

UJI PARAMETER DAN ARSITEKTUR *CONVOLUTIONAL NEURAL NETWORK* UNTUK MENDETEKSI CITRA WAJAH BERMASKER

Dewi Novita Sari^{*1}, Muh. Arif Rahman², Randy Cahya Wihandika³

^{1,2,3}Universitas Brawijaya, Malang

Email: ¹dewinovitasari197@gmail.com, ²m_arif@ub.ac.id, ³rendicahya@ub.ac.id

^{*}Penulis Korespondensi

(Naskah masuk: 09 Desember 2022, diterima untuk diterbitkan: 27 Desember 2022)

Abstrak

Deteksi citra wajah bermasker dibutuhkan pada masa pandemi COVID-19 oleh lembaga-lembaga yang terhubung langsung dengan masyarakat dikarenakan terbatasnya sumber daya manusia dalam melakukan deteksi wajah bermasker secara konvensional. Penggunaan masker dalam aktivitas sehari-hari merupakan salah satu protokol perlindungan diri dari COVID-19 yang wajib diterapkan. Citra wajah bermasker digunakan sebagai data masukan dengan proses deteksi menggunakan *Convolutional Neural Network* (CNN). Deteksi citra wajah bermasker telah banyak dilakukan dengan berbagai bentuk arsitektur model, akan tetapi tidak disertai dengan penjelasan dari pemilihan parameter yang digunakan. Pembuatan model dapat menjadi efisien jika dilakukan dengan mengetahui hubungan keterkaitan antar parameter yang diterapkan. Oleh karenanya, penelitian ini dilakukan dengan tujuan untuk mengetahui hubungan keterkaitan antar parameter dalam arsitektur model CNN. Sehingga dapat dihasilkan performa terbaik dalam mendeteksi citra wajah bermasker. Hubungan keterkaitan antar parameter yang diteliti terbatas pada ukuran *kernel* dan jumlah *kernel* karena peran aktif keduanya dalam melakukan pelatihan data. Dua ukuran *kernel* yang digunakan yaitu 3×3 dan 5×5 dengan jumlah 3 dan 6 buah. Empat arsitektur model dibangun dengan 7 *layer* penyusun menggunakan kombinasi parameter tersebut. Pelatihan model dilakukan menggunakan data citra wajah bermasker dan tidak bermasker berjumlah 3150 citra dengan 15 *epoch*, kemudian diuji menggunakan 1350 citra. Performa terbaik diperoleh dari kombinasi parameter ukuran *kernel* 5×5 berjumlah 6 buah pada setiap *convolutional layer*. Nilai *f1-score* terbaik yang diperoleh sebesar 0,95 dengan akurasi 0,95 dan nilai rata-rata *loss* 0,1692. Berdasarkan hasil tersebut, disimpulkan bahwa parameter ukuran *kernel* dan jumlah *kernel* memiliki hubungan keterkaitan dalam menghasilkan nilai performa arsitektur model CNN terbaik untuk pendeteksian citra wajah bermasker.

Kata kunci: COVID-19, CNN, wajah bermasker, deteksi citra, pengolahan citra, backpropagation.

EVALUATION OF PARAMETER AND CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE TO DETECT MASKED FACE IMAGES

Abstract

Detection of masked face images is needed during the COVID-19 pandemic by institutions directly connected to the community due to limited human resources to perform conventional masked face detection. Using masks in daily activities is one of the self-protective protocols from COVID-19 that must be implemented. Masked face images are used as input data, the detection process uses Convolutional Neural Network (CNN). Detection of masked face images has been carried out with various forms of model architecture but is not accompanied by an explanation of the selected parameters used. Modeling can be done efficiently by knowing the relationship between the applied parameters. Therefore, this study aims to know the relationship between parameters in the CNN model architecture so that the best performance can be produced in detecting masked face images. The study of the relationship between parameters is limited to the size of the kernel and the number of kernels because of their active role in the data training. The two kernel sizes used are 3×3 and 5×5 , with a total of 3 and 6 pieces. Four model architectures are built with seven layers using a combination of these parameters. The model training was carried out using masked and maskless faces of 3150 images with 15 epochs, then tested using 1350 images. The best performance is obtained from 6 pieces of 5×5 kernel size in each convolutional layer. The best *f1-score* value obtained is 0.95, with an accuracy of 0.95 and an average loss value of 0.1692. Based on these results, it is concluded that the kernel size parameter and the number of kernels have a relationship in producing the best CNN architectural performance value for masked face image detection.

Keywords: COVID-19, CNN, masked face, image detection, image processing, backpropagation.

1. PENDAHULUAN

Pandemi COVID-19 menyebabkan penderitanya mengalami gangguan saluran pernafasan ringan hingga berat, bahkan sampai dengan meninggal dunia (WHO, 2021). Di Indonesia, sebanyak lebih dari satu juta orang terkonfirmasi positif COVID-19 (WHO, 2021). Pembatasan aktivitas dengan mewajibkan masyarakat menggunakan masker dalam kegiatan sehari-hari, akses ke fasilitas publik, hingga memasuki tempat-tempat tertentu diterapkan sebagai protokol perlindungan diri dari COVID-19 (WHO, 2020). Oleh karena itu, deteksi wajah bermasker dinilai perlu dan dianggap penting sebagai partisipasi aktif penerapan teknologi dengan mempertimbangkan terbatasnya sumber daya manusia dan penerapan protokol perlindungan diri dalam menjaga jarak.

Citra digital merupakan kumpulan dari banyak piksel sebagai hasil dari tangkapan kamera. Citra digital direpresentasikan dengan matriks dua dimensi dengan jumlah piksel berhingga (Madenda, 2015). Komputer dapat mendeteksi citra digital wajah bermasker dengan melakukan pembelajaran mesin atau disebut *machine learning*. Pembelajaran mesin merujuk pada kemampuan dari mesin komputer untuk menirukan perilaku cerdas manusia dalam melakukan pembelajaran memproses data yang ada menjadi informasi (Loy, 2019).

Pembelajaran mesin yang ditujukan untuk mengolah data masukan berupa citra digital adalah *convolutional neural network* atau dapat disingkat dengan CNN (Loy, 2019). CNN dapat diandalkan dalam deteksi citra digital. Hal ini didukung dengan hasil penelitian oleh Krizhevsky (2012) yang mampu melakukan klasifikasi terhadap 1,2 juta citra *dataset* dengan *error rate* 15,3% yang lebih baik dari model lainnya dengan 26,2%. Deteksi masker pada citra wajah pernah dilakukan oleh Oumina (2021) dengan *dataset* citra berwarna menghasilkan akurasi sebesar 97,1% dan juga pernah dilakukan oleh Wihandika (2021) menggunakan *dataset* citra *grayscale* menghasilkan akurasi sebesar 98,30%.

CNN digunakan khusus mengolah citra digital karena kinerjanya yang mampu mendapatkan informasi dari beberapa kumpulan piksel sekaligus dan mampu mempelajari adanya informasi yang berkaitan antara satu piksel dengan piksel lain yang di sekitarnya. Dasar arsitektur CNN memiliki tiga *layer* utama yaitu *convolution layer* untuk mencari fitur atau informasi yang dapat diambil dari citra, *pooling layer* sebagai *layer* pembantu komputasi dengan mengeliminasi informasi yang dianggap tidak perlu karena hal ini ukuran citra yang dihasilkan *layer* ini memiliki ukuran yang lebih kecil dari citra masukan, dan yang terakhir *fully connected layer* untuk diterapkan pembobotan menggunakan bobot dalam melakukan klasifikasi terhadap masukan. CNN memiliki parameter inti yaitu *kernel* (jumlah dan

ukuran *kernel*), *bias*, dan *weight* atau bobot (Goodfellow, et al., 2016).

Backpropagation merupakan teknik yang sering digunakan untuk melakukan pembaharuan parameter yang sebelumnya diinisialisasikan secara *random* hingga kemudian didapatkan parameter terbaik untuk digunakan dalam arsitektur model. *Backpropagation* melakukan propagasi mundur untuk mencari nilai gradien dari parameter *kernel*, *bias*, dan *weight* hingga setelahnya didapatkan nilai parameter baru untuk digunakan kembali dalam proses *forward pass* atau propagasi maju untuk didapatkan nilai *loss* dan akurasi yang lebih baik. Untuk mendukung model arsitektur CNN dalam penelitian ini menggunakan *backpropagation* untuk melakukan pelatihan data secara mandiri. *Backpropagation* melakukan pembaharuan bobot atau *weight* pada model CNN dengan sendirinya.

Sebelumnya, deteksi citra wajah bermasker menggunakan CNN pernah dilakukan oleh Sidik (2021), Siradjuddin (2021), dan Pandey (2021). Akan tetapi, tidak dijelaskan hubungan keterkaitan antar parameter dalam arsitektur model yang digunakan. Sehingga, tidak diketahui secara menyeluruh dampak dari parameter yang diterapkan. Pandey dan rekan-rekan (2021) melakukan deteksi citra wajah bermasker dengan membuat perangkat lunak yang menerapkan model CNN dengan dua *convolution layer* yang menggunakan *kernel* berukuran 3×3, memiliki jumlah *kernel* 200 dan 100 buah, penelitian ini menghasilkan akurasi sebesar 96%. Sidik (2021) melakukan deteksi citra wajah bermasker dengan fokus terhadap pemilihan *learning rate* terhadap beberapa model yang dipilih dan hasil akurasi terbaik sebesar 82,46% didapatkan oleh RetinaFace. Siradjuddin dan kawan-kawan (2021) meneliti tentang deteksi citra wajah bermasker dengan menerapkan model VGG-16 ke dalam model arsitektur yang mereka buat dengan hasil presisi tertinggi sebesar 0,86 terhadap citra wajah normal dan presisi terendah 0,57 dari penggunaan masker yang salah.

Oleh karenanya, penelitian ini dilakukan dengan tujuan untuk mengetahui hubungan keterkaitan antar parameter dalam arsitektur model CNN. Sehingga dapat dihasilkan performa terbaik dalam mendeteksi citra wajah bermasker. Hubungan keterkaitan antar parameter yang diteliti terbatas pada ukuran *kernel* dan jumlah *kernel* karena peran aktif keduanya dalam melakukan pelatihan data. Dua ukuran *kernel* yang digunakan yaitu 3×3 dan 5×5 dengan jumlah 3 dan 6 buah. Dua ukuran *kernel* ini lebih sering digunakan pada penelitian sebelumnya dan penentuan jumlah *kernel* diusahakan menggunakan jumlah kecil untuk mempermudah dilakukannya analisis. Pada penelitian ini dibangun empat arsitektur model dengan 7 *layer* penyusun menggunakan kombinasi parameter jumlah dan ukuran *kernel*. Arsitektur model kemudian dilakukan pelatihan dan pengujian menggunakan data citra wajah bermasker. Nilai

keluaran yang dihasilkan untuk digunakan mengukur performa model menggunakan nilai *loss*, akurasi, *precision*, *recall*, dan *f1-score* yang kemudian dapat dilakukan penerikan kesimpulan berdasarkan nilai-nilai tersebut.

2. METODE PENELITIAN

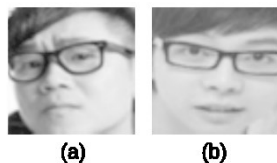
2.1. Citra Wajah Bermasker

Wajah merupakan bagian tubuh manusia yang dapat dijadikan sebagai objek deteksi unik. Wajah memiliki ciri khusus seperti mata, hidung, dan mulut yang dapat membedakan suatu objek pada citra digital dapat dikenali sebagai wajah atau bukan wajah. Sedangkan wajah bermasker adalah wajah seseorang dengan menggunakan masker sebagai bentuk perlindungan diri dari virus COVID-19 sehingga ciri khusus wajah sebagian besar tertutup.

Penelitian ini menggunakan data berupa citra digital wajah tidak masker dan wajah bermasker yang didapatkan dari Wang dan kawan-kawan (2020). Data citra ini hadir dalam berbagai ukuran, memiliki ekstensi atau tipe file citra yang tidak sama satu dengan yang lain, jenis citra yang beragam seperti citra RGB dan *grayscale*.

Untuk memperlancar penelitian ini, data yang digunakan sebanyak 4500 citra wajah dengan rincian 2564 citra wajah dan 1936 citra wajah bermasker diproses bertahap hingga dihasilkan *dataset* citra dengan ukuran 64×64 piksel, berjenis *grayscale*, dan berlabel. Untuk memberikan representasi secara jelas tentang keadaan data citra yang digunakan, dapat dilihat Gambar 1 dan 2 untuk citra wajah tanpa masker, serta Gambar 3 dan 4 untuk citra wajah bermasker.

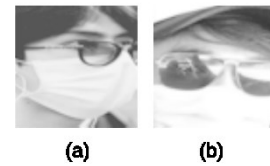
Data citra dibagi menjadi citra data latih dengan jumlah 3150 dan citra data uji dengan jumlah 1350. Citra diubah dalam bentuk *NumPy array file* (.npy) untuk memudahkan dalam memuat data ke dalam arsitektur model untuk diproses. *NumPy file* menyimpan data dalam bentuk *array* multidimensi. *Dataset* citra inilah yang menjadi *input* untuk model arsitektur CNN yang dibangun pada penelitian ini.



Gambar 1. Citra wajah berkacamata



Gambar 2. Citra wajah berbagai sisi



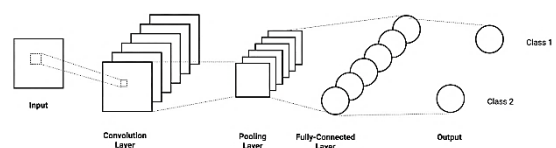
Gambar 3. Citra wajah bermasker dan berkacamata



Gambar 4. Citra wajah bermasker berbagai sisi

2.2. Convolutional Neural Network

Convolutional Neural Network (CNN) digunakan khusus untuk *dataset* berupa citra digital karena kemampuannya dalam mempertahankan informasi spasial, di mana suatu piksel memiliki keterkaitan dengan piksel di sekitarnya merepresentasikan informasi tertentu, di mana informasi ini tidak bisa didapatkan jika suatu piksel hanya merepresentasikan nilai piksel itu sendiri. Oleh karenanya, CNN mampu untuk mengenali fitur khusus suatu objek pada sebuah citra. Pada CNN terdapat istilah *parameter sharing* atau berbagi parameter yang mengacu pada penggunaan parameter yang sama untuk lebih dari satu fungsi dalam sebuah model (Goodfellow, et al., 2016). Arsitektur CNN memiliki beberapa *layer* inti penyusun yaitu *convolution layer*, *pooling layer*, dan *fully connected layer*. Model arsitektur CNN sederhana dapat dilihat pada Gambar 5.



Gambar 5. Arsitektur sederhana CNN

Convolution layer (CL) merupakan *layer* untuk melakukan operasi perkalian piksel citra dengan *kernel*, kemudian diikuti penjumlahan dengan *bias*, dan ditambahkan dengan seluruh piksel dengan cara perhitungan yang sama. Parameter dalam CL berupa ukuran *kernel*, jumlah *kernel*, dan bias. *Kernel* dapat diartikan sebagai *filter* atau pengekstraksi ciri untuk citra *input* yang merupakan matriks dua dimensi dengan ukuran 3×3 dan 5×5 penggunaannya bergantung pada kebutuhan dan ukuran citra masukan. Operasi konvolusi dan fungsi aktivasi yang digunakan dalam CL dapat dilihat pada Persamaan (1) dan (2).

Persamaan (1) merupakan persamaan untuk operasi konvolusi dengan r, c adalah indeks *kernel*. i, j merupakan indeks citra *input*. n, m sebagai banyak

kernel. PL citra hasil *pooling layer*. Persamaan (2) adalah persamaan aktivasi *ReLU* dengan Summa_{nij}^1 merupakan hasil operasi konvolusi.

$$S_{nij}^1 = \sum_{r=0}^{r-1} \sum_{c=0}^{c-1} \text{Input}_{r+i,c+j} \cdot \text{kernel}_{nrc}^1 + \text{bias}_n^1 \quad (1)$$

$$\text{ReLU}(S_{nij}^1) = \begin{cases} S_{nij}^1, & \text{jika } S_{nij}^1 > 0 \\ 0, & \text{jika } S_{nij}^1 \leq 0 \end{cases} \quad (2)$$

Pooling layer (PL) merupakan *layer* yang digunakan untuk mengurangi dimensi *output* dari CL dengan tujuan mengurangi komputasi yang dilakukan dalam model arsitektur CNN. Persamaan yang digunakan dapat dilihat pada Persamaan (3). Pada Persamaan (3), dilakukan operasi *max pooling* dengan menggunakan *kernel* 2×2 untuk didapatkan nilai maksimum. CL_{nij} merupakan *output* hasil CL yang telah diaktivasi dengan fungsi *ReLU*.

$$\text{PL}_{nrc} = \max(\text{CL}_{nij}), i = 2r, 2r + 1, j = 2c, 2c + 1 \quad (3)$$

Pada Persamaan (4), dilakukan untuk menghitung ukuran *output* yang dihasilkan tiap *layer*. n_A merupakan lebar piksel *input*, n_K sebagai lebar piksel *kernel*, p adalah *padding*, dan s sebagai *stride*.

$$n_B = \frac{n_A - n_K + 2p}{s} + 1 \quad (4)$$

Fully-Connected Layer atau FCL merupakan *layer* dengan satu dimensi yang merupakan hasil *flatten* dari *output pooling layer*. *Flatten* adalah hasil berubahan dua dimensi matrik menjadi satu dimensi tanpa menambah atau mengurangi piksel. Persamaan yang digunakan dalam FCL dapat dilihat pada Persamaan (5) dan (6).

Pada Persamaan (5), dilakukan perubahan 2 dimensi hasil *output* PL menjadi 1 dimensi. Pada Persamaan (6) diterapkan operasi pembobotan dengan weight_{ij} adalah nilai bobot baris i kolom j dan fl_j adalah hasil perubahan 2 dimensi menjadi 1 dimensi *output* PL.

$$\text{fl} = \text{flatten}(\text{PL}) \quad (5)$$

$$S_i = \sum_{j=0}^{j-1} \text{weight}_{ij} \cdot \text{fl}_j + \text{bias}_i \quad (6)$$

Setelah dilakukan pembobotan, kemudian dilakukan aktivasi *softmax* terhadap nilai *output* yang dapat dilihat pada Persamaan (7) dengan e^{S_i} merupakan hasil pemangkatan eksponensial dan e^{S_k} sebagai jumlah dari pemangkatan eksponensial.

$$\text{Output}_i = \text{softmax}(S_i) = \frac{e^{S_i}}{\sum_{k=0}^1 e^{S_k}} \quad (7)$$

2.2. Backpropagation

Backpropagation digunakan untuk mencari nilai turunan dari parameter model hingga dihasilkan

nilai optimal dari *loss function*. *Backpropagation* melakukan pembaruan parameter untuk mendapatkan nilai *loss* seminimal mungkin. Parameter yang dimaksud adalah *kernel*, *weight*, dan *bias*. Perhitungan CNN dengan *backpropagation* pernah dituliskan ke dalam artikel oleh (Zhang, 2016). Fungsi *loss* yang digunakan dapat dilihat pada Persamaan (8) dengan \ln sebagai *natural log*.

$$\text{Loss} = -\ln(\text{Output}_{\text{label}}), \text{label} \in \{0, 1\} \quad (8)$$

2.3. Confusion Matrix

Confusion matrix merupakan sebuah metode yang digunakan untuk menghitung performa dan mengevaluasi model pembelajaran mesin terhadap *output* dari data yang diujikan. *Confusion matrix* ditampilkan dalam matriks dua dimensi, satu dimensi untuk nilai sebenarnya, dan dimensi selanjutnya untuk nilai hasil prediksi yang telah dilakukan (Ting, 2011). *Confusion Matrix* ini menghasilkan nilai performa akurasi menggunakan Persamaan (9), *precision* dengan penerapan Persamaan (10), *recall* menggunakan Persamaan (11), dan *f1-score* yang didapatkan menggunakan Persamaan (12) dengan TP sebagai *True Positive*, TN yang merupakan *True Negative*, FP sebagai *False Positive*, dan FN singkatan dari *False Negative*.

$$\text{Akurasi} = \frac{(TP+TN)}{(TN+FP+FN+TP)} \quad (9)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (10)$$

$$\text{Recall} = \frac{TP}{FN+TP} \quad (11)$$

$$F1 - \text{Score} = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right) \quad (12)$$

3. PERANCANGAN MODEL

Model arsitektur CNN dibangun dengan menggunakan kombinasi parameter ukuran *kernel* dan jumlah *kernel* pada setiap CL. Ukuran *kernel* yang digunakan adalah 3×3 dan 5×5. Jumlah *kernel* yang digunakan sejumlah 3 dan 6 *kernel*. Skenario model dibutuhkan agar pengujian model tetap terfokus dan teratur. Skenario kombinasi parameter penyusunan model dapat dilihat pada Tabel 1.

Tabel 1. Skenario Model

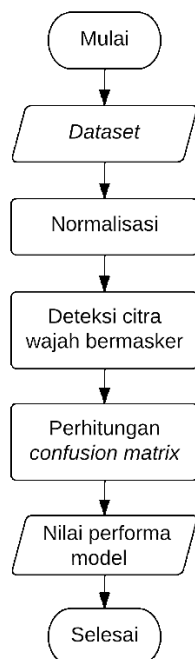
| Model | Ukuran <i>kernel</i> | | Jumlah <i>kernel</i> | |
|-------|----------------------|-------|----------------------|---|
| | 3 × 3 | 5 × 5 | 3 | 6 |
| A | ✓ | | ✓ | |
| B | ✓ | | | ✓ |
| C | | ✓ | ✓ | |
| D | | ✓ | | ✓ |

Empat arsitektur model CNN terdiri dari 7 *layer* penyusun dengan detail susunan CL1, PL1, CL2, PL2, CL3, PL3, FCL, dan diikuti dengan *backpropagation*. Pada setiap CL menggunakan *stride* 1 piksel, tanpa *padding*, dan diterapkan fungsi aktivasi ReLU. Pada PL diterapkan *window* 2×2 dengan *stride* sejumlah 2 piksel. Pada FCL jumlah bobot yang diterapkan berdasarkan jumlah piksel *output*, lalu diterapkan fungsi aktivasi *softmax*. Untuk memberikan representasi lebih jelas, dapat dilihat pada Tabel 2.

Tabel 2. Arsitektur Model

| Layer | Model | | | |
|--------|----------------|----------------|----------------|----------------|
| | A | B | C | D |
| Input | Input 64×64 | Input 64×64 | Input 64×64 | Input 64×64 |
| 1 | CL1 3×3×3 | CL1 6×3×3 | CL1 3×5×5 | CL1 6×5×5 |
| 2 | PL1 2×2 | PL1 2×2 | PL1 2×2 | PL1 2×2 |
| 3 | CL2 3×3×3×3 | CL2 6×6×3×3 | CL2 3×3×5×5 | CL2 6×6×5×5 |
| 4 | PL2 2×2 | PL2 2×2 | PL2 2×2 | PL2 2×2 |
| 5 | CL3 3×3×3×3 | CL3 6×6×3×3 | CL3 3×3×5×5 | CL3 6×6×5×5 |
| 6 | PL3 2×2 | PL3 2×2 | PL3 2×2 | PL3 2×2 |
| 7 | FCL | FCL | FCL | FCL |
| Output | Output | Output | Output | Output |

Untuk memberikan penjelasan lebih rinci dari alur proses model arsitektur CNN, representasi alur proses dapat dilihat pada Gambar 6..



Gambar 6. Alur Proses Model

Alur proses arsitektur model dapat dilihat pada Gambar 6, diawali dengan memasukkan *dataset* berupa data latih dan data uji ke dalam model. Kemudian dilakukan proses normalisasi terhadap nilai-nilai piksel citra *input* untuk mempermudah proses komputasi. Setelah itu dilakukan proses deteksi citra wajah bermasker yang di dalamnya dilakukan pelatihan dan pengujian data untuk kemudian didapatkan nilai-nilai hasil deteksi. Hingga akhirnya nilai ini dilakukan perhitungan dan dilakukan *plotting* sehingga didapatkan nilai performa model dan ditampilkan. Nilai yang dihasilkan berupa *nilai loss*, akurasi, *precision*, *recall*, dan *f1-score*.

4. HASIL DAN PEMBAHASAN

Pengujian dilakukan menggunakan empat model, yaitu A, B, C, dan D berdasarkan kombinasi parameter penyusun arsitektur model pada Tabel 1. Setiap model menggunakan data citra wajah bermasker dan tidak bermasker dengan jumlah keseluruhan 4.500 citra. Jumlah tersebut telah dibagi menjadi data citra latih dan data citra uji dengan pembagian 70% atau sebesar 3.150 untuk data citra latih dan 30% atau sebesar 1.350 untuk data citra uji. Model A hingga D menggunakan keseluruhan data yang sebelumnya telah dilakukan pengacakan susunan data. Dalam prosesnya, jumlah *epoch* yang diterapkan sebanyak 15 dengan nilai *learning rate* 0,005. Hasil pengujian berupa tabel *confusion matrix*, nilai akurasi, *precision*, *recall*, dan *f1-score*. Nilai *f1-score* menjadi krusial karena ketidakseimbangan data yang digunakan, dengan kondisi jumlah data citra wajah bermasker lebih kecil. Dengan tetap mempertimbangkan nilai akurasi yang merupakan nilai penting. Sehingga, kumpulan nilai tersebut digunakan untuk menentukan performa dari masing-masing model.

Model A dilakukan menggunakan *kernel* 3×3 sejumlah 3 buah dengan hasil yang dapat dilihat pada Tabel 3 dengan W sebagai citra wajah dan WB sebagai citra Wajah Bermasker.

Tabel 3. *Confusion Matrix* Model A

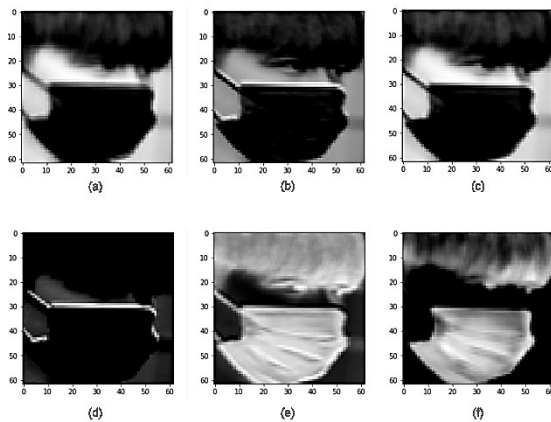
| Model A | | Hasil Deteksi | |
|------------------|----|---------------|-----|
| | | W | WB |
| Label Sebenarnya | W | 709 | 60 |
| | WB | 98 | 483 |

Model B dilakukan dengan *kernel* 3×3 berjumlah 6 buah dan menghasilkan *confusion matrix* yang dapat dilihat pada Tabel 4.

Tabel 4. *Confusion Matrix* Model B

| Model B | | Hasil Deteksi | |
|------------------|----|---------------|-----|
| | | W | WB |
| Label Sebenarnya | W | 744 | 25 |
| | WB | 77 | 504 |

Untuk memberikan visualisasi yang lebih jelas dari hasil penerapan *kernel* 3×3 sejumlah 6 buah untuk ekstraksi ciri citra di CL1 dapat dilihat pada Gambar 7.



Gambar 7. Hasil Penerapan *Kernel* 3×3

Model C dilakukan menggunakan *kernel* 5×5 sejumlah 3 buah dengan hasil yang dapat dilihat pada Tabel 5.

Tabel 5. *Confusion Matrix* Model C

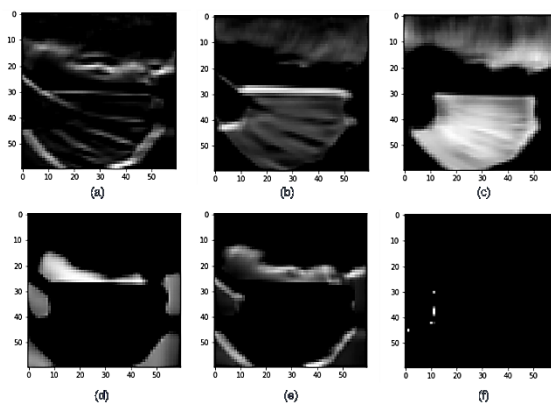
| Model C | | Hasil Deteksi | |
|------------------|----|---------------|-----|
| | | W | WB |
| Label Sebenarnya | W | 698 | 71 |
| | WB | 53 | 528 |

Model D dilakukan dengan *kernel* 5×5 berjumlah 6 buah dan menghasilkan *confusion matrix* yang dapat dilihat pada Tabel 6.

Tabel 6. *Confusion Matrix* Model D

| Model D | | Hasil Deteksi | |
|------------------|----|---------------|-----|
| | | W | WB |
| Label Sebenarnya | W | 731 | 38 |
| | WB | 26 | 555 |

Untuk memberikan visualisasi yang lebih baik dari hasil penerapan *kernel* 5×5 sejumlah 6 buah untuk ekstraksi ciri citra di CL1 dapat dilihat pada Gambar 8.



Gambar 8. Hasil Penerapan *Kernel* 5×5

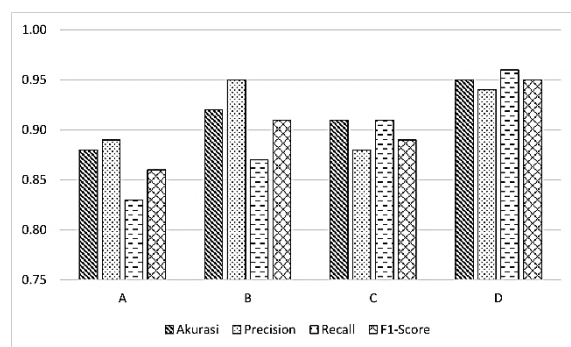
Hasil performa dari keseluruhan model dapat dilihat pada Tabel 7 dengan hasil performa terbaik didapatkan dari Model D sebesar 0,95 untuk nilai *f1-score* dan akurasi. Representasi dalam bentuk diagram batang dapat dilihat pada Gambar 9.

Tabel 7. Performa Model

| Model | Akurasi | Precision | Recall | F1-Score |
|-------|-------------|-------------|-------------|-------------|
| A | 0,88 | 0,89 | 0,83 | 0,86 |
| B | 0,92 | 0,95 | 0,87 | 0,91 |
| C | 0,91 | 0,88 | 0,91 | 0,89 |
| D | 0,95 | 0,94 | 0,96 | 0,95 |

Pada Gambar 9 dapat dilihat Model A dan B, keduanya memiliki nilai *recall* lebih rendah dibandingkan dengan Model C dan D. Hal ini dipengaruhi hasil perhitungan *confusion matrix* pada Tabel 3 dan 4 yang memperlihatkan nilai FN cenderung tinggi, sehingga berdampak terhadap nilai *recall*. Kesamaan antara Model A dan B adalah penggunaan *kernel* berukuran 3×3 . Sedangkan Model C dan D, keduanya memiliki nilai *precision* sedikit rendah yang dipengaruhi oleh nilai FP yang cenderung tinggi. Hal ini dapat dilihat pada Tabel 5 dan 6 yang dimana keduanya merupakan hasil penerapan *kernel* berukuran 5×5 .

Hubungan keterkaitan pada parameter ukuran *kernel* terhadap arsitektur model dapat mempengaruhi sensitivitas terhadap nilai *error* ketika ukuran *kernel* kecil dan akan memengaruhi nilai *precision* ketika ukuran *kernel* lebih besar. Hal ini disebabkan oleh *kernel* 3×3 yang kurang mampu melakukan ekstraksi ciri, sehingga dihasilkan ciri citra yang kurang terlihat. Hal ini dibuktikan dengan hasil penerapan *kernel* 3×3 CL1 yang dapat dilihat pada Gambar 7. Di sisi lain, *kernel* 5×5 mampu mendapatkan ciri citra *input* dengan baik seperti ciri *outline* mata, ciri masker, *outline* masker, *outline* wajah, dan batas wajah dengan masker seperti yang terlihat pada Gambar 8.

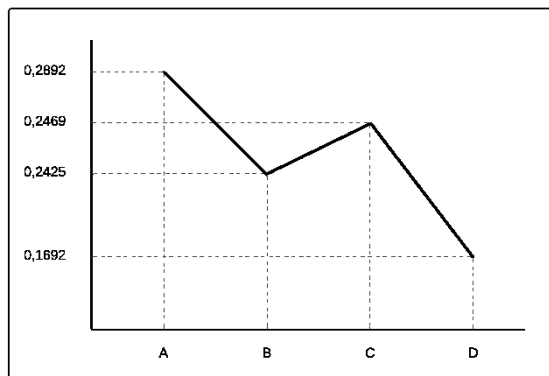


Gambar 9. Diagram Performa Model

Pada Gambar 9, Model B dan D dapat dilihat memiliki nilai yang lebih baik dibandingkan dengan Model A dan C. Dengan didasarkan hasil perhitungan *confusion matrix* pada Tabel 4 dan 6, keduanya

memiliki kesamaan pada jumlah *kernel* yang digunakan yaitu 6 buah *kernel*.

Hubungan keterkaitan parameter jumlah *kernel* terhadap arsitektur model dapat mempengaruhi hasil keseluruhan performa model. Lebih banyak *kernel* yang digunakan dalam CL, maka model mampu menghasilkan performa yang lebih baik. Jumlah *kernel* merepresentasikan jumlah ciri yang didapatkan dari citra. Sehingga, lebih banyak *kernel* yang diterapkan, maka ciri yang didapatkan dari citra *input* juga lebih banyak.



Gambar 10. Rata-rata Loss Model

Pada Gambar 10, dapat dilihat nilai rata-rata *loss* mengalami penurunan dalam tiap model, kecuali pada Model C yang memiliki kenaikan nilai rata-rata *loss* sebesar 0,0044 jika dibandingkan dengan hasil rata-rata *loss* Model B. Hal ini disebabkan oleh ukuran *kernel* yang digunakan Model B berbeda dengan Model C. Nilai rata-rata *loss* Model D lebih baik daripada Model C yang dipengaruhi jumlah *kernel* yang digunakan, Model D menggunakan jumlah *kernel* lebih banyak. Nilai tertinggi didapatkan oleh Model A sebesar 0,2892. Sedangkan nilai rata-rata *loss* terendah dimiliki oleh Model D dengan nilai 0,1692.

Tabel 8. Jumlah Parameter

| Model | A | B | C | D |
|------------------|--------|--------|--------|--------|
| Jumlah Parameter | 24176 | 95330 | 5432 | 11474 |
| Rata-rata Loss | 0,2892 | 0,2425 | 0,2469 | 0,1692 |

Pada Tabel 8 dapat dilihat jumlah parameter dari keseluruhan model. Jumlah parameter ini merupakan jumlah parameter yang dilatih dalam proses pelatihan data. Model A dan B menggunakan *kernel* berukuran 3×3 memiliki lebih banyak parameter, hal ini dikarenakan jumlah parameter bobot pada FCL mengikuti jumlah piksel *output* dari PL3 dimana ukuran *output*-nya memiliki ukuran lebih besar daripada ukuran dari hasil penerapan *kernel* 5×5. Tetapi, bukan berarti semakin banyak parameter yang dimiliki akan berdampak baik terhadap hasil deteksi. Hal ini dibuktikan dengan nilai rata-rata *loss* Model D yang memiliki nilai terbaik.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang didapatkan, Model D menghasilkan performa terbaik dalam deteksi citra wajah bermasker dengan nilai *f1-score* 0,95 dan akurasi 0,95. Penggunaan *kernel* 5×5 mendapatkan ciri citra yang lebih jelas, sehingga semakin banyak jumlah *kernel* yang diterapkan, maka semakin banyak pula ciri citra yang didapatkan. Akhirnya, dapat dihasilkan performa model yang terbaik. Semakin banyak jumlah parameter dalam suatu model, belum tentu menghasilkan performa yang lebih baik, karena hal ini juga bergantung pada ukuran *kernel* dengan tetap memperhatikan jumlah piksel perpindahannya. Berdasarkan pada temuan tersebut, maka dapat disimpulkan bahwa ukuran *kernel* dan jumlah *kernel* memiliki hubungan keterkaitan dalam menghasilkan performa model CNN yang lebih baik dalam deteksi citra wajah bermasker.

Saran dapat disampaikan untuk penelitian selanjutnya penggunaan parameter ukuran *kernel* dan jumlah *kernel* harus dapat disesuaikan dengan ukuran citra *input* dan dianjurkan untuk menggunakan *library* yang sudah ada jika arsitektur yang dibangun lebih besar dan menggunakan kombinasi parameter yang beragam.

DAFTAR PUSTAKA

- GOODFELLOW, I., BENGIO, Y. & COURVILLE, A., 2016. *Deep Learning*. Cambridge: The MIT Press.
- KRIZHEVSKY, A., SUTSKEVER, I. & HINTON, G. E., 2012. ImageNet Classification with Deep Convolutional Neural Network. *Advance in Neural Information Processing Systems*.
- LOY, J., 2019. *Natural Network Projects with Python*. Birmingham: Packt Publishing.
- MADENDA, S., 2015. *Pengolahan Citra dan Video Digital*. Jakarta: Erlangga.
- OUMINA, A., MAKHFI, N. E. & HAMDI, M., 2021. *Control The Covid-19 Pandemic: Face mask Detection Using Transfer Learning*. s.l., IEEE.
- PANDEY, V. K., GUPTA, V. K. & KUMAR, S., 2021. *Face Mask Detection Using Convolutional Neural Network*. Greater Noida, IEEE.
- SIDIK, R. P. & DJAMAL, E. C., 2021. *Face Mask Detection using Convolutional Neural Network*. Depok, IEEE.
- SIRADJUDDIN, I. A., R. & MUNTASA, A., 2021. *Faster Region-based Convolutional Neural Network for Mask Face Detection*. Semarang, IEEE.
- TING, K. M., 2011. *Confusion Matrix*. In: Sammut C., Webb G.i. (eds) *Encyclopedia of Machine Learning*. Boston: Springer.

- WANG, Z. et al., 2020. Masked Face Recognition Dataset and Application. *Computer Vision and Pattern Recognition*.
- WHO, 2020. *Mask use in the context of COVID-19 : World Health Organization*. [Online] Available at: www.who.int [Accessed 5 February 2021].
- WHO, 2021. *Health topics: Coronavirus*. [Online] Available at: <https://www.who.int/health-topics/coronavirus> [Accessed 23 August 2021].
- WHO, 2021. *Weekly epidemiological update : World Health Organization*. [Online] Available at: www.who.int [Accessed 5 February 2021].
- WIHANDIKA, R. C., 2021. Deteksi Masker Wajah Menggunakan Metode Adjacent Evaluation Local Binary Patterns. *Rekayasa Sistem dan Teknologi Informasi*, 5(4), pp. 705-712.
- ZHANG, Z., 2016. *Derivation of Backpropagation in Convolutional Neural Network (CNN)*, Knoxville: s.n.