SiW (*Spoofing in the Wild*) Database [15] berisi video *live* dan *spoof* dari 165 subjek. Untuk setiap subjek terdapat 8 video *live* dan hampir 20 video *spoof*. Setiap video direkam dalam *framerate* 30 FPS dengan resolusi 1080P HD. Video *spoof* dibuat dengan cetakan kertas citra wajah dan juga replay.
3. LCC FASD (*Large Crowdcollected Face Anti Spoofing Dataset*) [16] terdiri dari 3 subset: training, develop, dan evaluation. Totalnya ada 1.942 gambar asli dan 16.885 gambar *spoof* dikumpulkan dari Youtube, Amazon Mechanical Turk (https://www.mturk.com/) dan Yande Toloka (https://toloka.yandex.com/).
4. Dataset lapangan yang terdiri dari lima orang dengan masing-masing sepuluh foto wajah, yang kemudian foto wajah ini difoto ulang menggunakan perangkat tab Xiaomi Pad 6.

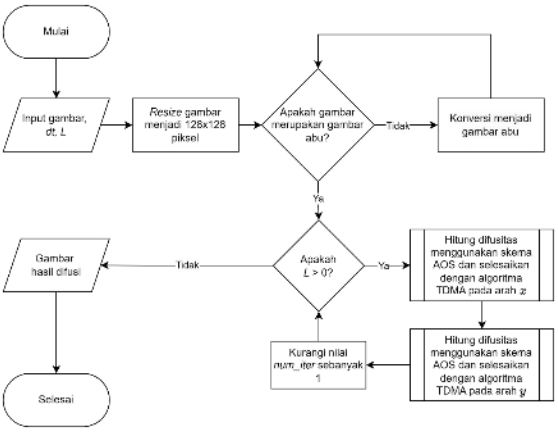
Dataset yang telah dikumpulkan akan dipisah ke dalam 3 kategori yaitu dataset *training* sebesar 70%, dataset testing sebesar 15% dan dataset *validation* sebesar 15%. Khusus untuk dataset lapangan hanya akan dijadikan dataset testing sebesar 100%.

Tabel 1. Tabel Link Dataset

Nama Dataset	**Persentase**	**Contoh**	**Link**
MSU MFSD	*Training* 70 % *Testing* 15 % *Validation* 15 %	Lampiran 1	https://drive.google.com/drive/folders/1nJCPdJ7R67xOiklF1omkfz4yHeJwhQsz
SiW	*Training* 70 % *Testing* 15 % *Validation* 15 %	Lampiran 2	https://www.kaggle.com/datasets/zyy0328/siw-image/data
LCC FASD	*Training* 70 % *Testing* 15 % *Validation* 15 %	Lampiran 3	https://www.kaggle.com/datasets/faber24/lcc-fasd
Lapangan	*Testing* 100%	Lampiran 4	Lapangan

B. Pre-Processing Dataset Menggunakan Non-Linear Diffusion

Pada tahap *pre-processing* dengan *non-linear diffusion*, gambar wajah terlebih dahulu di-resize ke dalam ukuran 128×128 piksel, kemudian gambar wajah dikonversi menjadi gambar abu. Lalu untuk setiap nilai perulangan num_iter, nilai difusitas dihitung pada arah x dan juga arah y. Pada proses perhitungan nilai difusitas, digunakan skema AOS (*Additive Operator Splitting*) dan juga algoritma TDMA (*Tridiagonal Matrix Algorithm*), lalu didapatkan gambar wajah hasil difusi. Adapun ringkasan proses *pre-processing* dengan *non-linear diffusion* dapat dilihat pada Gambar 2.



Gambar 2. Proses *pre-processing* dengan *non-linear diffusion*

Selama proses difusi, filter difusi *non-linear* akan mendeteksi tepi dan akan mempertahankan lokasi atau *region* dengan skema yang eksplisit. Skema eksplisit adalah pendekatan komputasi yang digunakan untuk mengatur proses difusi. J. Weickert [17] mengajukan sebuah skema semi implisit yaitu *Additive Operator Splitting* (AOS) yang lebih *efficient* dan dapat diandalkan. Skema ini memungkinkan difusi yang cepat, memperhalus tepian gambar citra palsu, dan mempertahankan tepian dalam citra asli. Informasi citra di dalam objek akan diblur, sedangkan informasi gambar di sepanjang tepinya akan dibiarkan utuh [3]. Skema AOS dapat dilihat pada persamaan di bawah ini.

$$(I_k)^{t+1} = \sum_{l=1}^{m} (mId - \tau m^2 A_l)^{-1} I_k^t \qquad (1)$$

Di mana *m* adalah jumlah dimensi, *k* adalah *channel*, *Id* adalah matriks identitas, A_l adalah difusi, dan τ adalah *time steps.* Untuk keperluan penelitian ini, variabel *m = 2* karena menggunakan citra 2D, dan persamaan matematika menjadi seperti berikut:

$$(I_k)^{t+1} = (2Id - 4\tau A_1) I_k^t + (2Id - 4\tau A_2)^{-1} I_k^t \quad (2)$$

Di mana A_1 dan A_2 adalah difusi dalam arah horizontal dan vertikal. AOS, bersama dengan algoritma matriks *Tridiagonal Matrix Algorithm* (TDMA), dapat digunakan untuk menyelesaikan persamaan difusi bernilai skalar *non-linear* secara efisien [11]. TDMA adalah bentuk eliminasi *Gaussian* yang disederhanakan, berguna untuk menyelesaikan sistem persamaan *tridiagonal*. Pada skema AOS, TDMA mampu menyelesaikan persamaan invers matriks $(2Id - 4\tau A_1)^{-1} I_k^t$ dengan efisien. Persamaan (2) dapat disederhanakan menjadi

$$(I_k)^{t+1} = {A_1}^{-1} d + {A_2}^{-1} d \qquad (3)$$

Dengan A_1 mewakili $2Id - 4\tau A_1$, d mewakili I_k^t. Biaya komputasi yang diperlukan untuk menghitung invers matriks A bisa menjadi sangat mahal jika ukuran matriks A besar, sehingga diperlukan suatu cara untuk mengoptimisasi perhitungan ini. Persamaan invers matriks dapat direpresentasikan sebagai berikut [11]

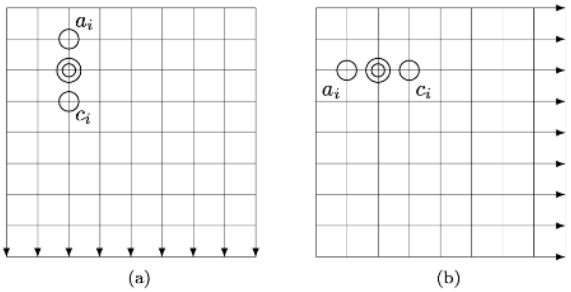
$$x = {A_1}^{-1} d$$
$$Ax = d \qquad (4)$$

Sehingga persamaan ini dapat diselesaikan dengan *Tridiagonal Matrix Algorithm* atau *Thomas Algorithm* [11]. Persamaan di atas dapat divisualisasi seperti Gambar 3.

$$\underbrace{\begin{bmatrix} b_1 & c_1 & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & 0 \\ 0 & a_3 & b_3 & \ddots & 0 \\ 0 & 0 & \ddots & \ddots & c_{N-1} \\ 0 & 0 & 0 & a_N & b_N \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_N \end{bmatrix}}_{d}$$

Gambar 3. *Matriks Tridiagonal* [11]

Kemudian akan dilakukan iterasi sebanyak panjang matriks *d*, dengan pemilihan koefisien *a* dan *c* seperti pada Gambar 4 (a) dan 4 (b)



Gambar 4. (a) iterasi *column wise*, (b) iterasi *row wise* [11]

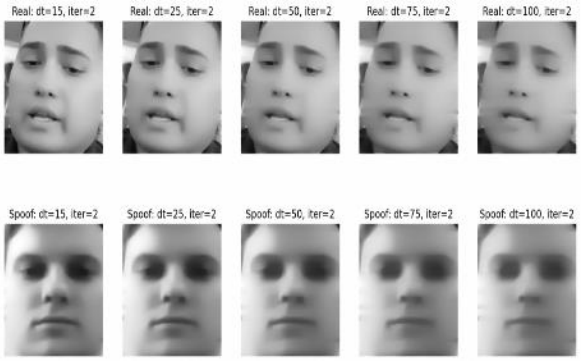
Kemudian algoritma TDMA untuk menyelesaikan persamaan (1.4). *Pseudocode* dari algoritma TDMA yang digunakan dapat dilihat pada Gambar 5.

Algorithm 1 Tridiagonal matrix algorithm. a, b, c are the column vectors for the compressed tridiagonal matrix, d is the right vector.

```
function x = TDMA(a, b, c, d)
    [rows cols dim] = size(a);

    %Modify the first-row coefficients
    c(1, :) = c(1, :) ./ b(1, :);
    d(1, :) = d(1, :) ./ b(1, :);

    %Forward pass
    for i = 2:rows-1
        temp = b(i, :) - a(i, :) .* c(i-1, :);
        c(i, :) = c(i, :) ./ temp;
        d(i, :) = (d(i, :) - a(i, :) .* d(i-1, :)) ./ temp;
    end

    %Backward pass
    x(rows, :) = d(rows, :);
    for i = rows-1:-1:1
        x(i, :) = d(i, :) - c(i, :) .* x(i + 1, :);
    end
end function
```

Gambar 5. Algoritma TDMA untuk mencari nilai x [11]

Perbandingan gambar wajah sesudah hasil *pre-processing* menggunakan *non-linear diffusion* dapat dilihat pada Gambar 6.

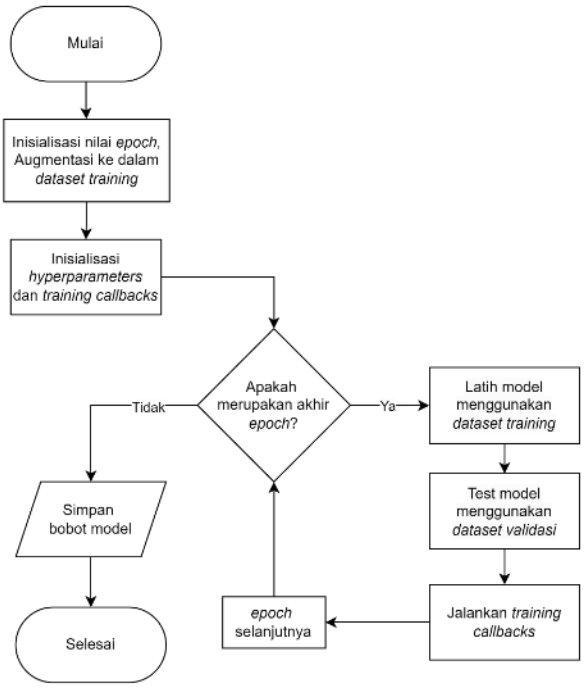


Gambar 6. Perbandingan gambar wajah sesudah *non-linear diffusion*

C. Pelatihan Model Menggunakan Dataset

Proses pelatihan model menggunakan *Inception v4* dimulai dengan inisialisasi nilai *epoch* yang digunakan untuk proses pelatihan, diikuti dengan inisialisasi *hyperparameters* dan *training callbacks. Inception* v4, yang merupakan model *deep learning* berbasis arsitektur *convolutional neural network* (CNN) yang sangat efisien, dioptimalkan untuk klasifikasi gambar yang lebih kompleks dan akurat [18].

Proses pelatihan dilakukan setiap *epoch*, dan pada setiap akhir *epoch*, *training callbacks* yang telah diinisialisasi sebelumnya akan dijalankan. Pada *epoch* terakhir, akan dilakukan penyimpanan dua jenis bobot atau *weight*, yaitu bobot dari *epoch* terakhir dan bobot terbaik berdasarkan metrik *validation*_loss. Pada tahap ini, dilakukan evaluasi terhadap kedua bobot tersebut menggunakan dataset pengujian (testing) untuk menghitung nilai *confusion matrix* masing-masing bobot. Ringkasan lengkap dari proses pelatihan model ini dapat dilihat pada Gambar 7.



Gambar 7. Proses pelatihan model

D. Evaluasi Model Hasil Pelatihan

Model yang telah dibangun akan dievaluasi menggunakan *metode confusion matrix. Confusion matrix* adalah matriks berukuran N×N di mana N adalah jumlah kelas klasifikasi. Matriks ini dapat digunakan untuk mengevaluasi kinerja model klasifikasi. Matriks akan membandingkan nilai target aktual dengan target yang diprediksi oleh model[19-21]. Terdapat empat istilah sebagai representasi hasil proses klasifikasi pada *confusion matrix*. Keempat istilah tersebut yaitu :

1. *True Positive* (TP) adalah kejadian di mana algoritma mendeteksi citra wajah yang asli dan mengklasifikasi citra wajah sebagai citra wajah yang asli.
2. *True Negative* (TN) adalah kejadian di mana algoritma mendeteksi citra wajah yang palsu dan mengklasifikasi citra wajah sebagai citra wajah yang palsu.
3. *False Positive* (FP) adalah kejadian di mana algoritma mendeteksi citra wajah yang asli dan mengklasifikasi citra wajah sebagai citra wajah yang palsu.
4. *False Negative* (FN) adalah kejadian di mana algoritma mendeteksi citra wajah yang palsu dan mengklasifikasi citra wajah sebagai citra wajah yang asli.

Confusion matrix dapat digunakan untuk menghitung berbagai *performance metrics.* Hal ini bertujuan untuk mengukur kinerja model yang telah dibuat. Beberapa *performance metrics* yang umum digunakan yakni [19], [21] :

1. *Accuracy*, yaitu seberapa akurat model dapat mengklasifikasikan dengan benar. Berikut adalah rumus untuk menghitung akurasi.

$$accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (5)$$

2. *Precision*, yaitu akurasi antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Berikut adalah rumus untuk menghitung *precision*.

$$precision = \frac{TP}{TP+FP} \quad (6)$$

3. *Recall*, yaitu keberhasilan model dalam menemukan kembali sebuah informasi. Berikut adalah rumus untuk menghitung *recall*.

$$recall = \frac{TP}{TP+FN} \quad (7)$$

4. *F1-score*, yaitu perbandingan rata-rata *precision* dan *recall* yang dibobotkan. Berikut adalah rumus untuk menghitung *F1-score.*

$$F1\ score = \frac{2 \times recall \times precision}{recall + precision} \quad (8)$$

Jika dataset memiliki jumlah data *False Negative* dan *False Positive* yang sangat mendekati (*symmetric*), maka *accuracy* bisa digunakan untuk acuan performansi algoritma, sebaliknya *F1 Score* digunakan sebagai acuan apabila jumlah data *False Negative* dan *False Positive* tidak mendekati

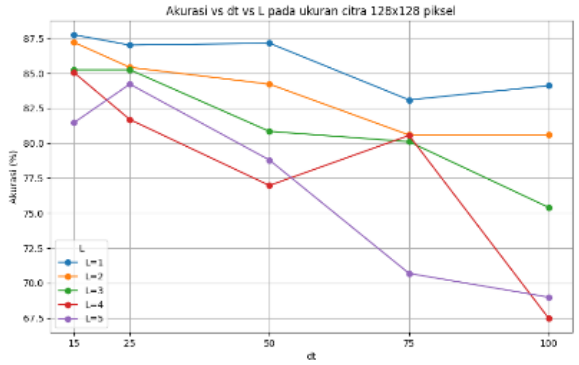
III. HASIL DAN PEMBAHASAN

Hasil pengujian model *Inception* v4 yang telah dibangun dapat dilihat pada Tabel 2.

Tabel 2. Hasil pengujian model *Inception* v4

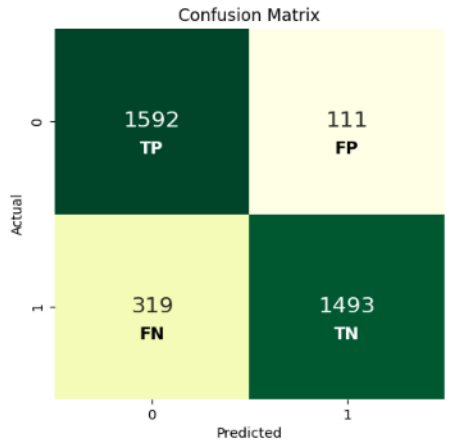
dt	L	akurasi	dt	L	akurasi	dt	L	akurasi
15	1	**87,76 %**	15	3	85,26%	15	5	81,47%
25	1	87,02%	25	3	85,26%	25	5	84,23%
50	1	87,16%	50	3	80,85%	50	5	78,80%
75	1	83,10%	75	3	80,11%	75	5	70,69%
100	1	84,12%	100	3	75,41%	100	5	68,99%
15	2	87,22%	15	4	85,06%			
25	2	85,43%	25	4	81.70%			
50	2	84,23%	50	4	76,98%			
75	2	80,59%	75	4	80,56%			
100	2	80,59%	100	4	67,48%			

Grafik perbandingan antara nilai *dt*, *L*, dan akurasi dapat dilihat pada Gambar 8.



Gambar 8. Grafik perbandingan antara nilai *dt, L* dan akurasi hasil pengujian model *Inception v4*

Berdasarkan Tabel 2, dari hasil pengujian dengan 25 kombinasi parameter yang berbeda, didapatkan kombinasi parameter terbaik dengan parameter *dt=15* dan *L=1* dengan akurasi sebesar 87,76%. Contoh hasil pengujian model dengan parameter *dt*=15 dan *L=1* menggunakan *confusion matrix* dapat dilihat pada Gambar 9.



Gambar 9. Tabel *confusion matrix* hasil pengujian model dengan parameter *dt=15* dan *L=1*

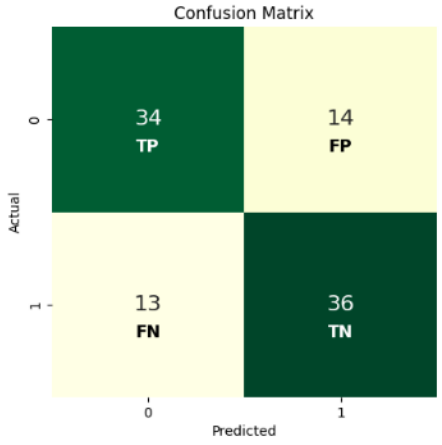
Berdasarkan *confusion matrix* hasil pengujian pada Gambar 8, maka perhitungan akurasi model dapat dihitung dengan rumus (5):

$$= \frac{1592 + 1493}{1592 + 111 + 1493 + 319}$$
$$= 87{,}76\%$$

Model *Inception* v4 dengan akurasi terbaik hasil pengujian pada Tabel 2 digunakan dalam pengujian terhadap dataset lapangan, dengan kondisi dataset lapangan tidak digunakan di dalam proses *training*, sehingga ideal digunakan sebagai acuan model dalam melakukan proses deteksi *liveness* pada data baru atau data yang belum pernah dilihat oleh model. Contoh pengujian terhadap *dataset* lapangan dapat dilihat pada Gambar 10.

Gambar 10. Contoh hasil pengujian model terhadap dataset lapangan

Adapun hasil pengujian model terhadap dataset lapangan menggunakan metode *confusion matrix* dapat dilihat pada Gambar 11.



Gambar 11. Tabel *confusion matrix* hasil pengujian model pada dataset lapangan

Berdasarkan *confusion matrix* hasil pengujian pada Gambar 10, maka perhitungan akurasi model dapat dihitung dengan rumus (5):

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$
$$= \frac{34 + 36}{34 + 14 + 36 + 13}$$
$$= 72{,}16\%$$

IV. KESIMPULAN DAN SARAN

Berdasarkan pengujian yang dilakukan terhadap model Inception v4 untuk proses deteksi liveness, dapat diambil beberapa kesimpulan sebagai berikut: Hasil pengujian menggunakan confusion matrix menunjukkan bahwa model Inception v4 berhasil melakukan generalisasi terhadap citra wajah asli dan spoof dengan baik, mencapai akurasi ≥80%. Hal ini menegaskan efektivitas model dalam membedakan antara citra yang valid dan citra palsu.

Algoritma non-linear diffusion yang diintegrasikan dengan model CNN Inception v4 terbukti efektif dalam proses deteksi liveness. Pengujian yang dilakukan dengan 25 kombinasi parameter difusi dan 3 parameter ukuran citra input menghasilkan kombinasi yang optimal. Parameter difusi dengan dt 15, iterasi (L) 1, dan ukuran citra input 128×128 piksel menghasilkan akurasi tertinggi sebesar 87,76%.

Untuk pengembangan sistem di masa mendatang, peneliti disarankan untuk menguji parameter training sebagai parameter terbuka, dengan harapan dapat meningkatkan tingkat akurasi model yang dihasilkan. Selain itu, peneliti di masa depan juga disarankan untuk mengeksplorasi dan membandingkan beberapa model CNN lainnya, sehingga dapat diidentifikasi model yang mungkin memberikan tingkat akurasi lebih tinggi dibandingkan model yang digunakan dalam penelitian ini. Dengan saran-saran tersebut, diharapkan penelitian di masa mendatang dapat mengoptimalkan metode deteksi liveness dan memberikan kontribusi yang lebih signifikan dalam meningkatkan keamanan sistem pengenalan wajah.

V. REFERENSI

[1] I. Chingovska, N. Erdogmus, A. Anjos, and S. Marcel, “Face recognition systems under spoofing attacks,” in *Face Recognition Across the Imaging Spectrum*, Springer International Publishing, 2016, pp. 165–194. doi: 10.1007/978-3-319-28501-6_8.

[2] G. Pan, Z. Wu, and L. Su, ‘Liveness Detection for Face Recognition’, *Recent Advances in Face Recognition*. InTech, Jun. 01, 2008. doi: 10.5772/6397.

[3] E. A. Raheem, S. M. S. Ahmad, and W. A. W. Adnan, “Insight on face liveness detection: A systematic literature review,” 2019, *Institute of Advanced Engineering and Science*. doi: 10.11591/ijece.v9i6.pp5165-5175.

[4] F. M. Chen, C. Wen, K. Xie, F. Q. Wen, G. Q. Sheng, and X. G. Tang, “Face liveness detection: Fusing colour texture feature and deep feature,” *IET Biom*, vol. 8, no. 6, pp. 369–377, Nov. 2019, doi: 10.1049/iet-bmt.2018.5235.

[5] T. Putra Agastiya (2023), “PENERAPAN PENGENALAN WAJAH DENGAN ALGORITME MTCNN DAN FACENET PADA APLIKASI PRESENSI GURU DAN PEGAWAI.”, Undergraduate (S-1) thesis, Universitas Mikroskil. https://repository.mikroskil.ac.id/id/eprint/3273/

[6] C. Yu, C. Yao, M. Pei, and Y. Jia, “Diffusion-based kernel matrix model for face liveness detection,” *Image Vis Comput*, vol. 89, pp. 88–94, Sep. 2019, doi: 10.1016/j.imavis.2019.06.009.

[7] X. Liu, R. Lu and W. Liu, "Face liveness detection based on enhanced local binary patterns," *2017 Chinese Automation Congress (CAC)*, Jinan, China, 2017, pp. 6301-6305, doi: 10.1109/CAC.2017.8243913.

[8] Institute of Electrical and Electronics Engineers, IEEE International Conference on Advances in Engineering and Technology Research 2014.08.01-02 Unnao, and ICAETR 2014.08.01-02 Unnao, *International Conference on Advances in Engineering and Technology Research (ICAETR)*, 2014 1-2 Aug. 2014, Unnao, Kanpur, [Uttar Pradesh], India.

[9] [9] W. Kim, S. Suh, and J. J. Han, “Face liveness detection from a single image via diffusion speed model,” *IEEE Transactions on Image Processing*, vol. 24, no. 8, pp. 2456–2465, Aug. 2015, doi: 10.1109/TIP.2015.2422574.

[10] R. Koshy and A. Mahmood, “Optimizing deep CNN architectures for face liveness detection,” *Entropy*, vol. 21, no. 4, Apr. 2019, doi: 10.3390/e21040423.

[11] J. Ralli, “PDE Based Image Diffusion and AOS,” 2012. [Online]. Available: www.jarnoralli.com

[12] A. Alotaibi and A. Mahmood, “Deep face liveness detection based on nonlinear diffusion using convolution neural network,” *Signal Image Video Process*, vol. 11, no. 4, pp. 713–720, May 2017, doi: 10.1007/s11760-016-1014-2.

[13] Sudeep, S.V.N.V.S., Venkata Kiran, S., Nandan, D., Kumar, S. (2021). An Overview of Biometrics and Face Spoofing Detection. In: Kumar, A., Mozar, S. (eds) ICCCE 2020. Lecture Notes in Electrical Engineering, vol 698. Springer, Singapore. doi : 10.1007/978-981-15-7961-5_82

[14] DiWen, Hu Han, and Anil K. Jain. Face spoof detection with image distortion analysis. IEEE Transactions on Information Forensics and Security, 10(4):746–761, 2015. doi:10.1109/TIFS.2015.2400395

[15] Y. Liu, A. Jourabloo and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision", *Proc. IEEE Comput. Vis. Pattern Recognit.*, pp. 389-398, Jun. 2018, doi 10.1109/CVPR.2018.00048

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

[17] J. Weickert, B. M. T. H. Romeny and M. A. Viergever, "Efficient and reliable schemes for nonlinear diffusion filtering," in IEEE Transactions on Image Processing, vol. 7, no. 3, pp. 398-410, March 1998, doi: 10.1109/83.661190.

[18] C. Szegedy, et al., "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, 2017, pp. 4278-4284, doi : https://doi.org/10.1609/aaai.v31i1.11231

[19] M. S. ANGGREANY, “Confusion Matrix”, Accessed: April 25, 2024. [Online]. Available:

https://socs.binus.ac.id/2020/11/01/confusion-matrix/

[20] Bhandari Aniruddha, "Understanding & Interpreting Confusion Matrices for Machine Learning,", Accessed: April 25, 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/

[21] J. Brownlee, "What is a Confusion Matrix in Machine Learning,", Accessed: April 25, 2024. [Online]. Available: https://machinelearningmastery.com/confusion-matrix-machine-learning/