IMPLEMENTASI ALGORITMA *CONVOLUTIONAL NEURAL NETWORK* UNTUK KLASIFIKASI CITRA JENIS SEPATU LARI BERDASARKAN PERMUKAAN LINTASAN

[1]Muhammad Ilham Aqli Dilan
[1]ilhamdilan@gmail.com

[1]Universitas Ahmad Dahlan

Abstract: *Many runners do not understand the running shoes used in track conditions. As a result, runners feel uncomfortable when running. Running shoes are divided into three according to the track: road-running shoes, trail-running shoes, and cross-training shoes which are used according to conditions. The author wants to make a breakthrough to help runners choose shoes before competing or running. By using a deep learning algorithm, namely a convolutional neural network, the author wants to help runners, both beginners and professionals, in choosing shoes by classifying the type of running shoe based on the track using a convolutional neural network algorithm. The method used in this research uses the CNN algorithm by creating 4 convolution layers, a kernel size of 2x2, the pooling used is max pooling with a size of 2x2 and 1024 hidden layers with the ReLU activation function. For the image size, use 128x128 with several channels of 3. For the first convolution layer, 64 filters are used, then the second convolution layer is 128, the third is 256, then the last is 512 with the ReLU activation function. For the output layer, use the softmax activation function. The number of parameters used is 34,248,387. The training data used is 500 per class with validation data of 150 per class. The results obtained were the level of accuracy using test data with a total of 50 data per class, getting a result of 0.846 with a loss result of 0.376. Then the precision for the Cross Training class was 0.92, Road Running was 0.75, and Trail Running was 0.90. Recall in the Cross Training class was 0.70, Road Running was 0.92, and Trail Running was 0.92. F1-score in Cross Training is 0.92, Road Running is 0.75, and Trail Running is 0.90.*

Keywords: *convolutional neural network* 1; *deep learning* 2*; image classification* 3*; running shoes based on the track surface* 4;

Abstrak: Banyak pelari yang tidak paham dengan sepatu lari yang digunakan di kondisi lintasan. Akibatnya pelari merasa tidak nyaman dalam melakukan berlari. Sepatu lari dibedakan menjadi tiga sesuai lintasannya: road-running shoes, trail-running shoes, dan cross-training shoes yang digunakan sesuai kondisi. Penulis ingin membuat sebuah trobosan untuk membantu pelari dalam menentukan sepatu sebelum melakukan lomba atau sedang berlari. Dengan menggunakan algoritma *deep learning* yaitu *convolutional neural network*, penulis ingin membantu para pelari baik pemula maupun yang sudah professional dalam menentukan sebuah sepatu dengan mengklasifikasi jenis sepatu lari berdasarkan lintasan menggunakan algoritma *convolutional neural network*. Metode yang digunakan dalam penelitian

ini menggunakan algoritma CNN dengan membuat 4 lapisan konvolusi, ukuran *kernel* 2x2, *pooling* yang digunakan adalah *maxpooling* dengan ukuran 2x2, serta lapisan *hidden* sebanyak 1024 dengan fungsi aktivasi ReLU. Untuk ukuran citra menggunakan 128x128 dengan jumlah *channel* sebesar 3. Untuk lapisan konvolusi yang pertama menggunakan jumlah filter sebanyak 64, lalu konvolusi ke dua sebanyak 128, ketiga 256, lalu yang terakhir 512 dengan fungis aktivasi ReLU. Untuk lapisan *output* menggunakan fungsi aktivasi *softmax*. Untuk jumlah parameter yang digunakan sebanyak 34.248.387. Data *training* yang digunakan sebanyak 500 per kelas dengan data *validation* sebesar 150 per kelas. Hasil yang diperoleh adalah tingkat akurasi menggunakan data *test* dengan jumlah data sebanyak 50 perkelas mendapatkan hasil sebesar 0,846 dengan hasil *loss* sebesar 0,376. Lalu untuk *precision* kelas *Cross Training* sebesar 0,92, *Road Running* sebesar 0,75, serta *Trail Running* sebesar 0.90. *Recall* pada kelas *Cross Training* sebesar 0,70, *Road Running* sebesar 0,92, serta *Trail Running* sebesar 0,92. *F1-score* pada *Cross Training* sebesar 0,92, *Road Running* sebesar 0,75, serta *Trail Running* sebesar 0,90.

Kata Kunci: convolutional neural network 1; deep learning 2; klasifikasi citra 3; sepatu lari berdasar lintasannya 4;

I. PENDAHULUAN

Penggunaan sepatu ketika berlari sangatlah penting. Banyak manfaat yang diperoleh pelari ketika menggunakan sepatu. Menurut artikel dari kompas.tv pada tanggal 22 Januari 2022, manfaat pelari ketika menggunakan sepatu antara lain menghindari pelari dari cidera seperti menghindari benda asing yang dapat melukai permukaan kaki(Surya, 2022). Tidak jarang pelari diluar sana baik yang sudah profesional(atlet) maupun tidak(hanya untuk hobby) menggunakan sepatu agar kaki juga terlindung dari bahaya yang tidak terduga.

Setiap pelari juga wajib mengetahui permukaan lari yang akan dilaluinya. Umumnya ada tiga jenis permukaan lintasan yaitu daratan datar dan keras, *trail*, dan *indoor*. Lintasan daratan datar dan keras umumnya berada di jalan raya(biasanya aspal), trotoar, maupun trek lari baik sintetis, rumput, serta sintetis. Lalu lintasan *trail* yaitu lintasan yang permukaannya berupa perbukitan atau gunung. Yang terakhir adalah permukaan *indoor* yang berupa lari di tempat *treadmill* atau tempat *gym* yang digunakan untuk olahraga *crossfit*.

Pelari juga wajib memilih sepatu sesuai dengan permukaan lintasan yang akan dilalui. Pelari juga harus tau sepatu mana yang cocok digunakan untuk lintasan jalan raya, lintasan yang banyak bebatuan, atau lari menggunakan alat *treadmil* yang bisa ditemui di tempat *gym*. Menurut artikel dari hellosehat.com pada tanggal 8 Maret 2022, sepatu dibedakan menjadi tiga

menurut lintasan: *road-running shoes*, *trail-running shoes*, dan *cross-training shoes*. Sepatu jenis *road-running* biasanya digunakan untuk lari di jalan raya, trotoar, atau daratan yang permukaannya datar dan keras. Lalu sepatu jenis *trail-running* digunakan untuk berlari di lintasan perbukitan yang biasanya dipenuhi oleh batu, lelumpuran, atau ranting pohong untuk memberikan stabilitas serta perlindungan ekstra kaki selama menghadapi jalur ekstrem. Yang terakhir adalah sepatu jenis *cross-training* yang digunakan untuk pelari di dalam *gym* atau latihan *crossfit*(Firdaus, 2022).

Dari penjelasan mengenai sepatu lari, maka pelari tidak perlu repot dalam memilih sepatu yang sesuai dengan lintasan yang akan dijalani. Jika pelari ingin berlari di jalan, trotoar, atau lintasan karet maka sepatu yang baik adalah jenis *road-running*. Jika ingin berlari di area perbukitan maka pelari wajib memilih jenis sepatu *trail-running* dan pelari ketika melakukan latihan *crossfit* maka wajib memilih *cross-training*. Namun, permasalahan muncul ketika pelari tidak bisa memilih sepatu yang tepat. Mereka hanya memilih sepatu dengan desain yang bagus namun tidak memperhatikan jenis lintasan lari yang akan dilakukan. Jika pelari tidak bisa membedakan sepatu tersebut, maka yang terjadi adalah sepatu tidak nyaman untuk lari yang bisa mengakibatkan cedera baik ringan atau parah. Menurut artikel dari voi.id bahwa salah memilih sepatu untuk berolahraga berakibat fatal seperti meningkatnya kemungkinan cedera, cedera pada kaki, serta keretakan pada tulang kaki(Tsaqofi, 2022). Dengan permasalahan yang timbul, muncul ide penulis untuk membantu para pelari untuk mencari sepatu sesuai dengan lintasan lari dengan cara mengklasifikasi sepatu tersebut dengan citra sepatu sesuai jenis sepatu lari. Data citra akan diproses menggunakan algoritma klasifikasi citra dan mendeteksi jenis sepatu tersebut. Dengan begitu diharapkan bisa membantu para pelari agar tidak kebingungan dalam mencari sepatu berdasarkan jenis lari.

Algoritma klasifikasi citra yang terkenal menurut jurnal sebelumnya yaitu *K-Nearest Neighbors*(KNN), *Support Vector Machine*(SVM), dan *Convolutional Neural Network*(CNN)(Naufal, 2021). Ketiga algoritma tersebut memiliki pengertian dan pendekatan berbeda-beda. Algoritma *K-Neariest Neighbors*(KNN) merupakan algoritma yang berfungsi untuk mengklasifikasi terhadap sesuatu objek, bersumber pada *k* buah data latih yang memiliki jarak sangat dekat dengan objek tersebut dengan ketentuan nilai *k* tidak boleh lebih besar dari jumlah data latih serta nilai *k* harus ganjil dan lebih dari satu(Rivki &

Bachtiar, 2017). Algoritma *Support Vector Machine*(SVM) merupakan metode yang dimiliki oleh *machine learning* dengan menerapkan prinsip *Structural Risk Minimization*(SRM) yang dimana bertujuan menciptakan *hyperplane* terbaik yang memisahkan kedua *class* pada *input space*(As et al., 2020). Algoritma *Convolutional Neural Network*(CNN) ialah pengembangan dari salah satu bagian dari *Multilayer Perceptron*(MLP) serta salah satu bagian dari algoritma jaringan syaraf tiruan(*neural network*) yang dibuat untuk mengolah data dua dimensi salah satunya merupakan data citra(Suartika E. P, I Wayan, Wijaya Arya Yudhi, 2016). Algoritma *Convolutional Neural Network*(CNN) digunakan untuk mengklasifikasikan data berlabel dengan memakai metode *supervised learning* di mana tujuan *input* yang diharapkan diterima melalui jaringan sudah dikenal sebelumnya(Wulandari et al., 2020).

Ketiga algoritma tersebut mempunyai kelebihan serta kekurangan masing-masing. Dari riset sebelumnya tentang membandingkan ketiga algoritma tersebut, algoritma *Convolutional Neural Network* sangat bagus digunakan buat klasifikasi citra dengan performa cukup tinggi dibandingkan algoritma *Support Vector Machine* serta *K-Nearest Neighbors* dengan jumlah *accuracy*, *precision*, *recall*, dan F1 *score* paling tinggi dibandingkan dengan algoritma *Support Vector Machine* dan *K-Nearest Neighbors* meski waktu yang diperlukan untuk meng *compile* data sangat lambat(Naufal, 2021). Dari penjelasan diatas, maka penerapan algoritma *Convolutional Neural Network* sangat efektif pada penelitian ini tentang klasifikasi sepatu lari berdasarkan lintasan.

Rumusan masalah yang dilakukan pada penelitian ini adalah sebagai berikut:

1. Bagaimana merancang dan membuat sistem klasifikasi jenis sepatu lari menggunakan algoritma *convolutional neural network*?
2. Bagaimana cara mengetahui tingkat akurasi yang didapat dalam pengklasifikasian jenis sepatu lari dengan menerapkan *convolutional neural network*?\

Berdasarkan rumusan masalah yang telah dipaparkan diatas, maka tujuan penelitian ini adalah sebagai berikut:

1. Mengimplementasikan algoritma *convolutional neural network* untuk klasifikasi jenis sepatu lari berdasarkan lintasan.
2. Mengetahui akurasi hasil implementasi dari algoritma *convolutional neural network* untuk klasifikasi.

Manfaat penelitian yang akan didapat dari penelitian ini adalah sebagai berikut:

1. Diperoleh suatu cara untuk mengklasifikasi jenis sepatu lari

berdasarkan lintasan lari dengan menggunakan algoritma *convolutional neural network*.

2. Membantu para pelari khususnya yang baru pemula untuk menentukan jenis sepatu yang akan mereka pilih.

II. METODE

Proses pengambilan data yang dilakukan pada penelitian ini menggunakan metode dengan penjelasan sebagai berikut:

1. Metode Studi Literatur dilakukan untuk mengetahui pengertian tentang algoritma *convolutional neural network* (CNN) dan cara menggunakannya. Lalu jurnal terkait sebelum menerapkan algoritma CNN. Terakhir adalah artikel tentang ciri-ciri sepatu lari yang akan menjadi objek penelitian.
2. Metode observasi untuk memperoleh informasi perbedaan ketiga sepatu lari yang akan menjadi objek penelitian.
3. Pengambilan dataset dibagi menjadi dua yaitu data primer dan sekunder. Data primer diambil dengan cara mengambil foto di toko sepatu olahraga dibawah naungan PT. MAP Active Adi Perkasa yang berada di Pakuwon Mall, Sleman, D.I Yogyakarta dengan cara memfoto sepatu lari di 6 sisi: Sisi atas, sisi dalam, sisi dalam atas, sisi luar, sisi luar atas, serta sol sepatu. Sementara data sekunder berasal dari *website* sepatu lari terkenal seperti Adidas, Asics, Hoka, New Balance, Nike, Puma, Reebok, Skecher, serta Under Armour. Jika data kurang, maka bisa mengambil dari toko online seperti *tokopedia.com*, *shopee.co.id*, *sportstation.id*, *planetsports.asia*, atau *zalora.co.id*. Setelah mendapatkan citra tersebut, peneliti melakukan *preprocessing* citra yang datanya akan diolah ke dalam model CNN.

Metode yang digunakan pada penelitian ini adalah menerapkan algoritma *convolutional neural network* klasifikasi citra sepatu lari berdasarkan permukaan lintasan.

Urutan penelitian ini adalah dimulai dengan mengumpulkan data, lalu *preprocessing* data, klasifikasi citra menggunakan algoritma CNN, pengujian model, terakhir melihat akurasi dan hasil klasifikasi. Berikut tahapan yang digunakan dalam penelitian ini:

1. Pengumpulan Data Citra

   Tahap pertama diawali dengan pengumpulan data citra. Peneliti mengumpulkan data,

mengelompokkan data sesuai kelas, dan membuat dataset. Pengambilan data dilakukan dengan cara memfoto sepatu lari menggunakan kamera *smartphone* serta mengunduh foto di internet.

2. *Preprocessing* Data

Pembagian data untuk penelitian ini dengan jumlah seperti pada Tabel …

Tabel 1 Jumlah Dataset

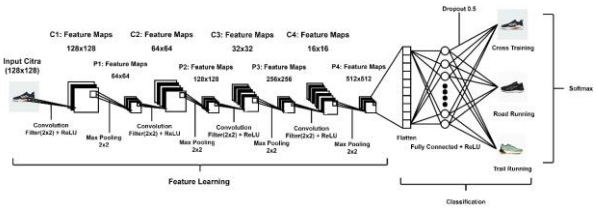
No	Dataset	Jumlah Kelas	Total Data	Data Per Kelas
1	Train	3	1500	500
2	Valid	3	450	150
3	Test	3	150	50

Pada Tabel 1 total yang digunakan pada penelitian adalah 2.100 data citra yang terbagi menjadi 3 kelas sehingga per kelas mendapatkan jumlah data sebesar 700. Untuk pembagian data *train, valid,* serta *test* sebesar 500, 150, dan 50 per kelas. Selanjutnya adalah melakukan proses *augmentasi* data dengan ketentuan sebagai berikut:

1. *Rescale=1./255*
2. *Rotation_range = 5*
3. *Widht_shift_range = 0.1*
4. *Height_shift_range = 0.1*
5. *Shear_range = 0.1*
6. *Zoom_range = 0.2*
7. *Horizontal_flip = True*
8. *Vertical_flip = False*
9. *Fill_mode = 'nearest'*

3. Klasifikasi Citra menggunakan *Convolutional Neural Network*

Gambar … merupakan arsitektur CNN yang digunakan pada penelitian ini



Gambar 1 Arsitektur CNN

Pada Gambar 1, konvolusi pertama menerapkan *filter* sebanyak 64, *max pooling* berukuran 2x2, serta *kernel* matriks 2x2. Konvolusi kedua menerapkan *filter* sebanyak 128, *max pooling* berukuran 2x2, serta *kernel* matriks 2x2. Konvolusi ketiga menerapkan filter sebanyak 256, *max pooling* berukuran 2x2, serta *kernel* matriks 2x2. Konvolusi ketiga menerapkan filter sebanyak 512, *max pooling* berukuran 2x2, serta *kernel* matriks 2x2. Setelah itu, di tahap *flatten* dilakukan proses untuk mengubah *output* dari proses konvolusi yang berupa matriks menjadi sebuah vektor, dengan dilanjutkan ke proses klasifikasi menggunakan *fully connected layer* dengan jumlah *neuron* pada lapisan

tersembunyi(*hidden layer*) sebesar 1024 dengan aktivasi ReLU. Jika sudah, maka pada lapisan terakhir yaitu mengklasifikasikan citra sepatu dengan jumlah *neuron* sama dengan kelas dataset berjumlah 3 yang terdiri dari *Cross Training*, *Road Running*, dan *Trail Running* dengan aktivasi *Softmax*.

4. Pengujian Model

   Setelah melakukan perencanaan model, langkah selanjutnya adalah dengan melakukan pengujian model. Pengujian model digunakan untuk mengimplementasikan perencanaan model yang sudah dibangun. Dalam tahap uji coba model, jumlah iterasi yang akan digunakan untuk memastikan berapa kali jaringan akan melihat seluruh kumpulan data yaitu dengan *epoch* sebesar 80, 90, dan 100 dengan *learning rate* sebesar 0,00001 dan 0,0001.

5. Hasil Akurasi dan Klasifikasi

   Hasil akurasi model menggunakan akurasi pada data *test* lalu untuk hasil klasifikasi menggunakan metode *confession matrix*.

III. HASIL DAN PEMBAHASAN

Hasil

Dilakukan pengujian parameter model, berikut parameternya serta hasil yang diperoleh:

1. Pengaruh Ukuran Input Citra

   Peneliti melakukan percobaan pengaruh ukuran input citra terhadap hasil akurasi yang diperoleh. Input citra yang dipakai antara lain 64x64, 128x128, serta 224x224. Untuk *epochs* peneliti menggunakan 80 serta *learning rate* sebesar 0,0001 dengan ukuran kernel sebesar 2x2. Berikut hasil yang diperoleh:

   Tabel 2 Perbandingan Ukuran Citra

No	Ukuran Citra	*Accuracy*	*Loss*	*Val_accuracy*	*Val_loss*
1	64x64	0,756	0,567	0,68	0,793
2	128x128	0,855	0,371	0,77	0,731
3	224x224	0,868	0,33	0,742	0,906

   Tabel 2 merupakan hasil yang didapat ketika melakukan percobaan. Dari hasil tersebut maka ukuran citra 224x224 lebih baik daripada kedua ukuran citra. Hasil yang diperoleh adalah tingkat *training accuracy* sebesar 0,868 dengan *training loss* sebesar 0,33. Namun untuk *validation* yang terbaik adalah ukuran citra 128x128 dengan *accuracy* sebesar 0,77 sementara untuk *loss* sebesar 0,731.

2. Pengaruh Ukuran Kernel

   Peneliti melakukan percobaan pengaruh ukuran kernel terhadap hasil

akurasi model. Peneliti melakukan percobaan dengan ukuran kernel 2x2, 3x3, serta 4x4 dengan jumlah epoch 80 dan *learning rate* sebesar 0,0001. Berikut hasil yang diperoleh:

Tabel 3 Perbandigan Ukuran Kernel

No	Ukuran Kernel	*Accuracy*	*Loss*	*Val_accuracy*	*Val_loss*
1	2x2	0,855	0,371	0,77	0,731
2	3x3	0,907	0,236	0,758	0,882
3	4x4	0,943	0,138	0,747	1,18

Tabel 3 merupakan hasil perbandingan ukuran kernel yang dilakukan. Hasil yang terbaik diperoleh adalah ukuran kernel 2x2 lebih baik dari pada kedua ukuran *kernel* lainnya. Meski hasil *training accuracy* lebih besar ukuran *kernel* 4x4, namun untuk *validation accuracy* dan *loss* lebih baik menggunakan ukuran 2x2, sebesar 0,77 dan 0,731.

3. Pengaruh Jumlah *Epoch*

Peneliti melakukan percobaan pengaruh jumlah *epoch* terhadap hasil akurasi yang diperoleh model. *Epoch* yang digunakan yaitu 70, 80, serta 90 dengan *learning rate* sebesar 0,0001 dan ukuran kernel sebesar 2x2. Berikut hasil yang diperoleh:

Tabel 4 Pengaruh Jumlah *Epoch*

No	*Epoch*	*Accuracy*	*Loss*	*Val_accuracy*	*Val_loss*
1	70	0,795	0,709	0,709	0,761
2	80	0,855	0,371	0,77	0,731
3	90	0,846	0,389	0,733	0,882

Tabel 4 merupakan hasil perbandingan jumlah *epoch*. Hasil *training* yang terbaik adalah menggunakan *epoch* 80 karena mendapatkan nilai akurasi paling besar daripada *epoch* 70 dan 90. Hasil yang diperoleh adalah untuk *training accuracy* sebesar 0,855 sementara untuk *training loss* sebesar 0,371. Untuk *validation accuracy* serta *loss* juga lebih baik dari kedua *epoch* lainnya yaitu 0,77 dan 0,731.

4. Pengaruh *Learning Rate*

Peneliti juga melakukan percobaan terhadap pengaruh *learning rate* ke model yang sudah dibuat. *Learning rate* yang digunakan adalah sebesar 0,0001 dan 0,00001. *Epoch* dan *kernel* yang digunakan sebesar 80 dan 2x2. Berikut hasil yang diperoleh:

Tabel 5 Perbandingan *Learning Rate*

No	*Learning Rate*	*Accuracy*	*Loss*	*Val_accuracy*	*Val_loss*
1	0,0001	0,855	0,371	0,77	0,731
2	0,00001	0,554	0,915	0,885	0,580

Dari hasil Tabel 5, maka *learning rate* terbaik sebesar 0,0001. *Learning rate* 0,0001 memiliki *training* akurasi sebesar 0,855 dan *training loss* sebesar 0,371. Sementara untuk *learning rate* 0,00001 memiliki *training* akurasi sebesar 0,554 dan *training loss* sebesar 0,915. Untuk *validation accuracy* serta *loss*nya juga lebih baik dari *learning rate* 0,00001 sebesar 0,77 dan 0,731.

5. Pengaruh *Convolutional Layer*

Peneliti melakukan percobaan pengaruh convolutional layer dan hidden layer terhadap model yang dibuat. Untuk *convolutional layer* menggunakan skema 2 *layer*, 3 *layer*, serta 4 *layer* dengan ketentuan ukuran *kernel* 2x2, dengan *epoch* 80, *learning rate* 0,0001, serta ukuran citra 128x128. Layer 2 dengan ukuran 128 lalu *layer* 3 hingga 4 dengan kelipatan 2. Berikut hasilnya:

Tabel 6 Perbandingan *Convolutional Layer*

No	*Layer Ke*	*Accuracy*	*Loss*	*Val_accuracy*	*Val_loss*
1	2	0,700	0,664	0,671	0,755
2	3	0,759	0,558	0,649	0,849
3	4	0,855	0,371	0,77	0,731

Dari hasil dari Tabel dapat disimpulkan bahwa memiliki 4 *layer* konvolusi lebih baik daripada hanya memiliki 2 atau 3 *layer*. Nilai akurasi baik *training* maupun *validation* lebih tinggi daripada kedua *layer* konvolusi. Untuk nilai *loss* juga lebih rendah dari kedua *layer* konvolusi.

6. Pengaruh *Hidden Layer*

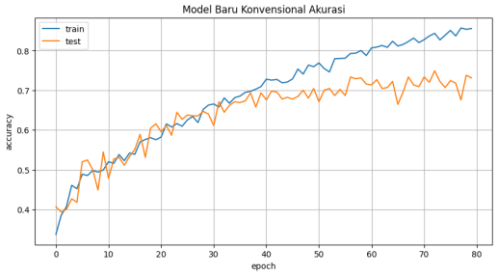
Peneliti melakukan percobaan dengan menambah hidden layer 3 dengan ketentuan ukuran kernel 2x2, dengan epoch 80, learning rate 0,0001, serta ukuran citra 128x128. *Hidden layer* 1 menggunakan *neuron* 1024, lalu menambahkan 2 *hidden layer* berjumlah 1024 dan 512, lalu 3 *hidden layer* berjumlah 256, 512, 1024. Berikut hasilnya:

Tabel 7 Perbandingan *Hidden Layer*

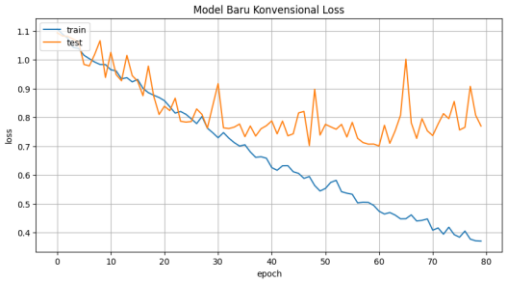
No	*Layer Ke*	*Accuracy*	*Loss*	*Val_accuracy*	*Val_loss*
1	1	0,855	0,371	0,77	0,731
2	2	0,849	0,371	0,756	0,759
3	3	0,854	0,376	0,764	0,856

Dari hasil dari Tabel maka hasil terbaik adalah menggunakan 1 *hidden layer* dengan jumlah *neuron* sebanyak 1024 karena nilai akurasi *training* serta nilai akurasi *validasi* paling tinggi dari ketiganya. Meski nilai *loss* pada *training* pada 1 *hidden layer* sama dengan 2 *hidden layer* namun untuk nilai *loss* pada validasi lebih rendah yang menggunakan 1 *hidden layer*.

Hasil training dan validation yang diperoleh dengan data training sebesar 1.500 serta validation sebesar 450 adalah sebagai berikut:



(a)



(b)

Gambar 2 Hasil (a) Akurasi dan (b) *Loss* pada Model CNN

Berdasarkan hasil dari Gambar 2, akurasi yang diperoleh untuk *train* sebesar 0,855 dengan *loss* sebesar 0,371. Sementara untuk hasil akurasi *valid* yang diperoleh sebesar 0,731 dengan *loss* sebesar 0,770. Hasil ini diperoleh dengan menggunakan model dengan ukuran citra sebesar 128x128, dengan jumlah kernel yaitu 2x2, *epoch* yang digunakan adalah 80, serta *learning rate* sebesar 0,0001**.**

Akurasi data test yang diperoleh dengan model yang sudah dibuat yang dimana memiliki *epoch* 80, *learning rate* 0,0001, ukuran citra 128x128, serta ukuran kernel berukuran 2x2 dengan 4 layer konvolusi serta 1 *hidden layer* adalah sebagai berikut:

```
[ ]  test_loss, test_acc = model.evaluate(test_gen, verbose=2)
     print('\n Test Accuracy: ', test_acc)

     5/5 - 8s - loss: 0.3761 - accuracy: 0.8467 - 8s/epoch - 2s/step

      Test Accuracy:  0.8466666693687439
```

Gambar 3 Hasil Akurasi Data *Test*

Gambar 3 merupakan hasil yang diperoleh untuk menguji data *test*. Hasil yang diperoleh menggunakan model tersebut adalah akurasi sebesar 0,846 dengan hasil loss sebesar 0,376.

Confussion matrix yang diperoleh dengan model yang sudah dibuat yang dimana memiliki *epoch* 80, *learning rate* 0,0001, ukuran citra 128x128, serta ukuran kernel berukuran 2x2 adalah sebagai berikut:

1. Kelas *Cross Training*

   Berikut hasil yang diperoleh:

   Tabel 8 *Confussion Matrix Cross Training*

	Positif	Negatif
Positif	35	15
Negatif	3	97

   Hasil yang diperoleh dari kelas *Cross Training* pada Tabel, terdapat 35 gambar yang terprediksi benar sesuai kelasnya. Sementara yang tidak terprediksi dengan benar sebanyak sejumlah 15 gambar.

2. Kelas *Road Running*

   Berikut hasil yang diperoleh:

   Tabel 9 *Confussion Matrix Road Running*

	Positif	Negatif
Positif	46	4
Negatif	15	85

   Hasil yang diperoleh dari kelas *Road Running* pada Tabel, terdapat 46 gambar yang terprediksi benar sesuai kelasnya. Sementara yang tidak terprediksi dengan benar sebanyak sejumlah 4 gambar.

3. Kelas *Trail Running*

   Berikut hasil yang diperoleh:

   Tabel 10 *Trail Running Confussion Matrix*

	Positif	Negatif
Positif	46	4
Negatif	5	95

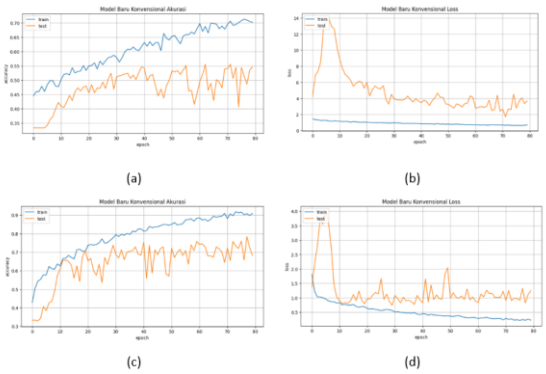
Hasil yang diperoleh dari kelas *Trail Running* pada Tabel 4.12, terdapat 46 gambar yang terprediksi benar sesuai kelasnya. Sementara yang tidak terprediksi dengan benar sebanyak sejumlah 4 gambar.

Pembahasan

Hasil yang diperoleh pada penelitian ini mendapatkan nilai akurasi sebesar 0,844 dengan loss sebesar 0,375 dengan menggunakan model yaitu ukuran citra berukuran 128x128, dengan ukuran kernel 2x2, jumlah *epoch* sebesar 80, jumlah *learning rate* sebesar 0,0001, dengan 4 layer konvolusi serta 1 *hidden layer*. Namun hasil tersebut masih belum dinyatakan belum optimal karena model tersebut masih mengalami fenomena *underfitting*. Oleh karena itu dilakukan percobaan untuk mengurangi *underfitting* yaitu dengan berikut:

1. Menambahkan *BatchNormalization*

   Pengujian ini menambahkan fungsi *BatchNormalization()* yang merupakan salah satu cara agar fenomena *underfitting* atau *overfitting* bisa berkurang atau terhindari. Peneliti melakukan 2 skenario penggunaan *BatchNormalization*: Menambahkan semua *BatchNormalization* dengan *Dropout* di semua lapisan dan hanya menambahkan *BatchNormalization* dengan *Dropout* hanya di *hidden layer*. Berikut hasil yang diperoleh dengan grafik:

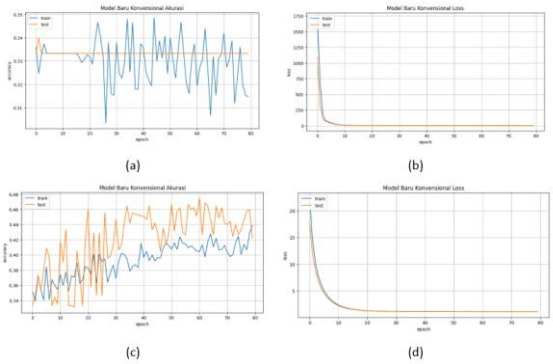


Gambar 4 (a) Akurasi (b) *Loss Dropout+Batch Normalisasi* dan (c) Akurasi (d) *Loss Batch Normalisasi*

Gambar 4 merupakan hasil yang dilakukan peneliti dalam mengurangi *underfitting*. Namun baik grafik (a), (b), (c), serta (d) hasil tersebut masih belum mendapatkan hasil maksimal. Oleh karena itu menambahkan fungsi *BatchNormalization()* dengan kedua skenario masih kurang optimal sehingga tidak perlu dilanjutkan kembali.

2. Menambahkan *Kernel Regularize*

   Pengujian ini menambahkan *Regularize* ke dalam lapisan CNN. Pada *Regularize* ada 2 macam yaitu L1 dan L2. Peneliti mencoba menambahkan *Regularize* ke dalam lapisan model yang sudah dibuat dengan menggunakan kedua macam tersebut. skenarionya adalah menggunakan pengaturan bawaan L1 dan L2. Berikut hasilnya:

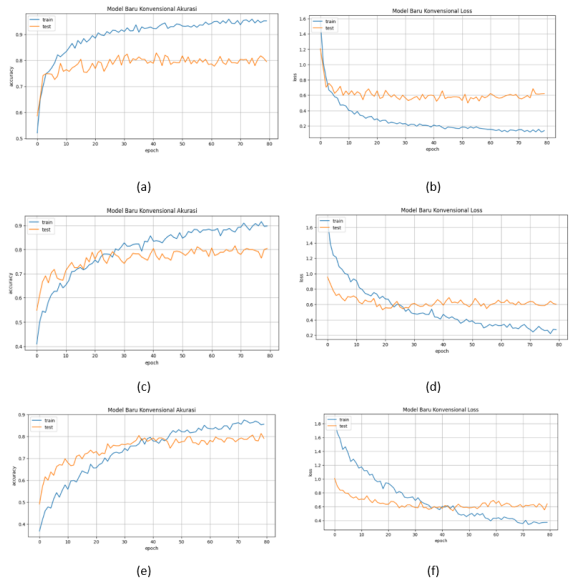


Gambar 5 Grafik (a) Akurasi dan (b) *Loss Regularize L1* sementara (c) Akurasi dan (d) *Loss Regularize L2*

Pada *Regularize* L1 pada Gambar 5 (a) dan (b) jika dibanding dengan *Regularize* L2 pada Gambar 5 (c) dan (d) lebih buruk, namun *Regularize* L2 juga tidak bagus karena terjadi *overfitting*. Meski nilai lossnya turun namun nilai akurasinya tidak stabil. Oleh karena itu penambahan *Regularize* tidak sesuai dengan kasus ini.

3. Menerapkan Model *Transfer Learning(MobileNetv2)*

   Menggunakan model MobileNetV2 dengan 3 skenario yaitu menambahkan 1 hidden layer dengan jumlah neuron 1024, 2 hidden layer dengan jumlah 512 dan 256, serta menambahkan 3 hidden layer dengan jumlah neuron 256, 512, dan 1024. Berikut hasil grafik yang diperoleh:

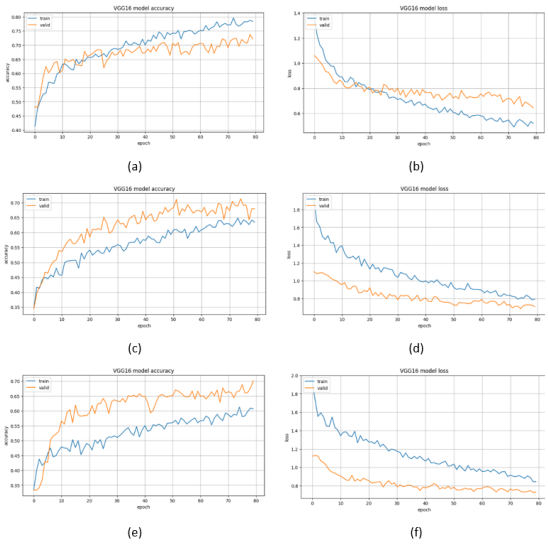


Gambar 6 Grafik (a) Akurasi (b*) Loss MobilenetV2 1 Hidden Layer*, (c) Akurasi dan (d) *Loss MobilenetV2 2 Hidden Layer*, serta (e) Akurasi (f) *Loss MobilenetV2 3 Hidden Layer*

Dari hasil yang sudah ditampilkan pada Gambar 6, maka model yang terbaik menggunakan MobileNetV2 dengan 2 dan 3 *hidden layer* seperti Gambar 6 bagian (c), (d), (e), serta (f). Grafik tersebut mendapatkan hasil pelatihan yang optimal daripada bagian (a) dan (b).

4. Menerapkan Model *Transfer Learning(VGG16)*

   Menggunakan model VGG16 dengan 3 skenario yaitu menambahkan 1 hidden layer dengan jumlah *neuron* 1024, 2 *hidden layer* dengan jumlah 512 dan 256, serta menambahkan 3 *hidden layer* dengan jumlah *neuron* 256, 512, dan 1024. Berikut hasil grafik yang diperoleh:



Gambar 7 Grafik (a) Akurasi (b) *Loss VGG16 1 Hidden Layer*, (c) Akurasi (d) *Loss VGG16 2 Hidden Layer*, dan (e) Akurasi (f) *Loss VGG16 3 Hidden Layer*

Dari hasil yang diperoleh pada Gambar 7, dapat disimpulkan bahwa menggunakan VGG16 lebih bagus dengan menerapkan 1 *hidden layer* dengan hasil yang lebih baik daripada kedua percobaan lainnya.

Dari percobaan yang sudah dilakukan, lebih baik menerapkan penambahan model *transfer learning* dengan menggunakan MobileNetV2 dan VGG16 dengan menambahkan *hidden layer* sesuai dengan penggunaan agar mengurangi fenomena *underfitting*.

Untuk akurasi yang diperoleh dari hasil *confusion matrix* adalah sebagai berikut:

$$Accuracy = \frac{TP + TN}{Jumlah\ Semua\ Data}$$

$$Accuracy = \frac{35 + 46 + 46}{150}$$

$$Accuracy = \frac{127}{150} = 0.846$$

Hasil akurasi yang diperoleh menggunakan rumus tersebut adalah 0,846. Akurasi diperoleh dari jumlah data yang dapat terprediksi dengan benar dibagi dengan total data. Untuk nilai *precision* setiap kelas mendapatkan hasil sebagai berikut:

1. *Cross Training*

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{35}{35 + 2 + 1}$$

$$Precision = \frac{35}{38} = 0{,}92$$

2. *Road Running*

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{46}{46 + 12 + 3}$$

$$Precision = \frac{46}{61} = 0{,}75$$

3. *Trail Running*

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{46}{46 + 2 + 3}$$

$$Precision = \frac{46}{51} = 0{,}90$$

Untuk nilai *recall* tiap kelas adalah sebagai berikut:

1. *Cross Training*

$$Recall = \frac{TP}{TP + FN}$$
$$Recall = \frac{35}{35 + 12 + 1}$$
$$Recall = \frac{35}{50} = 0{,}70$$

2. *Road Running*

$$Recall = \frac{TP}{TP + FN}$$
$$Recall = \frac{46}{46 + 2 + 2}$$
$$Recall = \frac{46}{50} = 0{,}92$$

3. *Trail Running*

$$Recall = \frac{TP}{TP + FN}$$
$$Recall = \frac{46}{46 + 3 + 1}$$
$$Recall = \frac{46}{50} = 0{,}92$$

Untuk nilai *F1-Score* per kelas adalah sebagai berikut:

1. *Cross Training*

$$F1 - Score = 2x\ \frac{Precision.Recall}{Precision + Recall}$$
$$F1 - Score = 2x\frac{0{,}92x0{,}7}{0{,}92 + 0{,}7}$$
$$F1 - Score = 2x\ \frac{0{,}644}{1{,}62} = 0{,}795 = 0{,}80$$

2. *Road Running*

$$F1 - Score = 2x\frac{Precision.Recall}{Precision + Recal}$$
$$F1 - Score = 2x\frac{0{,}75x0{,}92}{0{,}75 + 0{,}92}$$
$$F1 - Score = 2x\frac{0{,}69}{1{,}67} = 0{,}826 = 0{,}83$$

3. *Trail Running*

$$F1 - Score = 2x\frac{Precision.Recall}{Precision + Recall}$$
$$F1 - Score = 2x\frac{0{,}90x0{,}92}{0{,}90 + 0{,}92}$$
$$F1 - Score = 2x\frac{0{,}828}{1{,}82} = 0{,}909 = 0{,}91$$

Maka dari itu, hasil yang diperoleh dari program adalah sebagai berikut:

Tabel 11 Model *Confussion Matrix* CNN

Jenis Sepatu	Precision	Recall	F1-Score	Supprot
Cross Training	0,92	0,70	0,80	50
Road Running	0,75	0,92	0,83	50
Trail Running	0,90	0,92	0,91	50

Tabel 11 merupakan perhitungan yang diperoleh dari program. Hasil dari perhitungan manual dan program menunjukkan nilai yang sama.

IV. SIMPULAN

Hasil penelitian yang sudah dilakukan tentang perancangan serta pengujian model dengan menggunakan algoritma *convolutional neural network*, diperoleh kesimpulan bahwa:

1. Penentuan model dilakukan dengan menguji beberapa parameter seperti ukuran citra, ukuran kernel, jumlah *epoch*, serta jumlah *learning rate*. Diperoleh model yang terbaik dalam kasus mengklasifikasi citra sepatu lari berdasarkan permukaan lintasan yaitu ukuran citra berukuran 128x128, dengan ukuran kernel 2x2, jumlah *epoch* sebesar 80, jumlah *learning rate* sebesar 0,0001, dengan 4 layer konvolusi serta 1 *hidden layer*.
2. Akurasi yang diperoleh dari model menggunakan data *test* adalah 0,844 dengan loss sebesar 0,376.

Saran yang dapat diberikan pada penelitian ini adalah sebagai berikut:

1. Jika pembaca tertarik untuk melakukan penelitian serupa, saran dari peneliti bisa menggunakan algoritma klasifikasi citra lainnya untuk pembanding seperti menggunakan *transfer learning*, SVM, serta KNN.
2. Datasetnya bisa menggunakan jenis sepatu lainnya, seperti jenis sepatu futsal dan bola.

DAFTAR RUJUKAN

As, A. H., Ja, W., & Rahman, M. F. (2020). Jurnal Teknologi dan Manajemen Informatika Penentuan Metode Terbaik Dalam Menentukan Jenis Pohon Pisang Menurut Tekstur Daun (Metode K-NN dan SVM). *Jurnal Teknologi dan Manajemen Informatika*, *6*(2), 128–136.

Firdaus, dr. Y. (2022). *Tips dan Trik Memilih Sepatu Lari yang Tepat*. hellosehat.com. https://hellosehat.com/kebugaran/olahraga-lainnya/tips-memilih-sepatu-lari-yang-sehat/

Naufal, M. F. (2021). Comparative Analysis of Image Classification Algorithm for. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *8*(2), 311–318. https://doi.org/10.25126/jtiik.202184553

Rivki, M., & Bachtiar, A. M. (2017). Implementasi Algoritma K-Nearest Neighbor Dalam Pengklasifikasian Follower Twitter Yang Menggunakan Bahasa Indonesia. *Jurnal Sistem Informasi*, *13*(1), 31. https://doi.org/10.21609/jsi.v13i1.500

Suartika E. P, I Wayan, Wijaya Arya Yudhi, S. R. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101. *Jurnal Teknik ITS*, *5*(1), 76. http://repository.its.ac.id/48842/

Surya, G. (2022). *Berlari Memakai Sepatu atau Tanpa Alas Kaki, Mana yang Lebih Baik?* www.kompas.tv. https://www.kompas.tv/article/252759/berlari-memakai-sepatu-atau-tanpa-alas-kaki-mana-yang-lebih-baik

Tsaqofi, I. A. (2022). Ketahui 8 Risiko Salah Pilih Sepatu Olahraga: Cedera Kaki hingga Pengapuran Sendi. *Voi*. https://voi.id/olahraga/240365/risiko-salah-pilih-sepatu-olahraga

Wulandari, I., Yasin, H., & Widiharih, T. (2020). Klasifikasi Citra Digital Bumbu Dan Rempah Dengan Algoritma Convolutional Neural Network (Cnn). *Jurnal Gaussian*, *9*(3), 273–282. https://doi.org/10.14710/j.gauss.v9i3.27416