

# Implementation of Grid Search Method for Optimization in Convolutional Neural network Model

Penerapan Grid Search Method Untuk Optimasi pada Model Convolutional Neural network

Leonardo D. Wowor

Dept. of Electrical Engineering, Sam Ratulangi University Manado, Kampus Bahu St., 95115, Indonesia

e-mails : [leonardowowor239@gmail.com](mailto:leonardowowor239@gmail.com)

Received: [date]; revised: [date]; accepted: [date]

**Abstract** — *This research is motivated by the difficulty of finding hyperparameters in a Convolutional Neural network model. CNN models typically undergo a process of searching for several hyperparameters, known as Hyperparameter Tuning. Manual hyperparameter tuning, done one by one until an optimal combination is found with the best accuracy, takes a long time and can be quite inefficient. Therefore, this research applies a hyperparameter tuning Method to a CNN model for testing. The hyperparameter tuning Method used is the Grid Search Method. By using Grid search, the process of searching for hyperparameter combinations is expected to be faster and more optimal with the best accuracy. As a result, the CNN model created in the first trial achieved an accuracy rate of 99%. It can be concluded that the Grid Search Method works well with the model created, making the process of building the CNN model faster with just one hyperparameter tuning process.*

**Key words** — *Convolutional Neural network; Grid Search Method; Machine Learning; Optimization.*

**Abstrak** — Penelitian ini di latar belakang oleh permasalahan dalam sulitnya mencari *hyperparameter* pada suatu model *Convolutional Neural network*. Model *Convolutional Neural network* umumnya melalui proses pencarian beberapa *hyperparameter* atau disebut *Hyperparameter tuning*. *Hyperparameter tuning* yang dilakukan secara manual atau satu persatu hingga menemukan kombinasi *hyperparameter* yang optimal dengan akurasi terbaik, akan memakan waktu yang panjang dan cukup lama sehingga membuat proses menentukan *Hyperparameter* menjadi kurang efektif. Untuk itu, penelitian ini dilakukan menerapkan suatu metode *hyperparameter tuning* ke dalam model *Convolutional Neural network* yang akan diuji. Metode *Hyperparameter Tuning* yang digunakan adalah metode *Grid search*. Dengan menggunakan *Grid search* maka proses ada pencarian kombinasi *Hyperparameter* diharapkan akan menjadi lebih cepat dan optimal dengan akurasi terbaik. Hasilnya, model *Convolutional Neural network* yang dibuat dalam uji coba pertama mendapatkan tingkat akurasi 99 persen. Dengan demikian dapat disimpulkan bahwa metode *Grid search* mampu berjalan dengan baik pada model yang dibuat, sehingga proses pembuatan model *Convolutional Neural network* menjadi lebih cepat dengan hanya sekali proses *Hyperparameter tuning*.

**Kata kunci** — *Convolutional Neural network; Grid Search Method; Machine Learning; Optimasi*

## I. PENDAHULUAN

Pemanfaatan teknologi *Machine Learning* berupa metode *Convolutional Neural network* (CNN) yang termasuk salah satu konsep dari deep learning yang dapat mengidentifikasi bentuk objek khususnya citra atau gambar, telah berkembang dan mengalami kemajuan yang signifikan. Pemanfaatan CNN bisa digunakan untuk beberapa masalah seperti *image classification*, *object detection*, *object localization*, dan *image segmentation*. Penelitian yang telah dilakukan sebelumnya mengatakan bahwa tingkat keberhasilan dari metode CNN sangat baik dengan tingkatan akurasi yang tinggi terhadap bermacam jenis citra yang diberikan.

Model dengan hasil prediksi yang akurat bisa dilakukan dengan berbagai macam cara salah satunya dengan *hyperparameter tuning* yang merupakan arsitektur dari deep learning untuk meningkatkan performa dari model prediksi. *Hyperparameter tuning* memiliki peran yang sangat penting dalam mengoptimalkan kinerja dari algoritma *Machine Learning*.

Dalam pengembangan model pembelajaran mesin, salah satu aspek yang sangat krusial adalah penyediaan *hyperparameter*, yang merupakan proses menentukan nilai-nilai parameter yang tidak dipelajari secara langsung selama pelatihan model, melainkan diatur sebelum pelatihan dimulai. *Hyperparameter tuning* memainkan peran penting dalam meningkatkan performa dan akurasi model, terutama dalam konteks algoritma kompleks seperti *Convolutional Neural networks* (CNN).

Menentukan *hyperparameter* yang tepat seringkali merupakan tantangan besar dalam proses pengembangan model. Salah satu permasalahan utama adalah ruang pencarian yang sangat besar dan kompleks. Banyak model pembelajaran mesin memiliki beberapa *hyperparameter* yang masing-masing dapat memiliki rentang nilai yang luas, sehingga menghasilkan ruang pencarian yang eksponensial. Hal ini membuat pencarian nilai optimal menjadi sangat memakan waktu dan memerlukan sumber daya komputasi yang besar.

Selain itu, interaksi kompleks antara berbagai *hyperparameter* dapat membuatnya sulit untuk memahami bagaimana perubahan pada satu *hyperparameter* dapat

mempengaruhi kinerja model secara keseluruhan. Misalnya, kombinasi dari nilai-nilai *hyperparameter* yang berbeda mungkin memiliki efek yang tidak linier dan saling berinteraksi, sehingga sulit untuk menemukan konfigurasi yang optimal melalui metode pencarian sederhana.

Permasalahan lainnya adalah adanya *trade-off* antara berbagai aspek kinerja model, seperti akurasi, kecepatan pelatihan, dan penggunaan memori. Menemukan keseimbangan yang optimal antara faktor-faktor ini seringkali memerlukan eksperimen yang ekstensif dan pemahaman mendalam tentang bagaimana setiap *hyperparameter* mempengaruhi model. Berdasarkan permasalahan tersebut dibutuhkan penelitian untuk mengkaji lebih lanjut tentang *hyperparameter tuning*. Metode yang populer untuk melakukan *hyperparameter tuning* diantaranya adalah *Grid Search Method*. Metode *Grid search* masih menjadi solusi umum daripada harus mengubah satu *hyperparameter* pada satu waktu dan mengukur pengaruhnya terhadap kinerja model, justru hal tersebut akan menjadi tidak efisien dan tidak menjamin hasil yang optimal karena mengabaikan interaksi antara *hyperparameter*.

#### A. Convolutional Neural Network

*Convolutional Neural network* (CNN) merupakan salah satu pengembangan dari *Artificial Neural network* (ANN). Jaringan saraf konvolusional adalah jaringan canggih yang diterapkan pada banyak tugas seperti klasifikasi gambar, deteksi objek, dan pengenalan wajah. Salah satu keuntungan dari CNN yaitu mengurangi jumlah parameter dalam *Neural network*. Secara umum, arsitektur CNN terdiri dari 2 tahap. Pertama, tahap ekstraksi fitur (*feature extraction/feature learning*). Pada tahap ini biasanya mencakup operasi konvolusi, fungsi aktivasi (ReLU), dan pooling. Pada tahap ini akan dihasilkan parameter berupa angka-angka hasil representasi dari operasi-operasi yang dilakukan. Kedua, tahap klasifikasi. Hasil dari ekstraksi fitur akan dilakukan klasifikasi melalui full connected layer [1].

#### B. Grid Search Method

*Grid Search Method* merujuk pada Teknik yang digunakan untuk menemukan *hyperparameter* yang optimal untuk sebuah model. *Grid search* adalah metode yang digunakan untuk mencari parameter yang tepat untuk meningkatkan performa model dengan mencoba seluruh kombinasi *hyperparameter* yang diberikan. Dengan kata lain *Grid Search Method* adalah Teknik *tuning* yang mencoba untuk menemukan nilai optimal dari *hyperparameter*. *Grid search* juga menawarkan kemampuan untuk melakukan *cross-validation*. Langkah ini bertujuan untuk menilai kinerja model dengan cara yang lebih umum, mengingat kinerja model dapat bervariasi tergantung pada bagaimana data dibagi [2].

#### C. Hyperparameter

*Hyperparameter* adalah parameter yang tidak ditentukan oleh proses pembelajaran otomatis model, tetapi harus ditentukan oleh pengguna sebelum proses pelatihan dimulai. Pemilihan *hyperparameter* yang tepat dapat memiliki dampak signifikan pada kinerja dan konvergensi model *Convolutional*

*Neural network* (CNN). Berikut adalah beberapa *hyperparameter* umum dalam model CNN.

#### D. Hyperparameter tuning

*Hyperparameter tuning* adalah proses mencari kombinasi optimal dari *hyperparameter* untuk meningkatkan kinerja dan generalisasi model. Tujuan dari *hyperparameter tuning* adalah untuk menemukan konfigurasi *hyperparameter* yang paling cocok untuk dataset dan tugas yang spesifik.

#### E. Python

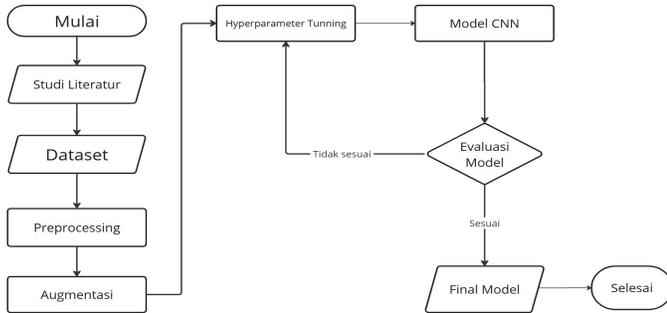
*Python* adalah bahasa pemrograman tujuan umum yang ditafsirkan, tingkat tinggi. Dibuat oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, filosofi desain *Python* menekankan keterbacaan kode dengan penggunaan spasi putih yang signifikan. Konstruksi bahasanya dan pendekatan berorientasi objek bertujuan untuk membantu pemrogram menulis kode yang jelas dan logis untuk proyek skala kecil dan besar. *Python* diketik secara dinamis dan pengumpulan sampah. Ini mendukung beberapa paradigma pemrograman, termasuk pemrograman terstruktur, berorientasi objek, dan fungsional. *Python* dibuat pada akhir 1980-an sebagai penerus bahasa ABC. *Python 2.0*, dirilis pada tahun 2000, memperkenalkan fitur-fitur seperti pemahaman daftar dan sistem pengumpulan sampah dengan penghitungan referensi. *Python 3.0*, dirilis pada tahun 2008, adalah revisi utama dari bahasa yang tidak sepenuhnya kompatibel dengan versi sebelumnya, dan banyak kode *Python 2* yang tidak berjalan tanpa modifikasi pada *Python 3*.

#### F. Google Colab

*Google Collaboratory* atau Google Colab adalah platform berbasis *cloud* untuk menulis, menjalankan, dan berbagi kode *Python* melalui web browser. Platform ini dirancang bagi analyst, developer, peneliti, dan pendidik yang bekerja di bidang data *science* dan *Machine Learning* dengan menyediakan environment komputasi yang fleksibel dan mudah diakses tanpa biaya. Google Colab juga menawarkan kemampuan untuk menjalankan Jupyter Notebook (web *app open-source* untuk kombinasi kode, teks terformat, dan visualisasi data) langsung dari web browser tanpa perlu konfigurasi apa pun. Salah satu fitur utama dari Google Colab adalah kemampuannya untuk berkolaborasi secara real-time. Mirip dengan berbagi dokumen di *Google Docs*, *user* dapat berbagi notebook mereka dengan orang lain. Selain itu, karena notebook disimpan di Google Drive, file ini diakses dari perangkat mana pun dan kapan pun, mirip dengan dokumen lain di ekosistem Google. Dalam konteks analisis data, Google Colab memiliki akses ke *library Python* yang sudah terinstal seperti *TensorFlow*, *PyTorch*, dan *OpenCV*, memungkinkan *analyst* mengembangkan dan menjalankan model *Machine Learning* lebih cepat dan efisien.

## II. METODE

### A. Diagram Alur Penelitian



Gambar 1. Diagram Alur Penelitian

#### 1. Studi Literatur

Studi literatur adalah upaya penulis untuk menyelidiki berbagai referensi yang relevan, baik berupa buku maupun karya ilmiah, yang memiliki keterkaitan atau relevansi dengan topik penelitian yang sedang dijalankan. Dalam tahapan ini, peneliti mencari, meninjau, dan menganalisis beragam sumber literatur guna memperoleh pemahaman yang lebih baik tentang konteks, kerangka kerja, temuan sebelumnya, dan kontribusi penelitian yang telah dilakukan oleh orang lain dalam bidang yang sama atau terkait. Berikut adalah ringkasan dari penelitian-penelitian relevan yang telah dilakukan sebelumnya mengenai *hyperparameter tuning* yang akan digunakan sebagai referensi utama dalam studi ini.

Optimasi *Hyperparameter Convolutional Neural network* Klasifikasi Citra Bunga Dengan Menggunakan Algoritma *Grid search*. Persamaannya adalah algoritma model yang digunakan yaitu *Convolutional Neural network* dan juga metode *Grid search* yang digunakan untuk melakukan *hyperparameter tuning* pada model tersebut. Sedangkan perbedaannya adalah dataset yang digunakan yaitu gambar bunga [3].

*Hyperparameter Tuning* pada Algoritma Klasifikasi dengan *Grid search*. Berdasarkan penelitian yang dilakukan menggunakan dataset UCI Pima Indian Diabetes bisa disimpulkan bahwa optimasi dengan tuning *hyperparameter* menggunakan *Grid search* terhadap model *Machine Learning* membuat proses pemilihan model menjadi lebih mudah. Hasil eksperimen menunjukkan bahwa model memperoleh nilai terbaik yaitu sebesar 0,772 sedangkan *Decision tree* memiliki nilai terendah yaitu 0,701 [4].

Peningkatan Kinerja Akurasi Prediksi Penyakit Diabetes Mellitus Menggunakan Metode *Grid Search* pada Algoritma Logistic Regression. Persamaannya adalah penelitian ini menggunakan metode *Grid search* untuk melakukan *hyperparameter tuning*, sedangkan perbedaannya adalah penelitian ini menggunakan algoritma Logistic Regression [5].

Perbandingan Optimasi Metode *Grid search* dan *Random search* dalam Algoritma XGBoost untuk Klasifikasi Stunting. Persamaannya adalah penelitian ini menggunakan metode *Grid*

*search* untuk melakukan tuning pada setiap Hyperparameter, sedangkan perbedaannya penelitian ini menggunakan algoritma XGBoost. Selain itu juga dipakai metode *Random search* untuk dilakukan perbandingan performa antara kedua metode tersebut [6].

Menentukan Nilai K Pada Metode *K-Means* Menggunakan Teknik *Grid search* Untuk Strategi Produk Pakaian Medis. Persamaannya penelitian ini menggunakan *Grid search* untuk mencari parameter yang tepat pada model yang digunakan, sedangkan perbedaannya adalah penelitian ini menggunakan algoritma *k-means* yang diuji coba dengan menggunakan *Grid search* untuk menentukan *hyperparameter* yang tepat pada model yang digunakan [7].

Klasifikasi Penyakit Diabetes Retinopati Menggunakan *Support vector machine* dengan Algoritma *Grid search* Cross-validation. Persamaannya penelitian ini menggunakan metode *Grid search*, sedangkan perbedaannya adalah penelitian ini menggunakan algoritma *support vector machine* (SVM) yang diterapkan untuk menklasifikasi penyakit diabetes retinopati [8].

Optimasi Parameter Support Vector Machine Berbasis Algoritma *Firefly* Pada Data Opini Film. Algoritma yang digunakan pada penelitian ini adalah Support Vector Machine dan metode *hyperparameter tuning* yang digunakan adalah *Firefly* [9].

#### 2. Dataset

Data yang digunakan dalam penelitian ini adalah dataset MNIST. MNIST adalah *dataset* berisi gambar angka tulisan tangan yang umumnya digunakan untuk melatih berbagai sistem pemrosesan gambar. Selanjutnya, dilakukan analisis dan eksplorasi data dengan menggunakan teknik visualisasi dan statistik deskriptif untuk memahami karakteristik dataset yang dimiliki. Isi dari dataset dapat dilihat pada gambar 2.



Gambar 2. Sampel dari dataset MNIST

Proses pengembangan model dimulai dengan mengunduh dataset dari sumber *National Institute of Standards and Technology database*[9]. Pada gambar 3 akan di tampilkan proses pengambilan data dengan menggunakan fungsi *load*.

Load dataset: MNIST

```
[ ] # loading data  
(train_data, train_labels), (test_data, test_labels) = mnist.load_data()
```

Gambar 3. Pengambilan data

### 3. Preprocessing Data

Langkah berikutnya adalah melakukan *preprocessing* data, yang meliputi normalisasi, *reshape*, dan pembagian dataset menjadi data pelatihan dan data validasi. Dengan melakukan serangkaian langkah ini secara sistematis, dapat dipastikan bahwa data yang akan digunakan untuk melatih model CNN telah dipersiapkan dengan baik, sehingga memungkinkan model untuk belajar secara efektif dari informasi yang terkandung dalam dataset.

```
X_train, X_test, y_train, y_test = preprocessing_data(train_data, test_data, train_labels, test_labels)  
  
X_train shape: (60000, 28, 28, 1)  
X_test shape: (10000, 28, 28, 1)  
y_train shape: (60000,)  
y_test shape: (10000,)  
-----  
Before one-hot encoding: 5  
After one-hot encoding: [0. 0. 0. 0. 1. 0. 0. 0. 0.]
```

Gambar 4. Persiapan data

### 4. Hyperparameter Tuning

Proses pelatihan model *Convolutional Neural network* (CNN) dilakukan dengan menerapkan teknik *Grid search* guna mencari kombinasi *hyperparameter* yang optimal. Selama proses ini, dilakukan evaluasi performa model CNN dengan memanfaatkan metrik-metrik penting seperti akurasi, presisi, recall, dan F1 score. Dengan menggunakan pendekatan ini, dapat memastikan bahwa model yang dihasilkan tidak hanya memiliki kinerja yang tinggi dalam memprediksi kelas target, tetapi juga memiliki kemampuan yang baik dalam mengukur performa secara menyeluruh, sehingga dapat diandalkan dalam menangani berbagai tugas pemrosesan gambar dan klasifikasi. Pada tabel 1 akan ditampilkan *hyperparameter* yang diuji cobakan.

### 5. Model CNN

Setelah tahap *preprocessing* data selesai dilakukan, langkah berikutnya adalah pembuatan model *Convolutional Neural network* (CNN) untuk melakukan pembelajaran dari dataset yang telah disiapkan. Pada tahap ini, dilakukan penentuan *hyperparameter* model yang akan dioptimasi menggunakan teknik *Grid search*. Dengan memilih kombinasi *hyperparameter* yang optimal, dapat meningkatkan performa model CNN dan menghasilkan hasil prediksi yang lebih akurat dan dapat diandalkan. Proses ini merupakan langkah penting dalam pengembangan model CNN yang efektif dan dapat menghasilkan solusi yang tepat untuk permasalahan yang sedang dihadapi.

### 6. Analisis Hasil

Setelah melalui proses pelatihan model *Convolutional Neural network* (CNN) dengan menggunakan teknik *Grid search*, langkah selanjutnya melakukan analisis terhadap hasil yang diperoleh dan membandingkannya dengan hasil pelatihan model CNN tanpa penggunaan *Grid search*. Dalam proses ini, dilakukan interpretasi terhadap perbedaan kinerja model, termasuk akurasi, presisi, recall, dan F1 score, antara kedua

pendekatan tersebut. Berdasarkan hasil analisis tersebut, dapat diambil kesimpulan mengenai efektivitas *Grid search* dalam mengoptimalkan *hyperparameter* model CNN dan dampaknya terhadap kinerja keseluruhan model. Informasi ini akan membantu dalam mengambil keputusan terkait dengan pemilihan metode yang paling sesuai untuk pengembangan model CNN di masa depan serta memberikan wawasan yang berharga dalam memahami bagaimana faktor-faktor tertentu memengaruhi kualitas model. .

### 7. Penulisan Laporan

Penulisan laporan skripsi melibatkan beberapa tahapan yang terstruktur dengan baik untuk memastikan kelengkapan dan kualitas laporan. Tahapan-tahapan tersebut mencakup pendahuluan, tinjauan pustaka, metodologi, hasil dan analisis, kesimpulan, dan saran. Pendahuluan memperkenalkan pembaca dengan topik penelitian dan tujuan dari penelitian tersebut. Tinjauan pustaka mengevaluasi literatur terdahulu yang relevan dengan topik penelitian untuk mendukung penelitian yang dilakukan. Metodologi menjelaskan secara rinci langkah-langkah yang diambil dalam proses penelitian, termasuk desain penelitian, teknik pengumpulan data, dan metode analisis yang digunakan. Hasil dan analisis menyajikan temuan utama dari penelitian beserta interpretasi dan analisis mendalam terhadap data yang diperoleh. Kesimpulan menyajikan ringkasan dari hasil penelitian dan implikasinya, sementara saran mengidentifikasi area penelitian yang mungkin dieksplorasi lebih lanjut di masa mendatang. Memastikan bahwa laporan mematuhi aturan format dan tata cara penulisan skripsi yang berlaku penting untuk memastikan kesesuaian dan kualitas laporan secara keseluruhan.

## III. HASIL DAN PEMBAHASAN

### A. Percobaan Pada Model Pertama

#### 1. Hyperparameter yang diujicobakan

TABEL 1  
HYPERPARAMETER YANG DIUJI COBAKAN

No	Hyperparameter	Value
1	Activation function	Sigmoid, ReLU, LeakyReLU, PReLU, tanh
2	Loss function	MSE, Cross-entropy
3	Batch size	8, 32, 128
4	Epoch	5
5	Optimizer	Adam

Tabel 1 memuat rangkaian *hyperparameter* yang diujicobakan untuk mencapai model terbaik. *Hyperparameter* tersebut meliputi *activation function* (Sigmoid, ReLU, LeakyReLU, PReLU, tanh), *loss function* (Mean Squared Error dan Cross-Entropy), batch size (8, 32, dan 128), epoch (5), serta *optimizer* (Adam). Kombinasi nilai-nilai tersebut akan dioptimalkan untuk mencapai hasil terbaik. Analisis ini bertujuan untuk menemukan konfigurasi *hyperparameter* yang optimal dan meningkatkan kinerja model.

2. Hasil

Hasil proses *hyperparameter tuning* menunjukkan bahwa kombinasi optimal dari *hyperparameter* tersebut tercantum pada Tabel 2, memperlihatkan konfigurasi terbaik untuk mencapai kinerja model yang maksimal. Dan hasil terbaik yang didapatkan seperti *accuracy*, *precision*, *recall*, serta *f1-score* dapat dilihat pada tabel 3.

TABEL 2  
HYPERPARAMETER TERBAIK

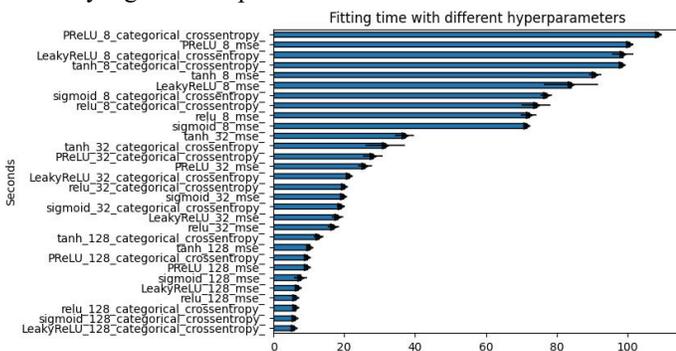
No	Hyperparameter	Value
1	Activation function	PRELU
2	Loss function	Categorical Crossentropy
3	Batch size	8
4	Epoch	5
5	Optimizer	Adam

TABEL 3  
HASIL YANG DICAPAI

No	Metrik Evaluasi	Hasil
1	Accuracy	92.71000000000001
2	Precision	92.86
3	Recall	92.71
4	F1-Score	92.73

3. Visualisasi Hasil Pencarian Grid Search

Fungsi `plot_grid_results` dari library `sklearn` digunakan untuk memvisualisasikan hasil pencarian *hyperparameter* menggunakan `Grid Search`. Fungsi ini memungkinkan peneliti melihat secara langsung nilai parameter terbaik, skor (seperti akurasi) untuk setiap kombinasi parameter, serta waktu pelatihan model. Dengan demikian, peneliti dapat menganalisis dan membandingkan hasil pencarian *hyperparameter* dengan lebih mudah dan efektif. Seperti yang dapat dilihat pada gambar 3, kita dapat melihat kombinasi *hyperparameter* terbaik yang bisa dicapai model.



Gambar 3. Visualisasi hasil proses *Grid search*

B. Percobaan Pada Model Kedua

1. *Hyperparameter* yang diujicobakan

TABEL 4  
HYPERPARAMETER YANG DIUJI COBAKAN

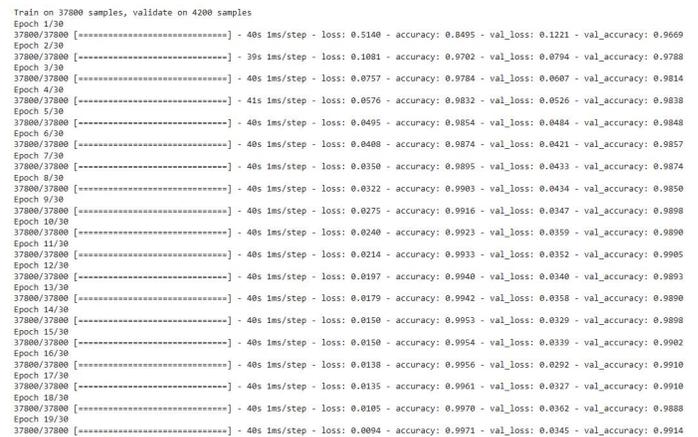
No	Hyperparameter	Value
1	Batch Size	16, 32, 64
2	Epoch	n_epoch
3	Dropout Rate	0.0, 0.10, 0.20, 0.30

Pada pengujian kedua digunakan model CNN yang berbeda atau arsitektur CNN yang digunakan berbeda dari arsitektur yang digunakan pada model pertama. Tabel 4 memuat rangkaian *hyperparameter* yang diujicobakan untuk mencapai model terbaik. *Hyperparameter* tersebut meliputi *batch size* (16, 32, 64), *epoch* (n\_epoch), serta *dropout rate* (0.0, 0.10, 0.20, 0.30). Kombinasi nilai-nilai tersebut akan dioptimalkan untuk mencapai hasil terbaik. Analisis ini bertujuan untuk menemukan konfigurasi *hyperparameter* yang optimal dan meningkatkan kinerja model.

2. Hasil Pengujian

Pada hasil pengujian ini akan dilakukan dengan cara yang berbeda dan juga teknik visualisasi yang berbeda. Proses pengujian akan dilakukan dengan cara membandingkan model yang menggunakan metode *Grid Search* dan yang tidak menggunakan metode *Grid Search*,

1) Sebelum Menggunakan *Grid Search*



Gambar 4. Proses train model CNN

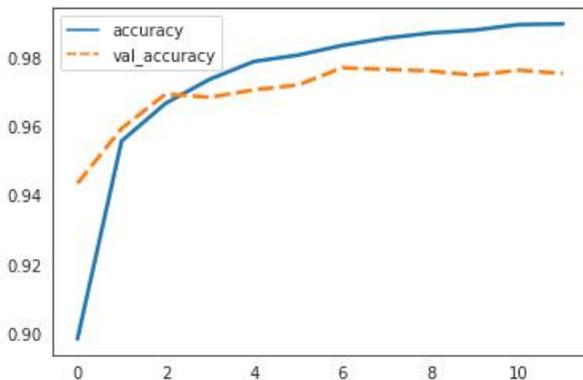
Berdasarkan gambar 2 kinerja model dievaluasi berdasarkan akurasi terbaik yang dicapai pada set validasi, yakni sekitar 0.975 (tegantung pada eksekusi yang dilakukan). Ketika mengukur kinerja suatu model *Machine Learning*, sangat penting untuk memperhatikan bagaimana model tersebut berperilaku pada data yang tidak digunakan selama proses pelatihan, yang disebut sebagai set validasi. Dalam kasus ini, kinerja model dinilai berdasarkan akurasi terbaik yang berhasil dicapai pada set validasi, yakni sekitar 0.975. Angka ini menunjukkan bahwa model kita mampu mengenali kelas dengan benar sekitar 97.5% dari waktu ketika diuji pada data baru yang tidak pernah dilihat sebelumnya. Ini berarti bahwa ketika model diuji pada data yang tidak pernah dilihat sebelumnya (set validasi), rata-rata model berhasil mengenali kelas dengan benar sekitar 97.5%.

Hasil terbaik sudah dicapai setelah sekitar 10 iterasi (atau "epoch") dari proses pelatihan model. Ini mengindikasikan bahwa setelah melatih model dengan data pelatihan sebanyak 10 kali, peningkatan tambahan dalam jumlah iterasi mungkin tidak akan memberikan peningkatan signifikan dalam kinerja model. Dengan kata lain, tambahan iterasi mungkin hanya membuang-buang waktu tanpa memberikan manfaat yang signifikan.

Selain itu, terdapat tanda-tanda yang jelas dari "Overfitting". *Overfitting* terjadi ketika model terlalu "menghafal" data pelatihan dan tidak dapat menggeneralisasi dengan baik pada data baru. Salah satu tanda *Overfitting* adalah perbedaan besar antara kesalahan (loss) pada set validasi dan pada set pelatihan. Dalam kasus ini, kerugian pada set validasi (sekitar 0.1) sekitar 10 kali lipat lebih besar daripada kerugian pada set pelatihan (sekitar 0.01). Hal ini menunjukkan bahwa model mungkin "terlalu cocok" pada data pelatihan, yang mengakibatkan kinerja yang buruk pada data yang belum pernah dilihat sebelumnya. Oleh karena itu, untuk mengatasi *Overfitting*, dapat mengambil berbagai langkah, seperti menggunakan teknik regularisasi, pengurangan kompleksitas model, atau mengumpulkan lebih banyak data pelatihan jika memungkinkan. Dengan memahami perilaku model secara menyeluruh.

## 2) Grafik Tren Akurasi Sebelum Grid search

Pada gambar 5, akan diperlihatkan sebuah grafik yang menampilkan tren akurasi dari data pelatihan dan data validasi seiring dengan peningkatan jumlah iterasi pada proses pelatihan model. Grafik ini akan membantu memvisualisasikan bagaimana akurasi model berubah seiring waktu pelatihan.



Gambar 5. Grafik tren akurasi sebelum *Grid search*

Dengan melihat gambar 5, ternyata terdapat perbedaan signifikan antara akurasi pada data pelatihan dan data validasi. Jika terdapat kesenjangan yang besar antara kedua metrik ini, maka itu bisa menjadi indikasi adanya *Overfitting*, di mana model lebih "menghafal" data pelatihan daripada mempelajari pola umum yang berlaku untuk data baru. Sebaliknya, jika kedua metrik tersebut memiliki tren yang serupa dan meningkat secara bersamaan, maka itu menunjukkan bahwa model mungkin mampu menggeneralisasi dengan baik. Dengan demikian, plot akurasi ini memberikan pemahaman yang lebih mendalam tentang kinerja model dan membantu dalam pengambilan keputusan terkait dengan langkah-langkah yang perlu diambil untuk meningkatkan kualitas model secara keseluruhan.

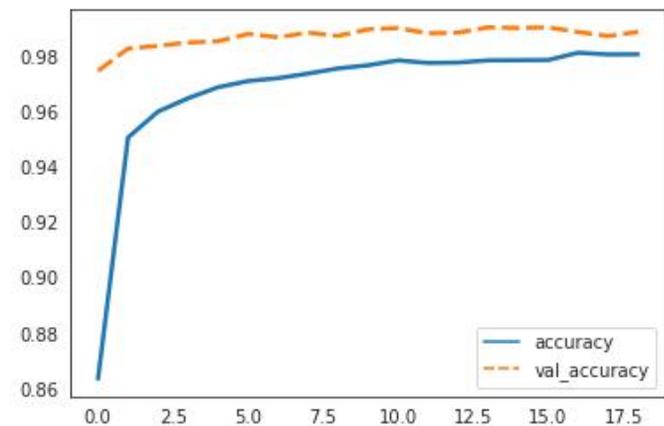
## B. Model CNN Setelah diterapkan Metode Grid search

```
Epoch 1/10
37800/37800 [-----] - 39s 1ms/step - loss: 0.2464 - accuracy: 0.9382
Epoch 2/10
37800/37800 [-----] - 39s 1ms/step - loss: 0.0682 - accuracy: 0.9818
Epoch 3/10
37800/37800 [-----] - 38s 1ms/step - loss: 0.0495 - accuracy: 0.9861
Epoch 4/10
37800/37800 [-----] - 38s 1ms/step - loss: 0.0414 - accuracy: 0.9878
Epoch 5/10
37800/37800 [-----] - 38s 1ms/step - loss: 0.0351 - accuracy: 0.9888
Epoch 6/10
37800/37800 [-----] - 39s 1ms/step - loss: 0.0292 - accuracy: 0.9912
Epoch 7/10
37800/37800 [-----] - 38s 1ms/step - loss: 0.0268 - accuracy: 0.9919
Epoch 8/10
37800/37800 [-----] - 39s 1ms/step - loss: 0.0235 - accuracy: 0.9934
Epoch 9/10
37800/37800 [-----] - 39s 1ms/step - loss: 0.0217 - accuracy: 0.9929
Epoch 10/10
37800/37800 [-----] - 39s 1ms/step - loss: 0.0189 - accuracy: 0.9941
time for grid search = 2974 sec
Best score = 0.9871 using {'conv_activation': 'tanh', 'epochs': 10, 'pool_type': 'max'}
mean test accuracy +/- std = 0.9862 +/- 0.0022 with: {'conv_activation': 'sigmoid', 'epochs': 10, 'pool_type': 'max'}
mean test accuracy +/- std = 0.9841 +/- 0.0017 with: {'conv_activation': 'sigmoid', 'epochs': 10, 'pool_type': 'average'}
mean test accuracy +/- std = 0.9871 +/- 0.0008 with: {'conv_activation': 'tanh', 'epochs': 10, 'pool_type': 'max'}
mean test accuracy +/- std = 0.9849 +/- 0.0036 with: {'conv_activation': 'tanh', 'epochs': 10, 'pool_type': 'average'}
```

Gambar 6. Proses train sesudah *Grid search*

Gambar 6, menunjukkan hasil penerapan metode *Grid search* pada model CNN. Model tersebut mendapatkan akurasi 99 persen pada epoch ke 10 dengan waktu 39 detik per epoch. Akurasi serta waktu yang didapatkan lebih tinggi dan dengan waktu yang lumayan singkat. Dalam sekali pengujian saja model sudah mendapatkan hasil tersebut tanpa perlu melakukan *hyperparameter tuning* secara manual atau mencoba setiap kombinasi *hyperparameter* satu persatu.

## D. Grafik tren akurasi sesudah Grid search



Gambar 7. Grafik tren akurasi sesudah *mGrid search*

Berdasarkan gambar 7, dapat dilihat model yang dihasilkan setelah menerapkan metode *Grid search* sudah tidak mengalami *Overfitting* lagi.

## IV. KESIMPULAN DAN SARAN

### A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan maka mendapatkan kesimpulan sebagai berikut.

1. Metode *Grid search* yang diterapkan pada model *Convolutional Neural network* yang diuji cobakan dapat meningkat sebanyak cukup signifikan. model mendapatkan hasil akurasi terbaik yang tinggi yaitu 99%.
2. Waktu pelatihan model juga dapat dipersingkat dengan mendapatkan 39 detik per epoch.
3. Selain mendapatkan akurasi 99%, metode *Grid search* juga membuat proses *Hyperparameter tuning* menjadi lebih singkat dengan hanya sekali pengujian saja.

4. Namun, untuk membuat model yang optimal dengan performa yang terbaik *hyperparameter tuning* saja tidak bisa menjadi tolak ukur agar suatu model menjadi optimal, tetapi arsitektur dari model juga berpengaruh untuk membuat model menjadi optimal.

#### B. Saran

Saran yang dapat penulis berikan setelah penelitian ini dibuat adalah sebagai berikut

1. Gunakan teknik hyperparameter tuning lainnya (misalnya, *Random Search*, *Bayesian Optimization*) untuk membandingkan hasil.
2. Uji Model CNN dengan dataset yang lebih besar dan beragam untuk meningkatkan generalisasi.
3. Implementasikan teknik regularisasi untuk mencegah overfitting.
4. Bandingkan kinerja Model CNN dengan model lainnya (misalnya, ResNet, *Inception*).
5. Lakukan analisis sensitivitas untuk menentukan pengaruh masing-masing *hyperparameter* terhadap kinerja model.
6. Penelitian inidapat menjadi dasar acuan bagi penelitian selanjutnya yang membahas tentang *Hyperparameter tuning* dan juga peneapan metode *Grid search* pada model dan algoritma yang berbeda dengan hasil dan juga penerapan yang lebih baik juga.

#### V. KUTIPAN

- [1] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017*, Institute of Electrical and Electronics Engineers Inc., Jul. 2017, pp. 1–6. doi: 10.1109/ICEngTechnol.2017.8308186.
- [2] A. C. Müller and S. Guido, "Introduction to Machine Learning with Python A GUIDE FOR DATA SCIENTISTS Introduction to Machine Learning with Python."
- [3] D. A. Wibowo, N. Suciati, and A. Yuniarti, "HYPERPARAMETER OPTIMIZATION OF CONVOLUTIONAL NEURAL NETWORK FOR FLOWER IMAGE CLASSIFICATION USING GRID SEARCH ALGORITHMS," *Jurnal Teknik Informatika (JUTIF)*, vol. 5, no. 1, pp. 313–320, 2024, doi: 10.52436/1.jutif.2024.5.1.1798.
- [4] W. Nugraha and A. Sasongko, "SISTEMASI: Jurnal Sistem Informasi Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search Hyperparameter Tuning on Classification Algorithm with Grid Search." [Online]. Available: <http://sistemasi.ftik.unisi.ac.id>
- [5] "JEPIN (Jurnal Edukasi dan Penelitian Informatika) Peningkatan Kinerja Akurasi Prediksi Penyakit Diabetes Mellitus Menggunakan Metode Grid Search pada Algoritma Logistic Regression".
- [6] N. A. Pramudhyta and M. S. Rohman, "Perbandingan Optimasi Metode Grid Search dan Random Search dalam Algoritma XGBoost untuk Klasifikasi Stunting," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 8, no. 1, p. 19, Jan. 2024, doi: 10.30865/mib.v8i1.6965.

R. Dwirahmanto and A. Bisri, "Menentukan Nilai K Pada Metode K-Means Menggunakan Teknik Grid Search Untuk Strategi Produk Pakaian Medis." [Online]. Available: <https://jurnal.publikasitecno.id/index.php/jim93>

Fatmawati and N. A. K. Rifai, "Klasifikasi Penyakit Diabetes Retinopati Menggunakan Support Vector Machine dengan Algoritma Grid Search Cross-validation," *Jurnal Riset Statistika*, pp. 79–86, Jul. 2023, doi: 10.29313/jrs.v3i1.1945.

Styawati, Andi Nurkholis, Zaenal Abidin, and Heni Sulistiani, "Optimasi Parameter Support Vector Machine Berbasis Algoritma Firefly Pada Data Opini Film," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 5, pp. 904–910, Oct. 2021, doi: 10.29207/resti.v5i5.3380.



**Leonardo D. Wowor.** Lahir di Lowian pada tanggal 9 Agustus 2001. Merupakan anak ketiga dari tiga bersaudara, dengan saudara kandung Serlan Wowor (Kakak) dan Christian Wowor (Kakak) dari 8 orang tua bernama Ronny Wowor (Ayah) dan Telly Manorek (Ibu). Berdomisili di Desa Lowian, Kecamatan Maesaan, Kabupaten Minahasa Selatan, Provinsi Sulawesi Utara. Penulis

memulai Pendidikan pertama di SD Negeri Lowian pada tahun 2007-2013, kemudian melanjutkan Pendidikan di SMP Negeri 3 Maesaan pada tahun 2013-2016 setelah itu melanjutkan Pendidikan di SMA Negeri 1 Maesaan pada tahun 2016-2019. Pada tahun 2019 penulis melanjutkan pendidikan Strata Satu (S1) di salah satu perguruan tinggi di Sulawesi Utara, yaitu Universitas Sam Ratulangi Manado dengan mengambil Program Studi Teknik Informatika Jurusan Teknik Elektro Fakultas Teknik. Selama masa perkuliahan, penulis tergabung dalam organisasi kemahasiswaan, yaitu Himpunan Mahasiswa Elektro (HME) dan penulis juga mengikuti organisasi kerohanian, yaitu Unit Pelayanan Kristen Fakultas Teknik (UPKFT) serta menjadi pengurus pada komunitas belajar yaitu Unsrat IT Community (UNITY). Penulis juga mengikuti organisasi kerohanian gereja PELMAP (Pelayanan Mahasiswa Pantekosta) sejak 2019-sekarang.