

Klasifikasi Spesies Tanaman Kaktus *Grafting* Berdasarkan Citra *Scion* Menggunakan Metode Convolutional Neural Network (Cnn)

Dedih Kurnia¹, Agung Toto Wibowo²

^{1,2} Universitas Telkom, Bandung

¹dedihkurnia@students.telkomuniversity.ac.id, ²agungtoto@telkomuniversity.ac.id

Abstrak

Keunikan dan keanekaragaman tanaman kaktus yang banyak dan tersebar di seluruh dunia dan memiliki umur yang panjang serta perawatan yang sederhana menjadikan tanaman kaktus banyak diminati bagi sebagian besar orang. Dengan kemajuan jaman, metode okulasi atau *Grafting* banyak digunakan oleh petani kaktus hias untuk memperkaya keanekaragaman tanaman kaktus sehingga menghasilkan tanaman kaktus yang berukuran kecil, pertumbuhan tanaman yang lambat, dan memiliki keragaman *Scion* pada kaktus *Grafting*. *Scion* merupakan bagian atas dari susunan tubuh kaktus *Grafting*, bagian *Scion* ini biasanya dipilih dari spesies kaktus lain yang bentuk dan warnanya unik dan mencolok. Hasil okulasi yang menyebabkan keanekaragaman pada *Scion* ini membuat masyarakat umum sulit untuk membedakan tanaman kaktus *Grafting*, dengan adanya sistem yang dapat mengenali dan mengklasifikasikan tanaman kaktus *Grafting* diharapkan akan mempermudah masyarakat umum dalam mengenali tanaman kaktus *Grafting* yang mempunyai ciri-ciri khasnya masing-masing. Sistem ini dibuat dengan metode *Convolutional Neural Network* (CNN) yang dibangun dengan arsitektur RESNET152V2 untuk memvalidasi struktur model CNN, memiliki dataset sebanyak 31.600 data citra tumbuhan bagian atas yang di sebut *Scion* pada 10 spesies tanaman kaktus *Grafting*, yaitu spesies *Enchinopsis Peruviana*, *Gymnocalycium Baldianum*, *Mammillaria Spinosissima cv. Un Pico*, *Opuntia Microdasys*, *Parodia Warasii*, *Astrophytum Capricorne*, *Blossfeldia Liliputana*, *Copiapoa Laui*, *Cephalocereus Senilis*, dan *Echinopsis Chocolate* yang umum dan populer dibudidayakan di pasaran dan menghasilkan model terbaik pada versi 1 sebesar 94.13% pada data Testing Lapangan, dan 97.60% pada data Tesing Internet, kemudian pada versi 2 menghasilkan model terbaik sebesar 82.10% pada data Testing Lapangan, dan 86.86% pada data Tesing Internet.

Kata Kunci: Klasifikasi Kaktus *Grafting*, *Convolutional Neural Network* (CNN), ResNet152V2, *Scion* Kaktus.

Abstract

The uniqueness and diversity of cactus plants are numerous and spread throughout the world and have a long life and simple care that make cactus plants in great demand for most people. With the advancement of time, the *Grafting* method is used by many ornamental cactus farmers to enrich the diversity of cactus plants so that they produce small cactus plants, slow plant growth, and have *Scion* diversity in *Grafting* cactus plants. *Scion* is the upper part of the *Grafting* cactus body structure, this part of the *Scion* is chosen usually from other cactus species whose shape and color are unique and striking. The results of *Grafting* that cause diversity in *Scion* make it difficult for the general public to distinguish *Grafting* cactus plants, with a system that can recognize and classify *Grafting* cactus plants, it is hoped that it will make it easier for the general public to recognize *Grafting* cactus plants that have their own characteristics. This system was made using the *Convolutional Neural Network* (CNN) method which was built with the RESNET152V2 architecture to validate the CNN model structure, has a dataset of 31,600 upper plant image data called *Scion* on 10 *Grafting* cactus plant species, namely *Enchinopsis Peruviana*, *Gymnocalycium Baldianum*, *Mammillaria Spinosissima cv. Un Pico*, *Opuntia Microdasys*, *Parodia Warasii*, *Astrophytum Capricorne*, *Blossfeldia Liliputana*, *Copiapoa Laui*, *Cephalocereus Senilis*, and *Echinopsis Chocolate* are common and popular cultivated in the market and produce the best model in version 1 by 94.13% in field testing data, and 97.60% in internet testing data, then version 2 produces the best model of 82.10% on field testing data, and 86.86% on internet testing data.

Keywords: *Grafting Cactus Classification*, *Convolutional Neural Network* (CNN), ResNet152V2, *Scion Cactus*.

1. Pendahuluan

1.1 Latar Belakang

Kaktus *Grafting* adalah 2 bagian kaktus yang berbeda yang disatukan menjadi 1 tanaman yang disambung dengan metode okulasi. Kaktus *Grafting* adalah salah satu jenis tanaman hias yang kini populer karena selain keindahannya, tanaman kaktus *Grafting* memiliki daya tahan tubuh yang kuat di bandingkan tanaman lainnya. Pada umumnya kaktus menunjukkan tingkat pertumbuhan yang rendah apalagi kaktus dapat beradaptasi dengan baik untuk tumbuh di daerah kering dan semi kering dimana masalah utamanya adalah ketersediaan air[1]. Sekitar 2.500 spesies kaktus tersebar di zona subtropis daerah kering di Amerika Utara dan Amerika Selatan. Kaktus *Grafting* diperkenalkan di Korea oleh kolektor pribadi pada pertengahan 1940-an. Peningkatan pesat dalam produksi kaktus disebabkan oleh okulasi, upaya ini telah mengangkat sekitar 10 *kultivar* setiap tahun dan total 90 *kultivar* telah dilepas sejak tahun 1994, termasuk 42 kultivar *G.mihanovichii*, 11 kultivar *C.silvestrii*, dan 1 dari *G.denudatum* di National Horticultural Research Institute (NHRI)[2].

Kaktus *Grafting* mempunyai nilai jual tinggi karena memiliki keindahan dan ciri karakteristiknya masing-masing[3]. Tumbuhan kaktus *Grafting* memiliki struktur tubuh dalam penamaan bagian-bagiannya yang terdiri dari bagian atasnya yang di sebut *Scion* pada bagian ini dipilih kaktus spesies lain yang bentuknya unik seperti *Gymnocalycium mihanovichii*, *Copiapoa Laui*, *Notocactus Magnificus*, *Opuntia Microdasys* dll. Kemudian pada bagian bawahnya di sebut dengan *stem* biasanya pada bagian bawah menggunakan spesies kaktus kebanyakan berasal dari spesies *Hylocereus undatus* atau sejenis batang pohon buah naga[3].

Sulitnya mengetahui perbedaan spesies pada tanaman kaktus *Grafting* yang semakin banyak karena proses okulasi yang dapat menghasilkan *kultivar* baru setiap tahunnya, maka perlu pembuatan sistem yang dapat mengklasifikasikan tanaman kaktus *Grafting* berdasarkan spesiesnya sehingga masyarakat bisa tahu informasi pada tanaman kaktus *Grafting* pada masing-masing spesiesnya. Pada kasus ini 10 spesies kaktus *Grafting* yang akan diklasifikasikan yaitu, *Enchinopsis Peruviana*, *Gymnocalycium Baldianum*, *Mammillaria Spinosissima cv. Un Pico*, *Opuntia Microdasys*, *Parodia Warasii*, *Astrophytum Capricorne*, *Blossfeldia Liliputana*, *Copiapoa Laui*, *Cephalocereus Senilis*, dan *Echinopsis Chocolate*, 10 spesies ini berdasarkan spesies yang banyak di jual di pasaran dan tempat budidaya kaktus *Grafting* di Lembang.

Semakin berkembangnya zaman, pemanfaatan citra digital mempermudah mengidentifikasi tanaman kaktus *Grafting* berdasarkan karakteristik bentuknya. Untuk mengatasi permasalahan yang disebutkan, klasifikasi spesies kaktus *Grafting* dapat dilakukan melalui citra *Scion* pada tanaman kaktus *Grafting*, citra *Scion* pada tumbuhan kaktus *Grafting* yang nantinya akan diproses menjadi sebuah data yang akan di klasifikasi menggunakan metode *Convolutional Neural Network* (CNN).

Convolutional Neural Network (CNN) banyak digunakan sebagai metode pengenalan data pada suatu citra dalam kinerjanya untuk mengklasifikasikan suatu objek, karena hasil akurasi dan kinerjanya yang tidak diragukan lagi pada penyimpanan informasi spasial dari data citra sehingga menghasilkan hasil klasifikasi yang baik[4]. Selain itu pada penelitian ini menggunakan arsitektur ResNet152V2.

Banyak sekali arsitektur CNN yang biasanya digunakan oleh peneliti untuk kasus pengklasifikasian gambar 2D entah dengan menggunakan arsitektur *Custom*(arsitektur yang dirancang sendiri oleh peneliti) maupun arsitektur yang sudah disediakan oleh para ahli dan sudah teruji melalui beberapa skenario pengujian. Namun karena arsitektur *custom* yang biasanya memiliki tingkat kedalaman lapisan yang lebih ringan dan sederhana dibandingkan arsitektur yang dibuat oleh para ahli sehingga akan mempengaruhi akurasi pengujian pada saat proses klasifikasi, hal ini diperkuat pada penelitian[5] kinerja jaringan akan menurun 2% jika ada satu lapisan konvolusi yang terhapus sehingga tingkat kedalaman lapisan jaringan arsitektur sangat penting untuk mencapai tingkat akurasi dalam kasus klasifikasi.

Arsitektur (*Residual neural network*) ResNet, cukup revolusioner dan arsitektur ini menjadi *state of the art* tidak hanya dalam klasifikasi, namun dalam semua kategori termasuk object detection, dan *semantic segmentation*. ResNet mempunyai kemampuan untuk menghindari masalah pengurangan gradien dan ResNet dengan kedalaman 152 lapisan dan memiliki tingkat kedalaman 20 dan 8 kali lebih dalam dari AlexNet dan VGG sehingga dengan kedalaman lapisan yang dimiliki Resnet152 akan mudah dalam mengklasifikasikan objek 2D dan akan meningkatkan akurasi klasifikasi, selain itu Resnet152 memenangkan kejuaraan 2015-ILSVRC[6].

1.2 Topik dan Batasannya

Berdasarkan deskripsi dari latar belakang yang telah dipaparkan, maka pada penelitian ini membahas topik bagaimana cara membangun sebuah sistem yang mampu mengklasifikasikan tanaman kaktus *Grafting* berdasarkan citra *Scion* pada kaktus *Grafting* dengan mengimplementasikan metode *Convolutional Neural Network* (CNN) dan menganalisis performansi sistem yang telah dibangun.

Berdasarkan identifikasi masalah maka perlu adanya batasan masalah agar ruang lingkup penelitian menjadi jelas, maka batasan pada penelitian ini yaitu sistem yang dibangun hanya dapat mengklasifikasikan 10 citra spesies tanaman kaktus *Grafting*, yaitu *Enchinopsis Peruviana*, *Gymnocalycium Baldianum*, *Mammillaria Spinosissima cv. Un Pico*, *Opuntia Microdasys*, *Parodia Warasii*, *Astrophytum Capricorne*, *Blossfeldia Liliputana*, *Copiapoa*

Laui, Cephalocereus Senilis, dan Echinopsis Chocolate. Citra tanaman kaktus memiliki pencahayaan yang baik, angle pengambilan gambar tidak terlalu bawah.

1.3 Tujuan

Tujuan pada penelitian ini yaitu untuk membangun sebuah model yang dapat mengklasifikasikan tanaman kaktus *Grafting* berdasarkan *Scion* dari spesies tanaman kaktus *Grafting* menggunakan metode *Convolutional Neural Network* (CNN) dan menganalisis performansi model sistem yang dibangun berdasarkan hasil pengujian. Selain itu diharapkan model ini dapat membantu sebagai media pendidikan bagi masyarakat dan petani yang ingin lebih mengenal lagi kaktus *Grafting*.

2. Studi Terkait

Penelitian pada bidang *computer vision*, khususnya pada klasifikasi citra digital untuk secara otomatis mengidentifikasi jenis tanaman pada gambar tertentu sudah banyak sekali yang membahas penelitian ini pada tanaman yang dapat di konsumsi, salah satunya pada penelitian [7] atau pada penelitian identifikasi tanamaman untuk memantau daerah pertanian dengan tujuan memantau perkembangan tanaman seperti pada penelitian[8]. Adapun penelitian lainnya pada bidang pengklasifikasian tanaman yang penulis ketahui sebagai berikut :

Pada penelitian [8] membahas tentang cara mengidentifikasi kaktus pada kawasan lindung dengan tujuan untuk melestarikan kawasan tersebut dengan merancang otonom pengawasan daerah secara otomatis dengan konsep *Machine Learning* untuk mengidentifikasi perkembangan kaktus pada daerah tersebut. Pada penelitian ini peneliti berfokus pada identifikasi urat daun atau pola vena sehingga diyakini memiliki tingkat akurasi tinggi pada hasil identifikasi. Selain itu peneliti menggunakan metode *Convolutional Neural Network* (CNN) dan sebagai data objeknya peneliti menggunakan Dataset kaktus yang diambil langsung di lembah endemik *Tehuacán-Cuicatlán selatan Meksiko* dengan bantuan Drone yang digunakan untuk pengambilan data set pada lembah tersebut. Pada penelitian ini menghasilkan akurasi sebesar 95%.

Pada penelitian [9] membahas tentang bagaimana mengklasifikasikan bibit tanaman dengan tujuan meningkatkan hasil dan produktivitas pada pertanian dan menjadi solusi dalam membantu petani mengoptimalkan dalam penanaman pohon di bidang pertanian. Pada penelitian ini penulis mengidentifikasi bibit *Epigeal* dan *Hypogeal*, penelitian ini dilakukan dengan bantuan metode *Convolutional Neural Network* (CNN) dan sebagai data objeknya peneliti menggunakan Dataset yang diambil dari “*ImagNet*” dan “*Aarhus University Signal, University of Denmark Selatan*”. Dengan kumpulan data yang berisi sekitar 5.000 gambar dengan 960 tumbuhan unik 10 spesies. Sistem yang dibuat oleh penulis menghasilkan 99,48% tingkat keakuratan dalam mengidentifikasi objek pada bibit tanaman.

Pada penelitian [10] membahas bagaimana membuat sistem Smart Drone yang mampu mengidentifikasi jenis dan informasi tentang tanaman yang terinfeksi dengan tujuan sistem akan menampilkan informasi jenis tanaman serta jenis obat dan takaran yang pas untuk diberikan kepada tanaman yang terinfeksi tersebut. Pada penelitian ini penulis mengidentifikasi tanaman yang terinfeksi dengan metode *Convolutional Neural Network* (CNN) dan sebagai data objek klasifikasinya diambil dari Dataset publik. Dataset berisi total 70.295 gambar berlabel RGB. 80% gambar (56.295 gambar) digunakan untuk pelatihan, sedangkan 20% sisanya (14.000 gambar) digunakan untuk pengujian. Penelitian yang dilakukan menghasilkan tingkat akurasi sebesar 99,78%.

Dari berbagai penelitian yang penulis ketahui di atas penelitian pada tanaman hias masih kurang populer dibandingkan tanaman konsumsi, khususnya pada tanaman kaktus *Grafting* yang memiliki informasi sangat minim didunia internet dan memiliki ambiguitas tinggi pada penamaan spesiesnya. Oleh karena itu, penelitian ini akan membahas tentang klasifikasi spesies tanaman kaktus *Grafting*.

2.1 Tanaman Kaktus Grafting

Kaktus merupakan tumbuhan yang beradaptasi dengan baik untuk tumbuh di daerah kering dan semi kering dimana masalah utamanya adalah ketersediaan air, pada umumnya tanaman kaktus menunjukkan tingkat pertumbuhan yang rendah yang membuat perkembangan kaktus menjadi sulit. Kaktus sangat dihargai sebagai tanaman hias karena keragaman bentuknya serta umurnya yang panjang dan indah, kaktus juga merupakan jenis tanaman hias yang paling sering diamati dan diminati[2].

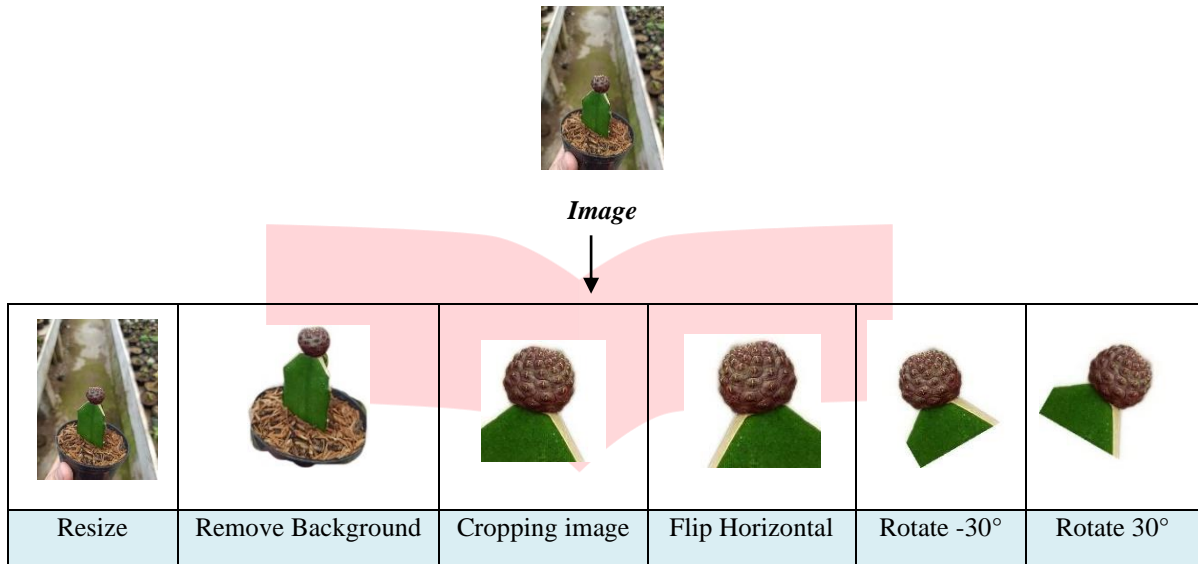
Semakin berkembangnya jaman dan teknologi melahirkan keanekaragaman pada berbagai jenis tanaman termasuk pada tanaman kaktus. Kaktus yang dikembangkan dengan cara diokulasi sekarang dianggap sebagai salah satu tanaman hias dalam ruangan yang paling dikagumi di seluruh dunia. Sekitar 15 juta kaktus yang dicangkok sedang diperdagangkan di pasar internasional, dan 10 juta di antaranya telah dikembangkan dan diproduksi di Korea. Sembilan puluh kultivar baru dengan banyak warna dan bentuk yang beraneka ragam telah dikembangkan oleh *National Horticultural Research Institute* (NHRI), *Rural Development Administration* (RDA), dan *Goyang Cactus Experiment Station*, Provinsi Gyeonggi[2].

2.2 Pengolahan Citra Digital (Image Processing)

Pengolahan citra atau *Image Processing* adalah suatu sistem dimana proses dilakukan dengan masukan (*input*) berupa citra (*image*) dan hasilnya (*output*) juga berupa citra (*image*) hasil pengolahan/modifikasi[11]. Objektivitas

dari *image* preprocessing adalah melakukan transformasi atau analisa suatu gambar sehingga informasi baru tentang gambar dibuat lebih jelas. Gambar digital terdiri dari kumpulan *pixel*. *Pixel* adalah titik yang berisi nilai tertentu yang membentuk sebuah gambar yang lokasinya berada pada koordinat x dan y.

Augmentasi citra adalah salah satu bagian dari pengolahan citra digital, yaitu proses penggandaan pada suatu data dengan melakukan translasi, transformasi, penambahan/pengurangan *noise*, rotasi, pembesaran, atau *Flipping* dari dataset citra[11]. Tujuan utama dari Augmentasi data itu sendiri yaitu untuk mengetahui ciri pada sebuah citra yang optimal yang dimana akan digunakan sebagai analisis, melakukan proses pengambilan informasi pada suatu objek, serta untuk proses kompresi atau reduksi data untuk tujuan penyimpanan data.

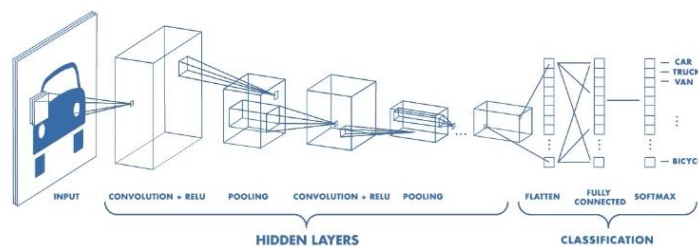


Gambar 1. Ilustrasi dari augmentasi data (Operasi Cropping, Flip, Rotate, dan Resize) yang digunakan pada penelitian[11].

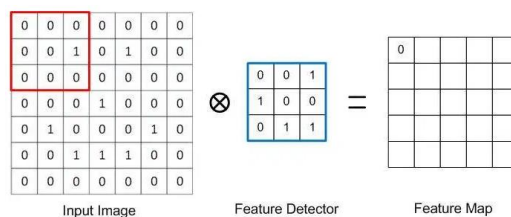
2.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan metode yang sangat menarik dan sering dijadikan metode suatu algoritma untuk pengolahan citra. Klasifikasi objek pada *Image Processing* menjadi suatu proses yang sering digunakan oleh metode ini. Karena kemampuannya yang dapat mengklasifikasi gambar pada data yang besar namun hasil akurasi yang akurat dan memiliki proses yang ringan dalam proses pengklasifikasian sehingga metode ini sering banyak digunakan[12].

Convolutional Neural Network (CNN) dikenal lebih baik dalam kemampuannya menyimpan informasi spasial dari data citra. Ada beberapa tahap dalam pengklasifikasian data dalam CNN, yaitu fungsi Aktivasi (*Softmax*) dan *fully-connected* yang memiliki keluaran berupa klasifikasi. Struktur CNN terdiri dari *Input*, *hidden layer*, proses klasifikasi, dan *output*.



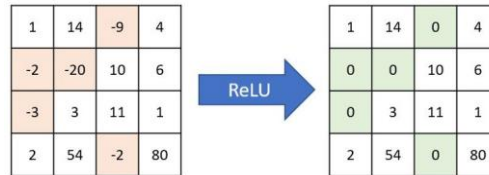
Gambar 2 Sumber Gambar: <https://towardsdatascience.com/>



Gambar 3 Ilustrasi proses konvolusi untuk mendapatkan fitur map[13].

a. Fungsi Aktivasi : ReLU

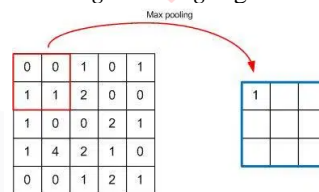
Pada setiap lapisan diberlakukan fungsi aktivasi *Rectified Linear Unit* (ReLU) yang akan menghasilkan nilai nol apabila $x < 0$ dan linier dengan kemiringan 1 ketika $x > 0$. Pada proses ini akan menerima *input* data secara langsung untuk kemudian diproses hingga menghasilkan *output* berupa vektor yang akan diolah pada lapisan berikutnya. Fungsi Aktivasi ReLU (*Rectified Linear Unit*) ini bekerja setelah melakukan proses konvolusi dan mendapatkan fitur pada suatu citra. Aktivasi ReLU digunakan dengan tujuan mengurangi linearitas yang terjadi dari proses konvolusi sehingga CNN lebih mudah mencapai nilai optimum[13].



Gambar 4. Ilustrasi proses fungsi aktivasi Relu[13].

b. Pooling Layer

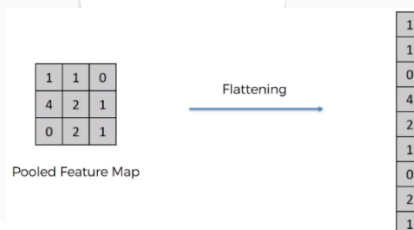
Pooling Layer adalah proses dimana bertujuan untuk meminimalkan ukuran sebuah data citra agar fitur yang di ambil memiliki ciri fitur yang jelas. Pada sebagian besar CNN, subsampling yang digunakan adalah *Max pooling*. *Max pooling* berkerja dengan membagi hasil *output* dari *convolution layer* menjadi beberapa *grid* kecil lalu mengambil nilai maksimal dari setiap *grid* untuk menyusun matriks citra yang telah direduksi[14]. Hal ini dilakukan agar mempercepat komputasi pada saat proses *training* karena bobot yang harus di update semakin sedikit. Tujuan lainnya yaitu agar sistem tetap dapat mendeteksi gambar meskipun terjadi perubahan posisi gambar. Ada dua jenis *pooling layer* yaitu *Max pooling* dan *Average Pooling*. *Max pooling* bertujuan untuk mencari nilai tertinggi sedangkan *Average Pooling* digunakan untuk mencari rata-rata nilai tertinggi[15].



Gambar 5. Ilustrasi proses max pooling dari feature map[16].

c. Flatten Layer

Flatten Layer adalah proses dimana terjadinya perubahan bentuk matrix dari tahap sebelumnya menjadi sebuah vektor(array satu dimensi) hal ini dilakukan agar fitur yang berhasil di kumpulkan bisa masuk ke tahap selanjutnya yaitu tahap *input Fully-Connected Layer* atau pada *layer* klasifikasi.

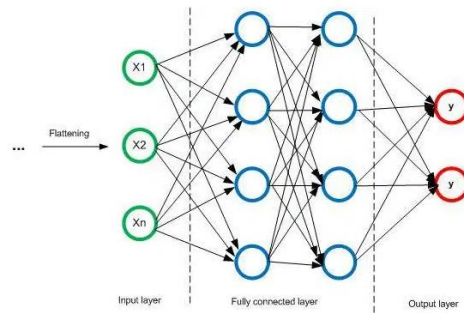


Gambar 6. Ilustrasi proses Flattening

d. Fully Connected Layer

Pada Lapisan *Fully Connected Layer* biasanya digunakan untuk melakukan transformasi pada dimensi data, hal ini bertujuan untuk data dapat diklasifikasikan secara linear. Pada setiap *neuron* pada *convolution layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu agar *neuron* bisa dimasukkan ke dalam *Fully Connected Layer*. Pada tahap perubahan menjadi 1 dimensi, hanya informasi spasial yang penting saja yang masuk kedalam *Fully Connected Layer*. *Fully Connected Layer* hanya diimplementasikan pada akhir jaringan[17].

Pada tahap ini adalah proses dimana mencari klasifikasi sebuah kelas yang paling cocok dari *input* sebelumnya dengan menghasilkan probabilitas tertinggi. *Fully Connected Layer* terdiri dari *Input Layer*, *Hidden Layer* dan *Output Layer*. Pada *layer* ini semua *node (neuron)* dari *layer* sebelumnya terhubung satu sama lain dengan *node* di *layer* selanjutnya. Di setiap *hidden layer* terdapat fungsi aktivasi ReLU yang umum digunakan, dan pada *output layer* terdapat fungsi aktivasi yang umum digunakan yaitu *Softmax* jika dalam kasusnya lebih dari 2 label/kelas/kategori).



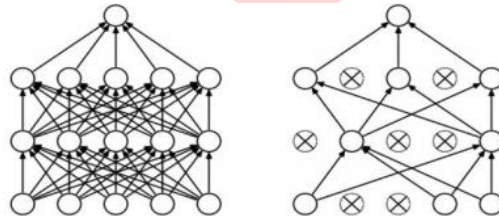
Gambar 7. Ilustrasi Fully-Connected Layer[18].

e. Fungsi Aktivasi: Softmax

Fungsi Aktivasi *Softmax* biasanya digunakan pada kasus klasifikasi lebih dari 2 kelas (*multi-class*), proses ini berlangsung pada *output layer* pada tahapan *Fully Connected/classification*. Tujuan proses Aktivasi *Softmax* untuk menghitung probabilitas pada setiap kelas dan menentukan kelas untuk *input* citra yang diproses sebelumnya. Fungsi Aktivasi *Softmax* menghasilkan nilai antara 0 dan 1.

f. Dropout Regularization

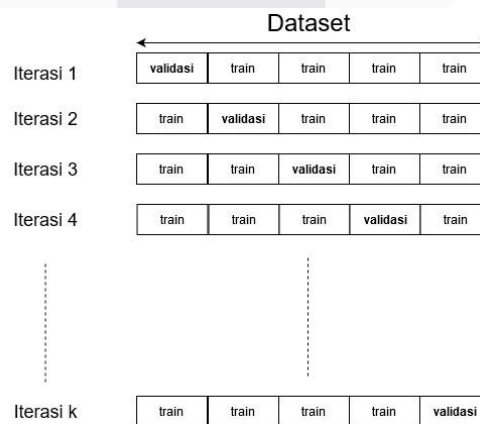
Dropout adalah sebuah teknik untuk meminimalisir terjadinya *overfitting*, yaitu kondisi dimana beberapa *neuron* yang dipilih secara acak tidak akan dipakai selama *training* model. *Neuron-neuron* ini dibuang secara acak sehingga *neuron* yang dibuang akan diberhentikan. Tujuan lain selain untuk meminimalisir *overfitting*, penggunaan *Dropout* pun mempercepat proses pelatihan (*training*)[19].



Gambar 8. Ilustrasi Pengaplikasian Dropout[19].

2.4 Cross Validation : K-Fold

Cross validation adalah sebuah metode statistik yang umum banyak digunakan dengan tujuan mengevaluasi dan membandingkan model yang dibangun. Cara kerja metode statistik ini yaitu dengan cara membagi 2 jenis subset yaitu data *train* dan data *validation*. Data *train* yang digunakan untuk melatih model pembelajaran dan data *validation* yang digunakan untuk memvalidasi model pembelajaran. Data *train* dan *validation* akan disilangkan sebanyak K perulangan, dan setiap perulangan data akan secara terurut tervalidasi sehingga model yang dihasilkan akan berjumlah K model. Pada setiap perulangan dengan jumlah data validasi yang sama namun berbeda data sehingga data validasi pada iterasi sebelumnya akan dijadikan data *train*, kemudian data *train* pada iterasi sebelumnya yang diubah menjadi data validasi. Tujuannya menggunakan metode ini yaitu untuk mengetahui prediksi dari model yang di latih seberapa akurat model yang digunakan[20].



Gambar 9. Ilustrasi K - Fold Cross Validation

Sumber Gambar : <https://satishgunjal.com/kfold>

2.5 Underfitting & Overfitting

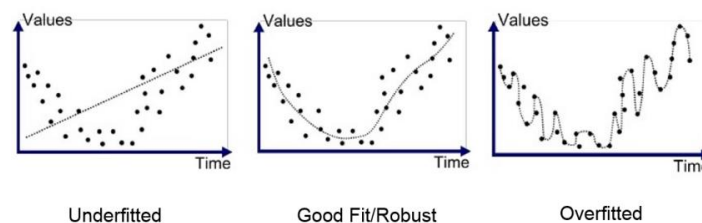
Underfitting dan *Overfitting* model adalah suatu kondisi yang terjadi ketika membuat sebuah model CNN. Model yang *underfitting* atau *overfitting* akan mengalami ketidakmampuan prediksi pada suatu kelas dengan benar [19]. Untuk lebih jelasnya berikut pengertian *Underfitting* dan *Overfitting* :

a. Underfitting

Underfitting adalah suatu kondisi dimana model tidak bisa melihat dan mempelajari logika dibelakang data, sehingga hal ini model tidak bisa melakukan prediksi dengan tepat, baik untuk dataset *training* maupun dataset *testing*. Pada kasus ini *Underfitting* model memiliki kondisi ciri-ciri *loss* tinggi dan akurasi rendah.

b. Overfitting

Overfitting adalah suatu kondisi dimana model yang dibuat terlalu fokus pada *training* dataset tertentu, sehingga hal ini model tidak bisa melakukan prediksi dengan tepat jika diberikan dataset *testing*. Pada kasus ini biasanya kondisi *Overfitting* model akan melihat dan mempelajari data *noise* yang seharusnya diabaikan. *Overfitting* model memiliki kondisi ciri-ciri *loss* rendah dan akurasi rendah.



Gambar 10. Underfitting, Good Fitting dan Overfitting

Sumber Gambar: <https://medium.com/greyatom/>

Model yang baik adalah model yang memiliki *loss* rendah dan akurasi tinggi ketika proses *training* dilakukan. Tetapi tidak menutup kemungkinan dalam pengimplementasian pada CNN *Overfitting* biasa terjadi, untuk mengatasi masalah ini biasanya para ahli melakukan langkah *fine-tuning* atau penyesuaian yang bisa dilakukan, salah satunya yaitu penyesuaian *Hyperparameter* untuk mengatasi masalah *Overfitting* pada pengujian modelnya.

2.6 Hyperparameter

Hyperparameter merupakan sebuah teknik *eksperimental* pada parameter yang nilainya disesuaikan oleh perancang model CNN, hal ini bertujuan agar mendapat nilai *score* akurasi yang lebih baik dan menghindari masalah *overfitting* pada studi kasus yang diangkat. Berikut adalah *hyperparameter* yang bisa disesuaikan oleh perancang model CNN:

a. Input Shape

Sebelum melakukan *input shape* ada baiknya citra yang akan digunakan tidak *blur*, gelap, atau gangguan *noise* lainnya sebelum model CNN mempelajari *features* pada citra gambar hal ini bertujuan agar model CNN tidak mengalami *overfitting*. Selain itu ukuran gambar yang terlalu kecil bisa menjadi faktor CNN tidak bisa mempelajari *features* pada citra dengan baik. Beberapa arsitektur CNN menyarankan ukuran *input shape* dengan ukuran 224x224x3 piksel dengan color RGB.

b. Batch size

Pada saat pelatihan data (*Training*) jika dilakukan sekaligus akan memberatkan proses pelatihan secara komputasi, apalagi jika data yang dilatih memiliki bobot yang besar hingga ribuan apalagi data yang di proses merupakan data RGB. Untuk mengatasi hal ini *batch size* berfungsi untuk memecah dataset menjadi beberapa bagian kecil. Biasanya nilai yang digunakan secara umum untuk digunakan sebagai *batch size*, yaitu 16/32/64/128/256.

c. Epoch

Epoch adalah suatu representasi angka yang menunjukkan satu putaran penuh saat proses pelatihan (*training*) terhadap seluruh *dataset*. Semakin tinggi nilai *epoch* maka nilai skor akurasi akan semakin tinggi dan nilai *loss* akan semakin rendah. Proses ini di hentikan apabila nilai *loss* dan nilai akurasi tidak mengalami naik dan turun perubahan secara signifikan.

d. Jumlah Filter & Filter Size

Jumlah filter pada *Convolutional Layer* bisa disesuaikan, secara umum banyak peneliti yang menggunakan nilai 2^n (misalnya: 8/16/32/64/128/256) sebagai angka pada filter. Dan untuk ukuran filter pada *Convolutional Layer* dan *Pooling Layer* bisa disesuaikan juga, secara umum banyak peneliti menggunakan ukuran filter *Convolutional* yaitu 3x3 atau 5x5 dengan *stride*=1 atau *stride*=2, dan untuk *Pooling* yaitu 2x2 dengan *stride*=2.

e. Stride

Stride merupakan parameter yang menentukan jumlah pergeseran pada filter jika nilai *stride*=1 maka filter akan bergeser sebanyak 1 piksel secara *horizontal* lalu *vertikal* terus berulang.

f. Padding

Padding berfungsi untuk menentukan jumlah piksel yang berisi nilai 0 yang ditambahkan di setiap sisi dari

input dimensi map. Hal ini bertujuan untuk memanipulasi dimensi *output* saat proses *convolutional* atau *pooling*, sehingga dimensi *output* dapat diukur agar tetap sama seperti dimensi *input*.

g. Optimizer

Optimizer merupakan sebuah metode yang biasanya digunakan untuk meng-*update* bobot (*weight*). Hal ini bertujuan untuk menurunkan nilai *loss*. Biasanya para ahli menggunakan *optimizer* : *Adam Optimizer* karena *Optimizer* ini terkenal memiliki hasil yang baik dan dianggap sebagai *optimizer* yang cepat dalam mencapai *loss* minimum (*converge*)[21].

h. Learning rate

Learning rate merupakan parameter yang mengatur cepat atau lambatnya proses model dalam mempelajari data saat pelatihan (*training*). Hal ini sangat berpengaruh terhadap hasil dari skor akurasi pada model yang dilatih. Ada kekurangan dan kelebihan dalam penggunaan *learning rate* yaitu :

- Penggunaan *learning rate* yang terlalu kecil akan memerlukan waktu yang lama pada saat proses *training* demi mencapai bobot yang optimal.

- Penggunaan *Learning rate* yang terlalu besar, akan mempercepat waktu saat proses *training* dilakukan namun hal ini akan melewatkan bobot optimal yang di dapat. Nilai *learning rate* yang biasanya umum digunakan oleh para ahli adalah : 0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3 (Source : <https://playground.tensorflow.org/>).

i. Jumlah Hidden Layer & Jumlah Node/Neuron

Hidden Layer pada *Fully Connected* bisa disesuaikan kebanyakan para ahli menggunakan *hidden layer* dimulai dari 2 *hidden layer*. Kemudian untuk *neuron* pada setiap *Hidden Layer* bisa disesuaikan juga, kebanyakan para ahli biasanya menggunakan nilai 2^n (misalnya: 8/16/32/64/128/256)[22]. Namun jika jumlah *neuron* yang digunakan semakin banyak, maka waktu pelatihan akan semakin lama dan ukuran model CNN akan besar juga.

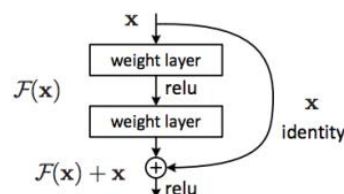
2.7 ResNet152v2

Untuk mengatasi masalah yang dihadapi selama pelatihan, pada tahun 2015, He et al. mengusulkan ResNet. Arsitektur ini cukup revolusioner pada saat itu karena arsitektur ini menjadi *state of the art* pada saat itu tidak hanya dalam klasifikasi, namun dalam semua kategori termasuk object detection, dan semantic segmentation. Namun, di ResNet informasi sisa selalu diteruskan dan pintasan identitas tidak pernah ditutup[6]. Tautan sisa (koneksi pintas) mempercepat konvergensi jaringan dalam, sehingga memberi ResNet kemampuan untuk menghindari masalah pengurangan gradien yaitu suatu keadaan dimana hasil gradien yang dipelajari oleh model, tidak dapat mencapai *layer* pertama karena mengalami perkalian berkali-kali sehingga *layer* pertama tidak menerima gradien apa-apa, atau secara singkatnya hal ini menyebabkan CNN tidak dapat belajar dari error yang telah dikalkulasi[15]. ResNet dengan kedalaman 152 lapisan, (memiliki kedalaman 20 dan 8 kali lebih dalam dari AlexNet dan VGG, masing-masing) memenangkan kejuaraan 2015-ILSVRC[6]. Bahkan dengan kedalaman yang meningkat, ResNet menunjukkan komputasi yang lebih rendah dalam mencapai kompleksitasnya dari VGG[6].

| Model | Size | Top-1 Accuracy | Top-5 Accuracy | Parametes | Depth |
|-------------|--------|----------------|----------------|------------|-------|
| ResNet152V2 | 232 MB | 0.780 | 0.942 | 60,380,648 | - |

Gambar 11. Resnet152V2 di Keras Application

Hal yang diusung oleh Kaiming He et al. pada saat itu adalah dengan menggunakan sesuatu yang bernama *residual block*, blok ini adalah blok yang ada pada tiap lapis arsitektur CNN Resnet dan menjadi fundamental dari arsitektur tersebut, gambaran dari blok ini dapat dilihat pada Gambar 5.



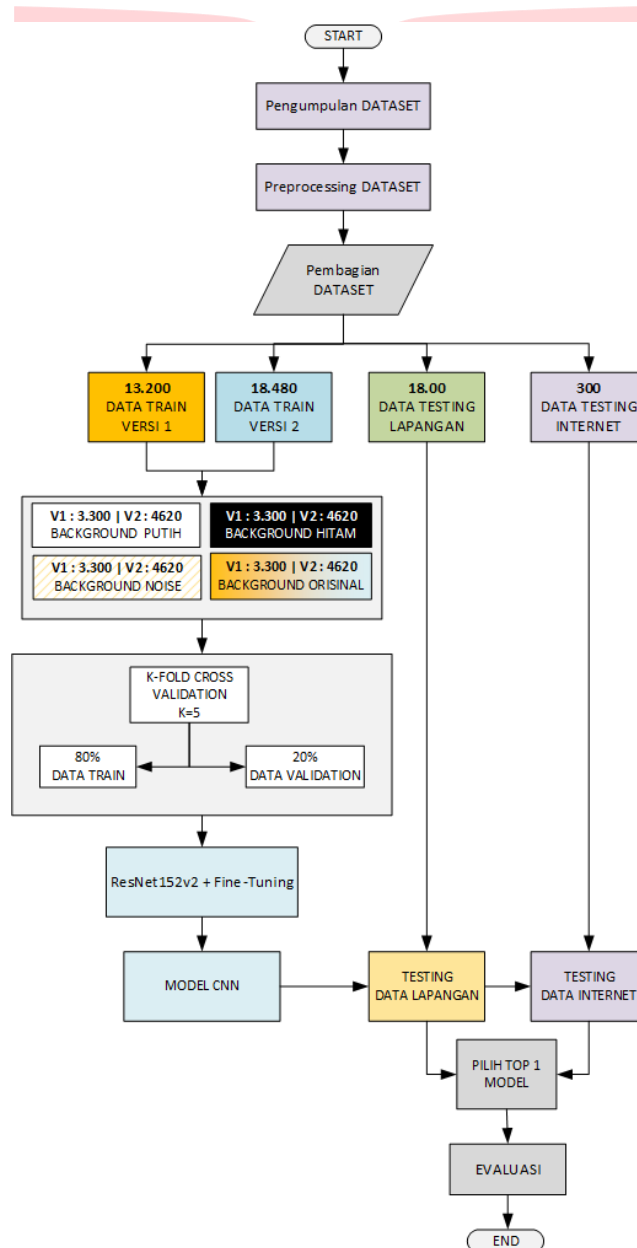
Gambar 11. Ilustrasi Blok residual yang menjadi fondasi dari arsitektur Resnet[23].

Blok ini menambahkan suatu jalan pintas yang berfungsi sebagai fungsi identitas, yang secara tidak langsung akan melewati proses *training* untuk satu *layer* atau lebih, sehingga membuat sesuatu yang bernama *residual block*, dengan melewati beberapa *layer* arsitektur ini dapat mengatasi permasalahan *vanishing gradient problem* karena gradien dapat sampai kepada *layer* pertama tanpa harus melewati jumlah perkalian yang sama dibandingkan dengan arsitektur yang sama tanpa menggunakan *residual block*.

3. Sistem yang Dibangun

3.1 Deskripsi Umum

Sistem yang dibangun pada penelitian ini yaitu untuk mengklasifikasikan kaktus *Grafting* berdasarkan citra *input* pada bagian *Scion* kaktus *Grafting*. Metode yang digunakan pada penelitian ini yaitu *Convolutional Neural Network* (CNN) dengan arsitektur ResNet152V2. Studi kasus yang diangkat memiliki 2 versi dataset *train* yaitu pada data *train* versi 1 mengklasifikasikan 5 spesies kaktus *Grafting* yang memiliki persamaan warna hijau, 5 spesies yang diklasifikasikan antara lain *Enchinopsis Peruviana*, *Gymnocalycium Baldianum*, *Mammillaria Spinosissima cv. Un Pico*, *Opuntia Microdasys*, *Parodia Warasii*, hal ini bertujuan mengukur seberapa akurat model dalam mengklasifikasikan objek dari sisi bentuk yang berbeda dan kemiripan warna yang sama. Kemudian pada data *train* versi 2 mengklasifikasikan 7 spesies kaktus *Grafting* yang dimana 2 spesies diantaranya diambil dari data *train* versi 1 agar menambah variatif data dan memiliki 3 perbedaan warna dan pada setiap 2-3 spesies memiliki warna yang sama, hal ini bertujuan mengukur seberapa akurat model dalam mengklasifikasikan objek dari sisi bentuk dan warna yang berbeda. Adapun 7 spesies yang di klasifikasikan yaitu : *Astrophytum Capricorne*, *Blossfeldia Liliputana*, *Copiapoa Laui*, *Cephalocereus Senilis*, dan *Echinopsis Chocolate*. Gambaran lebih jelasnya penulis menggambarkan sistem yang dibangun dengan menggunakan flowchart diagram sebagai berikut :



Gambar 12. FlowChart Alur Pembuatan Sistem

3.2 Pengumpulan DATASET






Pada Tahap ini penulis mengumpulkan dataset berupa foto dari tanaman Kaktus *Grafting* yang ada di pasaran dan tempat budidaya yang berlokasi di Lembang. Peneliti berhasil mengumpulkan 10 spesies tanaman kaktus *Grafting* yang dibagi ke dalam 2 versi penelitian berikut tabel dataset spesies kaktus *Grafting* :

3.2.1 Versi 1

Pada data *train* versi 1 ini diujikan 5 spesies kaktus yang memiliki persamaan warna pada setiap spesiesnya yaitu kaktus *Grafting* yang berwarna hijau, hal ini bertujuan menguji seberapa akurat model dalam mengklasifikasikan kaktus *Grafting* dari sisi bentuk tanaman yang berbeda namun memiliki kesamaan warna yang hampir mirip. Dari jumlah masing-masing setiap spesies memiliki gambar sekitar 660 gambar data *Train*, 150 gambar data *Testing* atau data lapangan, dan 25 gambar internet. Berikut gambaran dataset setiap spesiesnya :

Tabel 1. Tabel informasi data train versi 1 (No, Nama spesies, jumlah data, contoh citra).

Sumber : <http://www.llifile.com/Encyclopedia>

| No. | Nama Spesies | Ciri-ciri Spesies | Jumlah data | Contoh Data |
|-----|---|---|-------------|---|
| 1. | <i>Echinopsis Peruviana</i> | Kaktus ini memiliki bentuk tubuh seperti bantal vertikal, dengan struktur duri bergaris. | 660 |  |
| 2. | <i>Gymnocalycium Baldium</i> | Kaktus ini memiliki bentuk batang bulat, dengan duri-duri yang melengkung, dan akar lebat yang menjulur jauh ke dalam tanah | 660 |  |
| 3. | <i>Mammillaria Spinosissima cv. Un Pico</i> | Kaktus ini memiliki bentuk batang bulat, memiliki duri yang halus yang memanjang yang bisa mencapai hingga ± 5 cm. | 660 |  |
| 4. | <i>Opuntia Microdasys</i> | Kaktus ini biasanya bertubuh pendek, banyak bercabang, memiliki batang kecil seperti bantal dan tidak berduri tajam, namun memiliki glochids kuning keemasan (duri seperti rambut). bantal sering tumbuh berpasangan memberikan penampilan telinga kelinci. | 660 |  |
| 5. | <i>Parodia Warasii</i> | Kaktus ini memiliki batang Tunggal, pendek, dan mengkilap, dan memiliki duri seperti jarum dengan panjang 10-40 mm dan membentuk barisan di bawah rusuknya. | 660 |  |






| | |
|---------------|--------------|
| Jumlah | 3.300 |
|---------------|--------------|



3.2.2 Versi 2

Pada data *train* versi 2 ini diujikan 7 spesies kaktus yang dimana 2 spesies diantaranya diambil dari data *train* versi 1 agar menambah variatif data, data *train* versi 2 memiliki 3 perbedaan warna dan memiliki persamaan warna pada setiap 2-3 spesiesnya, hal ini bertujuan menguji seberapa akurat model dalam mengklasifikasikan kaktus *Grafting* dari sisi bentuk dan warna tanaman yang berbeda. Dari jumlah masing-masing setiap spesies memiliki gambar sekitar 660 gambar data *Train*, 150 gambar data Testing atau data lapangan dan 25 gambar internet. Berikut gambaran dataset setiap spesiesnya :

Tabel 2. Tabel informasi data train versi 2 (No, Nama spesies, jumlah data, contoh citra).

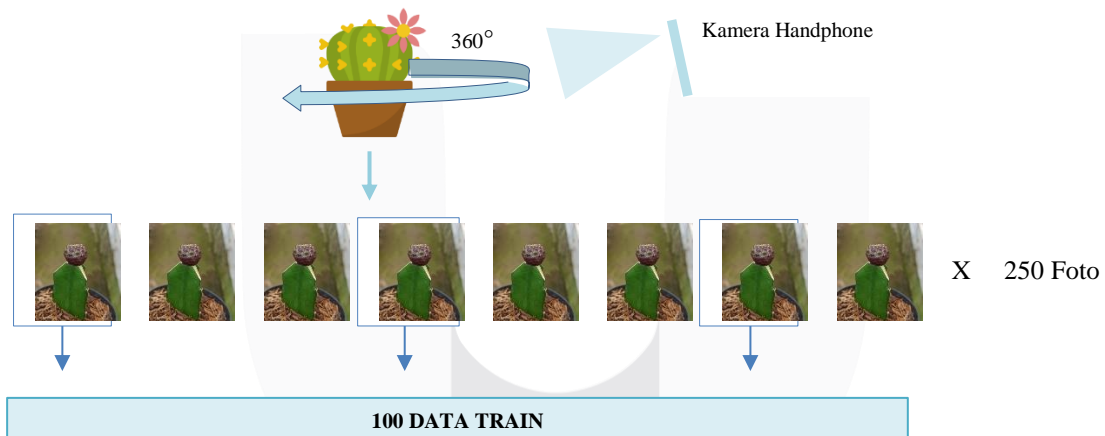
Sumber : <http://www.llifile.com/Encyclopedia>

| No. | Nama Spesies | Ciri-ciri Spesies | Jumlah data | Contoh Data |
|-----|--|---|-------------|---|
| 1. | <i>Copia Laui</i> | Kaktus ini berbentuk mini, terkadang berkelompok, dengan warna abu-abu jernih hingga merah-coklat-abu-abu, kadang-kadang kehijauan. Bagian atas cekung ditutupi dengan wol keputihan. | 660 |  |
| 2. | <i>Echinopsis Chocolate</i> | Kaktus ini membentuk kelompok dengan membentuk seperti jambul merah-cokelat bertekstur lembut, biasanya kaktus ini tidak memiliki duri pada tubuhnya. Pertumbuhan kaktus ini biasanya tumbuh seperti bentuk jambul. | 660 |  |
| 3. | <i>Echinopsis Peruviana</i> | Kaktus ini memiliki bentuk tubuh seperti bantal vertikal, dengan struktur duri bergaris. | 660 |  |
| 4. | <i>Mammillaria Spinossissima cv. Un Pico</i> | Kaktus ini memiliki bentuk batang bulat, memiliki duri yang halus yang memanjang yang bisa mencapai hingga ± 5 cm. | 660 |  |
| 5. | <i>Astrophytum Capricorne</i> | Kaktus ini membentuk hampir membulat. biasanya ditutupi dengan bintik-bintik putih yang khas, memiliki duri dengan panjang 5 hingga 10 cm, tumbuh lebat, membengkok seperti tanduk Capricorn. | 660 |  |

| | | | | |
|---------------|-------------------------------|--|--------------|---|
| 6. | <i>Cephalocereus Senilis</i> | Kaktus ini memiliki bentuk tubuh melingkar dan panjang dengan ciri rambut halus putih salju seperti mantel, memiliki Duri berwarna kuning yang di tutupi rambut putih dengan panjang 1-5 cm. | 660 |  |
| 7. | <i>Blossfeldia Liliputana</i> | kaktus ini memiliki bentuk mini biasanya berdiameter hingga 1,2 cm, berkelompok dan tanpa duri, berwarna hijau dengan corak keputihan dan salah satu yang paling mahal harganya. | 660 |  |
| Jumlah | | | 4.620 | |

3.2.3 Dataset Primer

Dataset Primer merupakan dataset yang dikumpulkan dengan cara peneliti mengunjungi tempat budidaya kaktus *Grafting*nya langsung ataupun penjual yang menjual kaktus *Grafting* langsung yang berlokasi diLembang. Kemudian kaktus di foto dengan *Smartphone* dengan memutari tanaman dari samping tanaman kaktus *Grafting* atau berdasarkan *Best-Viewnya*, setiap 1 spesies tanaman dalam 1 pot memiliki 25-30 foto, hal ini bertujuan untuk mencari sedetail mungkin ciri-ciri dari tanaman spesies tertentu agar hasil akurasi klasifikasi tinggi danantisipasi apabila terjadi *blur* pada gambar. Berikut ini adalah contoh pengambilan dari dataset Primer :

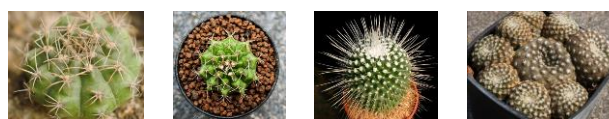


Gambar 13. Ilustrasi proses pengambilan Data Train

Peneliti mengambil gambar sebanyak 10 pot tanaman kaktus *Grafting* sehingga menghasilkan 250 foto dalam 1 spesies, namun dalam 250 gambar akan di pilah lagi untuk dimasukan kedalam data *Train*, dan peneliti mengambil 100-150 foto secara acak dari 250 foto untuk di Augmentasi. Hal ini bertujuan agar model dapat mempelajari data secara variatif.

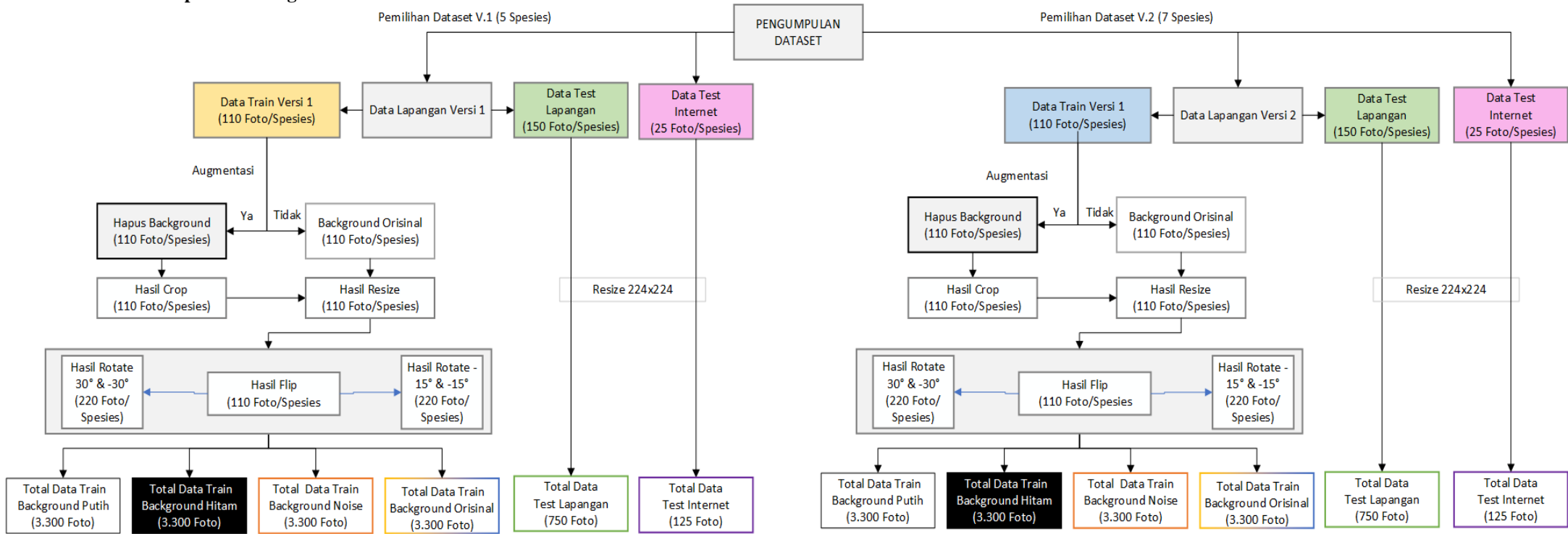
3.2.4 Dataset Sekunder

Dataset Sekunder merupakan dataset yang dikumpulkan dengan cara mencari data pada internet atau dengan data publik yang dikumpulkan dari sumber Internet, Sosial media, Forum website, Paper, dan lain-lain. Berikut ini adalah contoh dari dataset Sekunder :



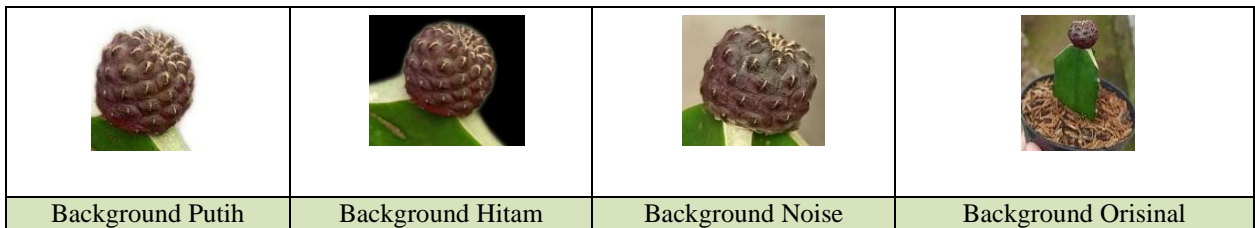
Gambar 14. Gambar dataset Sekunder dari internet, sosial media, Forum website, Paper, dll.

3.3 Praproses & Augmentasi DataSet



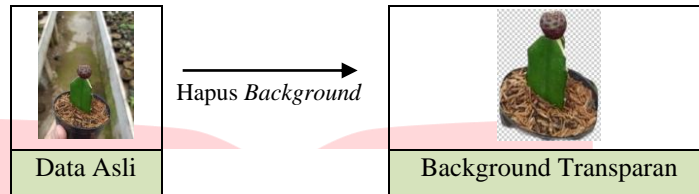
Gambar 15. Praproses dan Augmentasi dataset

Pada tahap ini penulis akan memilah data *train* dari 250 menjadi 100-150 data per spesies yang dimana proses ini dilakukan dengan cara menghapus data foto yang *blur*, *noise* dan gangguan lainnya pada foto yang diambil, hal ini bertujuan agar model dapat mempelajari setiap *features* pada citra dengan baik serta meningkatkan hasil akurasi yang didapatkan. 100-1500 foto yang sudah di ambil akan dilakukan penghapusan *background*. Karna penulis melakukan pengujian skenario perbedaan *background* dataset (*background* Putih, *background* Hitam, *background* Noise, dan *background* Orisinal). Ilustrasi langkah-langkah proses mendapatkan 4 skenario perbedaan warna *background* sebagai berikut :



Gambar 16. Contoh dataset skenario perbedaan warna Background yang akan digunakan

Langkah awal, sebelum melakukan pengujian skenario *background* maka perlu melakukan penghapusan *background* menjadi *background* transparan yang dilakukan satu persatu dari 110 gambar/spesies dengan dengan bantuan *tools Photoshop* untuk dijadikan data pada 2 skenario *background* putih dan *background* hitam.









Gambar 17. Ilustrasi Hapus Background

Penghapusan *background* dilakukan sebelum proses *resize*, hal ini bertujuan agar gambar tidak mengalami banyak kehilangan detail informasi warna pada gambar saat proses kompresi/*render* setelah dilakukannya pengeditan gambar berlangsung. Kemudian penulis melakukan Augmentasi data pada foto Dataset dimana data akan di *Resize*, *Cropping*, *Flip*, dan *Rotate*. Hal ini bertujuan agar gambar semakin banyak dan semakin bervariasi pada saat proses komputasi sistem akan bekerja dengan baik saat proses pelatihan. Berikut contoh langkah-langkah Augmentasi pada *background* putih sebagai berikut :

3.3.1 Ubah Ukuran Resolusi (*Resize*) dan Ubah warna background

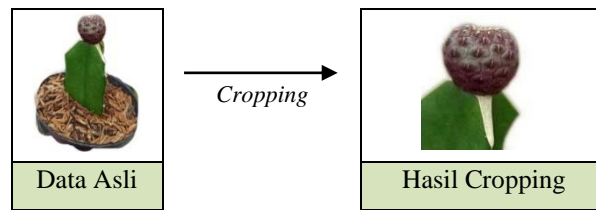
Langkah pertama *Resize*, *Resize* adalah proses dimana gambar di ubah ukuran pixelnya entah itu menjadi besar atau menjadi kecil sesuai dengan kebutuhan masing-masing. Karna pada kasus ini membutuhkan ukuran data seminimal mungkin maka penulis melakukan pengecilan pada gambar dengan tujuan agar mempercepat proses komputasi pada saat *training* dalam pencarian fitur pada gambar karena gambar yang diolah mempunyai ukuran *pixel* yang kecil. Agar proses *resize* dan pengubahan warna *background* tidak memakan waktu penulis melakukan pengaturan *resize* dengan ukuran 224x224x3 dengan bantuan 2 *tools*. Dengan bantuan *tools IrfanView* secara otomatis *tools* ini akan mengubah *background* transparan menjadi *background* berwarna hitam dengan ukuran gambar yang sudah di atur oleh penulis, kemudian untuk *background* putih menggunakan *tools ACDSee* dan secara otomatis *tools* ini akan mengubah *background* transparan menjadi *background* berwarna putih dengan ukuran gambar yang sudah di atur oleh penulis. Berikut adalah contoh proses *Resize* dan pengubahan warna background pada suatu gambar :

| Data Asli | Tools | Hasil Resize dan ubah warna |
|---|---|---|
|  |  → <i>Resize</i> |  |
| Background Transparan | ACDSee | Background Putih |
|  |  → <i>Resize</i> |  |
| Background Transparan | IrfanView | Background Hitam |

Gambar 18. Proses Resize+Ubah Warna pada foto.

3.3.2 Pemotongan (*Cropping*)

Langkah kedua *Cropping*, *Cropping* adalah proses dimana gambar di potong pada bagian tertentu. Tujuan dari proses ini yaitu agar proses klasifikasi tidak terpengaruh oleh bagian gambar yang tidak penting. Berikut adalah contoh proses *Cropping* pada suatu gambar :

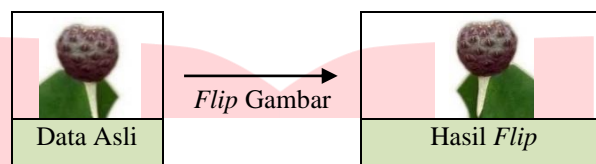


Gambar 19. Proses Cropping pada foto.

Karna penulis ingin model mempelajari *Scion* pada tanaman kaktus *Grafting* maka proses *Cropping* di lakukan pada bagian atas tanamanaan kaktus *Grafting*, hal ini bertujuan agar model tidak mempelajari objek lain di sekitar objek yang ditentukan.

3.3.3 Pencerminan (*Flipping*) : *Horizontal*

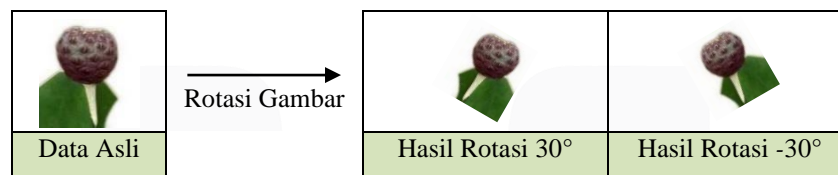
Langkah ke tiga *Flipping*, proses ini penulis melakukan operasi *Flipping* yaitu suatu perubahan orientasi citra secara *Horizotal* dengan tujuan menambah variatif suatu data. berikut proses *Flip* dilakukan :



Gambar 20. Proses Flip pada foto.

3.3.4 Rotasi (*Rotate*)

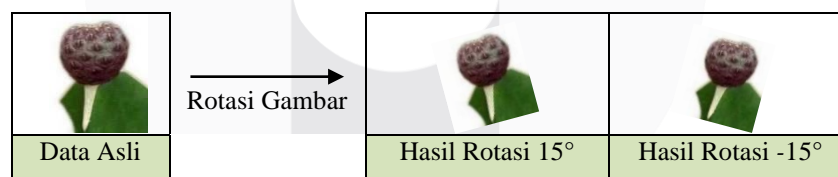
Langkah ke empat, melakukan proses rotasi, yaitu suatu operasi memutar suatu citra terhadap titik pusatnya terhadap arah jarum jam atau berlawanan jarum jam. Hal ini dilakukan bertujuan menambah data agar semakin variatif sehingga model dapat mempelajari data baru dengan mudah.



Gambar 21. Proses Rotasi -30° & 30° pada foto.

3.3.5 Pencerminan (*Flipping*) : *Horizontal* + Rotasi (*Rotate*)

Terakhir penulis melakukan proses *Flip* + rotasi 15°, yaitu suatu operasi memutar pada suatu citra yang telah di *Flip* terhadap titik pusatnya terhadap arah jarum jam atau berlawanan jarum jam. Hal ini dilakukan bertujuan menambah data agar semakin variatif sehingga model dapat mempelajari data baru dengan mudah.



Gambar 22. Proses Rotasi -15° & 15° pada foto.

3.3.6 Data Testing

Data *Testing* kaktus *grafting* perlu dilakukan pemilihan data dari data yang *blur* dan pada data *testing* citra gambar tidak memiliki *noise* yang ramai. Pada data *Testing* dilakukan proses Augmentasi data hanya dengan melakukan oprasi *Resize* saja, dan disini penulis menyediakan data Testing lapangan berjumlah 150 data dan data Testing Internet berjumlah 25 data pada setiap spesiesnya. Hal ini dikarenakan data Testing adalah data yang akan di ujikan kepada model dari hasil proses *Train* nanti untuk menentukan seberapa akurat model dalam mengenali data baru yang di berikan. Kemudian data *Train* dan data Testing akan di pisahkan kedalam folder yang berbeda.

3.4 K-Fold Cross Validation

Setelah menentukan data yang akan dimasukan, kemudian melakukan pengujian 4 skenario *background* data *train* dengan metode *K-Fold Cross Validation* pada setiap kategori data *train* yang di uji, yang dimana disini penulis memberikan nilai $k=5$ dan sistem akan secara otomatis melakukan pembagian 80% data *Train* dan 20% data *Validation*. Tujuan menggunakan K-fold disini agar seluruh data *train* tervalidasi dengan baik agar memaksimalkan nilai akurasi yang di dihasilkan oleh model menjadi lebih akurat.

3.5 Perancangan CNN

Pada penelitian ini menggunakan arsitektur ResNet152v2 dengan menggunakan *dropout*. Pada arsitektur yang digunakan akan dilatih pada data (*Training & validation*) pada sistem model yang dibangun, data yang akan dilatih pada arsitektur ini memiliki 2 versi yaitu versi 1 dengan warna tanaman sama dan versi 2 dengan warna tanaman berbeda. Kemudian disetiap versinya memiliki 4 skenario *background* yang berbeda (*Background Putih, Background Hitam, Background noise, dan Background Orisinal*).

Untuk memaksimalkan hasil akurasi kemudian perlu mengatur *hyperparameter* seperti *learning rate, dropout*, dan lain-lain. Karna penulis mengatur *fold=5* sehingga model yang dihasilkan berjumlah 5 model yang dimana setiap model akan diujikan ke data test dan data internet sehingga di dapatlah model terbaik untuk setiap versi data dan skenario yang di uji.

3.6 Evaluasi Model

Evaluasi pada setiap model versi 1 yang di ujikan akan dilihat melalui visualisasi dalam bentuk *confusion matrix* dan melihat rata-rata akurasi yang dihasilkan oleh setiap model. Setiap model dari hasil *training* dataset sebanyak 3.300 untuk setiap skenario *background* data atau sekitar 660 data *training* untuk setiap spesiesnya pada data *train* versi 1 akan diujikan pada data testing sebanyak 750 data testing lapangan dan 125 data testing internet atau sebanyak 150 data testing lapangan dan 25 data testing internet untuk setiap spesiesnya.

Evaluasi pada setiap model versi 2 yang di ujikan akan dilihat melalui visualisasi dalam bentuk *confusion matrix* dan melihat rata-rata akurasi yang dihasilkan oleh setiap model. Setiap model dari hasil *training* dataset sebanyak 4.620 untuk setiap skenario *background* data atau sekitar 660 data *training* untuk setiap spesiesnya pada data *train* versi 2 akan diujikan pada data testing sebanyak 1050 data testing lapangan dan 175 data testing internet atau sebanyak 150 data testing lapangan dan 25 data testing internet untuk setiap spesiesnya.

4. Evaluasi





4.1 Skenario Pengujian & Hasil Penelitian

Pada proses pembangunan sistem klasifikasi pada spesies kaktus *Grafting* dilakukan dengan menerapkan 2 versi dataset *train* dan 4 skenario *background* yang sebelumnya telah di lakukan augmentasi data. Kemudian penulis melakukan penyesuaian pada *hyperparameter* untuk menyesuaikan performansi akurasi. Beberapa skenario dilakukan sebagai berikut :

4.1.1 Versi 1

Pada versi 1 penulis mengumpulkan data *train* dengan 5 spesies kaktus yang memiliki warna *Scion* dominan sama yaitu berwarna hijau, dengan total 660 per spesies. Kemudian pada setiap spesies penulis melakukan 4 pengujian skenario *background* pada citra gambar untuk menguji pengaruh performansi akurasi pada data citra yang di uji. Sehingga pada data *train* versi 1 memiliki total data sebagai berikut :





Tabel 3. Paparan data train Versi 1 & skenario pengujian.


| No. | Skenario Background | Spesies Kaktus Grafting | | | | | Total |
|---------------------------|---|-------------------------|-------------------------|--------------------------------------|--------------------|-----------------|---------------|
| | | Echinopsis Peruviana | Gymnocalycium Baldianum | Mammillaria Spinosissima cv. Un Pico | Opuntia Microdasys | Parodia Warasii | |
| 1. |  | 660 | 660 | 660 | 660 | 660 | 3.300 |
| | Background Putih | | | | | | |
| 2. |  | 660 | 660 | 660 | 660 | 660 | 3.300 |
| | Background Hitam | | | | | | |
| 3. |  | 660 | 660 | 660 | 660 | 660 | 3.300 |
| | Background Noise | | | | | | |
| 4. |  | 660 | 660 | 660 | 660 | 660 | 3.300 |
| | Background Orisinal | | | | | | |
| Total Data Versi 1 | | | | | | | 13.200 |

4.1.2 Versi 2

Pada versi 2 penulis mengumpulkan data *train* dengan 7 spesies kaktus yang dimana 2 spesies diantaranya diambil dari data *train* versi 1 agar menambah variatif data yang memiliki warna dan bentuk *Scion* yang berbeda, namun dari ke 7 spesies memiliki 2-3 spesies yang memiliki warna yang sama hal ini bertujuan agar sistem dapat mempelajari lagi data yang lebih kompleks. Data yang di uji berjumlah 660 data per spesies. Kemudian pada setiap spesies penulis melakukan 4 pengujian skenario *background* pada citra gambar untuk menguji pengaruh performansi akurasi pada data citra yang di uji. Sehingga pada data *train* versi 2 memiliki total data sebagai berikut :

Tabel 4. Paparan data train Versi 2 & skenario pengujian.

| No. | Skenario Background | Spesies Kaktus Grafting | | | | | | | Total |
|-----|---|-------------------------|----------------------|----------------------|--------------------------------------|------------------------|------------------------|-----------------------|-------|
| | | Copiapoa Laui | Echinopsis Chocolate | Echinopsis Peruviana | Mammillaria Spinosissima cv. Un Pico | Blossfeldia Liliputana | Astrophytum Capricorne | Cephalocereus senilis | |
| 1. |  | 660 | 660 | 660 | 660 | 660 | 660 | 660 | 4.620 |
| | Background Putih | | | | | | | | |
| 2. |  | 660 | 660 | 660 | 660 | 660 | 660 | 660 | 4.620 |
| | Background Hitam | | | | | | | | |
| 4. |  | 660 | 660 | 660 | 660 | 660 | 660 | 660 | 4.620 |
| | Background Noise | | | | | | | | |
| 4. |  | | | | | | | | |

| | | | | | | | | | |
|---|---------------------|-----|-----|-----|-----|-----|-----|-----|---------------|
|  | 660 | 660 | 660 | 660 | 660 | 660 | 660 | 660 | 4.620 |
| | Background Orisinal | | | | | | | | |
| Total Data Versi 2 | | | | | | | | | 18.480 |

Arsitektur *Convolutional Neural Network* (CNN) yang digunakan untuk *training* yaitu ResNet152V2. Proses *training* menggunakan teknik *K-Fold Validation* hal ini bertujuan agar model dapat memvalidasi data *train* secara optimal dengan nilai $K = 5$, yang berarti data *train* yang digunakan sebesar 80% dan data *validation* sebesar 20%. Kemudian penulis melakukan pengaturan *hyperparameter* untuk mengoptimalkan performansi akurasi sebagai berikut :

- *Input shape* : 224 x 224 x 3 (RGB)
- *Batch size* : 8 (Literasi dalam satu kali *epoch*)
- *Epoch* : 10 (Pengujian Terbaik)
- *Dropout* : 0.45 %
- *Optimizer* : Adam
- *Learning rate* : Adam 0.001

Karena pada proses *training* menggunakan teknik *K-Fold* maka akan menghasilkan 5 model pada setiap skenario *background* yang di uji sehingga untuk 1 versi menghasilkan 20 model, karena penulis menggunakan 2 versi maka total seluruh model yang dihasilkan berjumlah 40 model. Kemudian pada setiap model akan dilakukan pengujian (*Testing*) pada data lapangan dan data internet yang memiliki tingkat kemiripan data yang sangat jauh dengan data *train* untuk memvalidasi model dengan akurasi terbaik dalam mengenali kaktus *Grafting*.

Tabel 5. rincian jumlah data Testing

| No. | Jenis data Testing | Dataset Versi 1 | Dataset Versi 2 | Total |
|-----|--------------------|-------------------|--------------------|--------------|
| 1. | Data Lapangan | 150/Spesies (750) | 150/Spesies (1050) | 1.800 |
| 2. | Data Internet | 25/Spesies(125) | 25/Spesies (175) | 300 |

Masing-masing model akan di ujikan ke data Testing seperti yang terlihat pada tabel di atas dan hasil pengujiannya sebagai berikut :

Tabel 6. Akurasi Terbaik pada data train Versi 1 dan Versi 2 dengan skenario background putih

| Dataset | Skenario Background | Model | Akurasi | | | |
|---------|---------------------|-------------|----------------|----------------|--------------------|--------------------|
| | | | Training | Validation | Testing (Lapangan) | Testing (Internet) |
| Versi 1 | ORISINAL | ResNet152v2 | 85.45 % | 98.03 % | 81.33 % | 90.40 % |
| | HITAM | | 92.27 % | 99.24 % | 54.27 % | 88.00 % |
| | NOISE | | 90.76 % | 99.39 % | 91.87 % | 95.20 % |
| | PUTIH | | 95.23 % | 99.55 % | 94.13 % | 97.60 % |
| Versi 2 | ORISINAL | | 82.77 % | 97.19 % | 72.47 % | 76.00 % |
| | HITAM | | 91.80 % | 99.13 % | 64.57 % | 79.42 % |
| | NOISE | | 85.23 % | 99.46 % | 65.33 % | 80.57 % |
| | PUTIH | | 93.83 % | 99.46 % | 82.10 % | 86.86 % |

Dari pemaparan tabel di atas dapat dilihat model terbaik yang memiliki tingkat akurasi yang tinggi dari setiap versi dan skenario pengujian terhadap *background* yang telah diujikan kepada data testing Lapangan dan data testing Internet. Model terbaik yang memiliki tingkat akurasi yang tinggi yaitu :

a. Versi 1 [skenario *Background* Putih]

Dengan *hyperparameter* sebagai berikut :

- *Input shape* : 224 x 224 x 3 (RGB)
- *Batch size* : 8 (Literasi dalam satu kali *epoch*)
- *Epoch* : 10 (Pengujian Terbaik)
- *Dropout* : 0.45 %
- *Optimizer* : Adam
- *Learning rate* : Adam 0.001

Menghasilkan *Score* akurasi Testing Lapangan sebesar 94.13% dan Testing Internet sebesar 97.60%.

b. Versi 2 [skenario *Background* Putih]

Dengan *hyperparameter* sebagai berikut :

- *Input shape* : 224 x 224 x 3 (RGB)
- *Batch size* : 8 (Literasi dalam satu kali *epoch*)
- *Epoch* : 10 (Pengujian Terbaik)
- *Dropout* : 0.45 %
- *Optimizer* : Adam
- *Learning rate* : Adam 0.001

Menghasilkan *Score* akurasi Testing Lapangan sebesar 82.10% dan Testing Internet sebesar 86.86%.

Kemudian pemaparan *Score* akurasi *training*, *Validation*, *Testing* Lapangan, dan *Testing* Internet dari 40 model CNN akan direkap dan dirata-ratakan dari akurasi *Score 5 fold* pada setiap skenario *background* dalam bentuk tabel sebagai berikut :

Tabel 7. Rata-rata Akurasi pada data train Versi 1 dan Versi 2 dengan 4 skenario background dari 5 fold.

| Data Train | Skenario Background | Model | Akurasi | | | |
|------------|---------------------|-------------|----------------|----------------|--------------------|--------------------|
| | | | Training | Validation | Testing (Lapangan) | Testing (Internet) |
| Versi 1 | ORISINAL | ResNet152v2 | 84.07 % | 98.18 % | 74.10 % | 86.72 % |
| | HITAM | | 92.30 % | 98.60 % | 64.53 % | 90.08 % |
| | NOISE | | 89.63 % | 98.39 % | 81.14 % | 88.80 % |
| | PUTIH | | 94.20 % | 99.33 % | 81.22 % | 89.60% |
| Versi 2 | ORISINAL | | 79.97 % | 97.21 % | 73.10 % | 72.33 % |
| | HITAM | | 92.31 % | 99.06 % | 71.80 % | 82.74 % |
| | NOISE | | 85.57 % | 98.98 % | 62.95 % | 82.51 % |
| | PUTIH | | 93.33 % | 99.48 % | 77.35 % | 85.59 % |

Rata-rata *Score* akurasi dari ke-5 *fold* diatas membuktikan seberapa baik model dalam melakukan klasifikasi secara umum dan dapat dilihat model terbaik pada versi 1 dalam mengklasifikasikan 5 spesies tanaman kaktus *grafting* yang memiliki warna dominan hijau dengan skenario data *train background* putih menghasilkan akurasi *Testing* lapangan sebesar 81.22% dan *Testing* Internet sebesar 89.60%. Kemudian pada versi 2 dalam mengklasifikasikan 7 spesies tanaman kaktus *grafting* yang memiliki warna dominan berbeda dengan skenario *background* putih juga menghasilkan akurasi *Testing* lapangan sebesar 77.35% dan *Testing* Internet sebesar 85.59%. Dari pemaparan hasil testing dari tabel-tabel di atas pada kasus klasifikasi citra *Scion* pada kaktus *Grafting* dapat disimpulkan bahwa :

1. Persamaan dan perbedaan warna sangat mempengaruhi akurasi sehingga :
 - jika warna sama dan bentuk berbeda akan memudahkan model dalam mengenali objek.
 - Jika warna sama dan bentuk sama akan menyulitkan model dalam mengenali objek.
 - Jika warna berbeda dan bentuk berbeda akan memudahkan model dalam mengenali objek.
 - Jika warna berbeda dan bentuk sama akan memudahkan model dalam mengenali objek.

Maka untuk menyasiasi hal ini agar model dapat mengenali objek dengan mudah penulis perlu mengatur *hyperparameter* serta mengatur data *train* sevariatif mungkin agar model dapat mengenali objek dengan sangat mudah.

2. Untuk meningkatkan akurasi model dalam mengklasifikasikan citra *Scion* pada kaktus *Grafting* perlunya mengatur *hyperparameter* yang cocok. *Hyperparameter* yang cocok pada studi kasus ini adalah sebagai berikut:

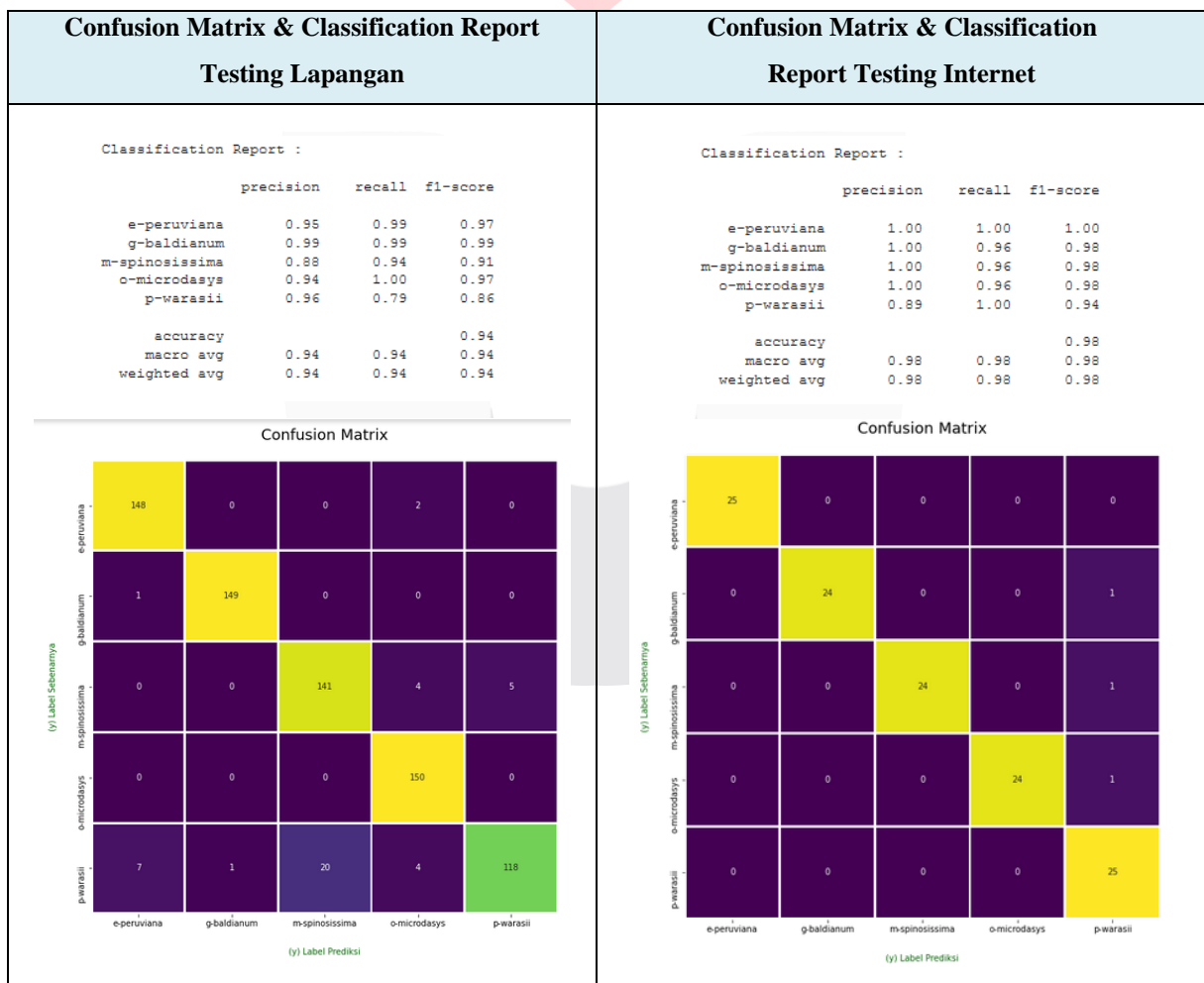
- *Input shape* : 224 x 224 x 3 (RGB)
- *Batch size* : 8 (Literasi dalam satu kali *epoch*)
- *Epoch* : 10 (Penguujian Terbaik)
- *Dropuot* : 0.45 %
- *Optimizer* : Adam
- *Learning rate* : Adam 0.001

4.2 Analisis Model Terbaik

Dari hasil pengerjaan sebelumnya dari mulai *Training* dan *Testing* dengan mengatur *hyperparameter* menghasilkan model CNN terbaik dengan *score* akurasi terbaik. Kemudian perlunya analisis terhadap kedua model terbaik yang telah dihasilkan, analisis dilakukan dengan menggunakan Confusion Matrix dan Classification Report untuk mendapatkan analisis secara visual, maka berikut rincian dari hasil analisis dari model yang di uji :

4.2.1 Model Versi 1



Model terbaik pada versi 1 yaitu versi dimana 5 spesies tanaman kaktus *Grafting* memiliki warna yang sama dengan skenario 4 macam *background*, model terbaiknya dihasilkan pada fold ke-2 dengan skenario *background* putih yang mencapai akurasi *Training* 95.13 %, *Validation* 99.55 %, *Testing Lapangan* 94.13%, dan *Testing Internet* 97.60%. Untuk detail visualnya sebagai berikut :



Gambar 23. Confusion Matrix dan Classification Report model CNN terbaik Versi 1 Skenario Background putih.

Kemudian tahap selanjutnya untuk mengetahui data prediksi yang salah di klasifikasikan oleh model, penulis melakukan analisa pengujian dengan program untuk mengetahui data spesies yang di prediksi salah. Berikut



contoh Visualiasasi 1 dari 20 data *Parodia Warasii* yang dianggap *Mammillaria Spinosissima cv. Un Pico* oleh model :

| Analisis Sistem | Data Train | Data Lapangan |
|--|---|---|
| <p>Hasil Prediksi: m-spinosissima Gambar Ke-13</p> <p>Hasil Prediksi: p-warasii Gambar Ke-14</p> <p>Hasil Prediksi: p-warasii Gambar Ke-15</p> <p>Hasil Prediksi: p-warasii Gambar Ke-16</p> <p>Hasil Prediksi: p-warasii Gambar Ke-17</p> <p>Hasil Prediksi: p-warasii Gambar Ke-18</p> |  |  |
| Data <i>Mammillaria Spinosissima cv. Un Pico</i> Dengan urutan gambar ke-13 pada data Testing Lapangan | <i>Parodia Warasii</i> | <i>Mammillaria Spinosissima cv. Un Pico</i> |

Gambar 24. Contoh kemiripan data Train dan data Testing Lapangan

Berdasarkan hasil informasi di atas berikut adalah beberapa hal yang dapat disimpulkan :

a. Hasil testing pada data testing lapangan yang dapat mempengaruhi akurasi merupakan kesamaan bentuk yang dimana dari total 150 data spesies *Parodia Warasii*, 20 diantaranya di prediksi salah dan di anggap spesies *Mammillaria Spinosissima cv. Un Pico* oleh model, hal ini terjadi kemungkinan akibat karena warna yang dominan sama yaitu hijau dan kemiripan bentuk yang memang memiliki bentuk sama-sama bulat duri yang hampir menyerupai. Kemudian analisa pengujian dengan program untuk mengetahui data spesies yang di prediksi salah juga dilakukan pada data internet. Berikut contoh Visualiasasi salah satu spesies yang di prediksi salah mengklasifikasikan oleh model :

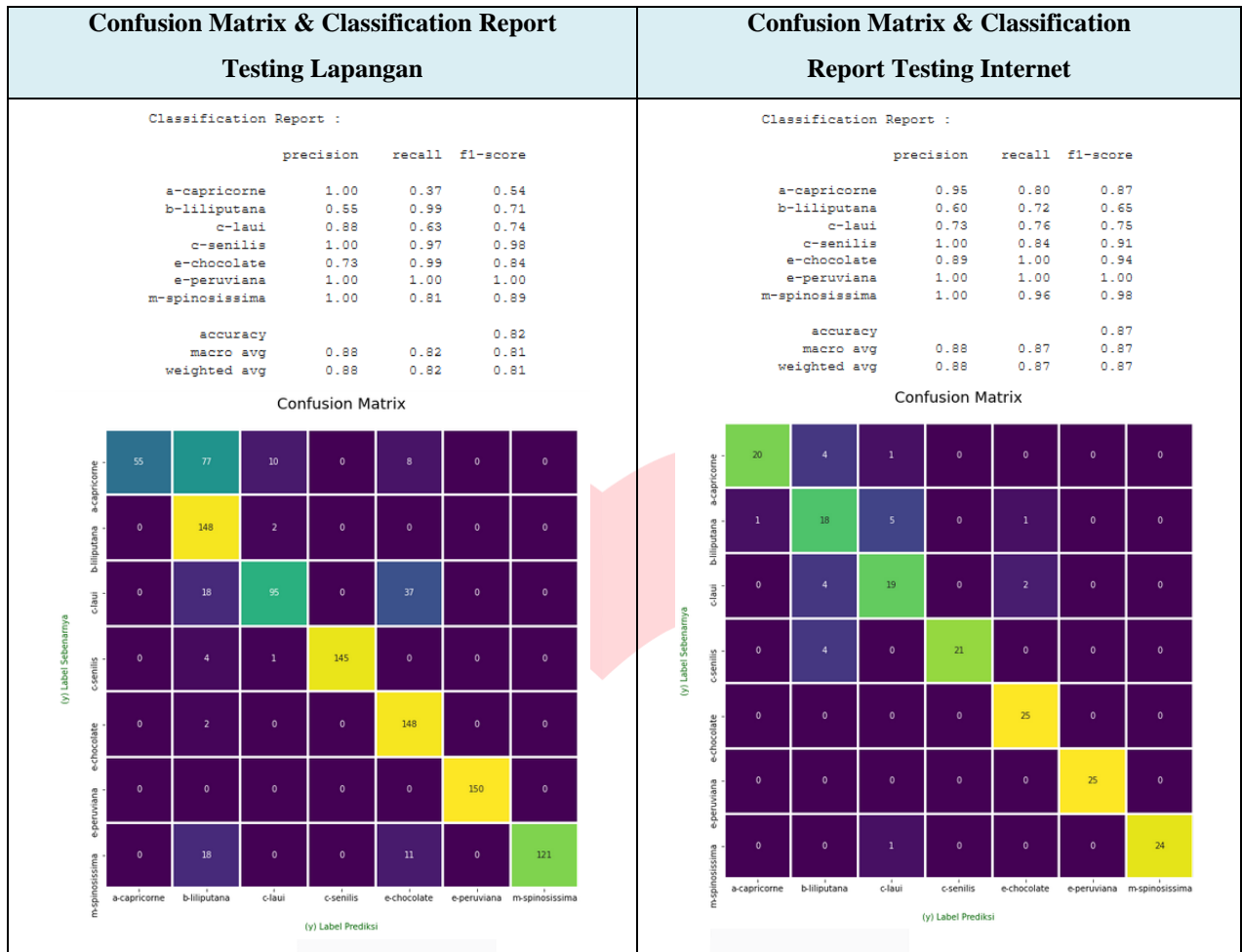
| Analisis Sistem | Data Train | Data Internet |
|---|---|---|
| <p>Hasil Prediksi: p-warasii Gambar Ke-18</p> <p>Hasil Prediksi: g-baldianum Gambar Ke-19</p> <p>Hasil Prediksi: g-baldianum Gambar Ke-20</p> <p>Hasil Prediksi: g-baldianum Gambar Ke-21</p> <p>Hasil Prediksi: g-baldianum Gambar Ke-22</p> <p>Hasil Prediksi: g-baldianum Gambar Ke-23</p> <p>Hasil Prediksi: g-baldianum Gambar Ke-24</p> |  |  |
| Data <i>Parodia Warasii</i> Dengan urutan gambar ke-18 pada data Testing Internet | <i>Gymnocalycium Baldianum</i> | <i>Parodia Warasii</i> |

Gambar 25. Contoh kemiripan data Train dan data Testing Internet

b. Hasil testing pada data internet lapangan yang dapat mempengaruhi akurasi merupakan kesamaan bentuk yang dimana dari total 25 data spesies *Gymnocalycium Baldianum*, 25 data spesies *Mammillaria Spinosissima cv. Un Pico* dan 25 data spesies *Opuntia Microdasys* masing-masing 1 diantaranya di prediksi salah dan di anggap spesies *Parodia Warasii* oleh model, hal ini menyimpulkan sangat kuat bahwa bentuk yang memiliki tingkat kemiripan yang sama bisa mempengaruhi *score* akurasi.



4.2.2 Model Versi 2

Model terbaik pada versi 2 yaitu versi dimana 7 spesies tanaman kaktus *Grafting* yang dimana 2 spesies diantaranya diambil dari data *train* versi 1 agar menambah variatif data memiliki warna yang berbeda dan setiap 2-3 spesiesnya memiliki warna yang sama, diuji dengan 4 skenario *background*, model terbaiknya dihasilkan pada fold ke-5 dengan skenario *background* putih yang mencapai akurasi *Training* 93.83%, *Validation* 99.46%, Testing Lapangan 82.10%, dan Tesing Internet 86.86%. Untuk detail visualnya sebagai berikut :



Gambar 26. Confusion Matrix dan Classification Report model CNN terbaik Versi 2 Skenario Background putih.

Kemudian tahap selanjutnya untuk mengetahui data prediksi yang salah di klasifikasikan oleh model, penulis melakukan analisa pengujian dengan program untuk mengetahui data spesies yang di prediksi salah. Berikut contoh Visualiasasi 1 dari 77 data *Astrophytum Capricorne* yang dianggap Blossfeldia Liliputana oleh model :



| Analisis Sistem | Data Train | Data Lapangan |
|---|---|---|
| <pre> Hasil Prediksi: b-liliputana Gambar Ke-86 Hasil Prediksi: a-capricorne Gambar Ke-87 Hasil Prediksi: a-capricorne Gambar Ke-88 Hasil Prediksi: b-liliputana Gambar Ke-89 Hasil Prediksi: e-chocolate Gambar Ke-90 </pre> |  |  |
| Data <i>Blossfeldia Liliputana</i> Dengan urutan gambar ke-87 pada data Testing Lapangan | <i>Astrophytum Capricorne</i> | <i>Blossfeldia Liliputana</i> |

Gambar 27. Contoh kemiripan data Train dan data Testing Lapangan

Berdasarkan hasil informasi di atas berikut adalah beberapa hal yang dapat disimpulkan :

- a. Hasil testing pada data lapangan yang dapat mempengaruhi akurasi merupakan kesamaan bentuk dan warna yang dimana dari total 150 data spesies *Astrophytum Capricorne*, 77 diantaranya diprediksi salah dan di anggap

spesies *Blossfeldia Liliputana* hal ini kemungkinan akibat kemiripan bentuk dan warna citra. Kemudian analisa pengujian dengan program untuk mengetahui data spesies yang di prediksi salah mengklasifikasikan juga dilakukan pada data internet. Berikut contoh Visualiasasi salah satu spesies yang di prediksi salah mengklasifikasikan oleh model :

| Analisis Sistem | Data Train | Data Internet |
|---|---|---|
| <p>Hasil Prediksi: b-liliputana Gambar Ke-8</p> <p>Hasil Prediksi: b-liliputana Gambar Ke-9</p> <p>Hasil Prediksi: b-liliputana Gambar Ke-10</p> <p>Hasil Prediksi: b-liliputana Gambar Ke-11</p> <p>Hasil Prediksi: c-lalui Gambar Ke-12</p> |  |  |
| Data <i>Copiapoa Laui</i> Dengan urutan gambar ke-87 pada data Testing Internet | <i>Blossfeldia Liliputana</i> | <i>Copiapoa Laui</i> |

Gambar 28. Contoh kemiripan data Train dan data Testing Lapangan

b. Hasil testing pada data internet lapangan yang dapat mempengaruhi akurasi merupakan kesamaan bentuk yang dimana dari total 25 data spesies *Copiapoa Laui*, 4 diantaranya diprediksi salah dan di anggap spesies *Blossfeldia Liliputana* dan dari total 25 data spesies *Blossfeldia Liliputana*, 5 diantaranya diprediksi salah dan di anggap spesies *Copiapoa Laui*, kedua spesies ini yang paling banyak dianggap sama oleh model, hal ini menyimpulkan sangat kuat bahwa bentuk yang memiliki tingkat kemiripan yang sama bisa mempengaruhi *score* akurasi.

5. Kesimpulan

Dari penelitian yang telah dilakukan dalam membangun sebuah model yang mampu mengklasifikasikan spesies tanaman kaktus *Grafting* berdasarkan *Scion* yang paling banyak dibudidayakan dilembang, dapat ditarik kesimpulan bahwa metode *Convolutional Neural Network* (CNN) mampu mengklasifikasikan dengan baik pada studi kasus untuk mengklasifikasikan kaktus *Grafting* dengan citra *Scion* dan dengan akurasi yang cukup baik yaitu :

- Versi 1 : *Training* 95.23%, *Validation* 99.55%, *Testing Lapangan* 94.13%, dan *Tesing Internet* 97.60%.
- Versi 2 : *Training* 93.83%, *Validation* 99.46%, *Testing Lapangan* 82.10%, dan *Tesing Internet* 86.86% pada versi 2.

dengan sama-sama menggunakan skenario *background* putih dengan arsitektur ResNet152V2. Dengan catatan disini penulis harus mengatur *hyperparameter* agar model dapat mempelajari citra dengan baik dan benar. Kemudian pada penelitian ini juga membuktikan kesimpulan lainnya bahwa :

- a. Arsitektur ResNet152v2 terbukti menghasilkan akurasi yang tinggi pada kasus klasifikasi kaktus *Grafting* berdasarkan citra *Scion*.
- b. Untuk mengatasi masalah *overfitting* penggunaan *hyperparameter* perlu dilakukan agar hasil akurasi tepat dan tidak terjadi *overfitting*.
- c. Skenario *background* mempengaruhi akurasi yang dihasilkan, dan hasil terbaik terdapat pada penggunaan *background* putih dibandingkan *background* lain yang diuji cobakan.
- d. Bentuk dan persamaan warna yang memiliki kesamaan yang signifikan dapat mempersulit model dalam mengenali objek yang diklasifikasikan namun hal ini dapat diatasi dengan mengatur *hyperparameter* yang tepat.
- e. Banyaknya kesalahan prediksi disebabkan oleh kemiripan dari segi bentuk dan warna yang hampir sama terbukti dengan pengujian Versi 1 dan Versi 2 pada confusion matrix.

Saran untuk penelitian selanjutnya, menambah jumlah dataset sehingga dapat mengklasifikasikan spesies-spesies kaktus *Grafting* lainnya dan karna pada penelitian ini skenario pengujian *background* terbaik adalah *background* putih sehingga untuk penelitian kedepannya disarankan menggunakan dataset *train background* putih untuk kasus-kasus lainnya yang akan diteliti khususnya pada kasus pengklasifikasian suatu objek. Kemudian harapannya model yang telah dibuat dapat diimplementasikan pada sistem berbasis android ataupun website

sehingga dapat digunakan secara umum khususnya masyarakat yang ingin mengenal kaktus grafting maupun pada penelitian-penelitian lain yang akan diteliti.

REFERENSI

- [1] E. Pérez-Molphe-Balch, M. D. S. Santos-Díaz, R. Ramírez-Malagón, and N. Ochoa-Alejo, "Tissue culture of ornamental cacti," *Sci. Agric.*, vol. 72, no. 6, pp. 540–561, 2015.
- [2] M. Il Jeong, C. Cho, and J. Lee, "Production and Breeding of Cacti for Grafting in Korea," pp. 1–11, 2003.
- [3] F. Davies, R. Geneve, S. Wilson, and H. Hudson, "Principles of grafting and budding," *Hartmann Kester's Plant Propag. Princ. Pract.*, no. 110, pp. 415–463, 2017.
- [4] W. S. Eka Putra, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101," *J. Tek. ITS*, vol. 5, no. 1, 2016.
- [5] E. . H. Alex, Krizhevsky; Ilya, Sutskever; Geoffrey, "222. ImageNet Classification with Deep Convolutional Neural Network.pdf." .
- [6] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020.
- [7] E. N. Arrofiqoh and H. Harintaka, "Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi," *Geomatika*, vol. 24, no. 2, p. 61, 2018.
- [8] E. López-Jiménez, J. I. Vasquez-Gomez, M. A. Sanchez-Acevedo, J. C. Herrera-Lozada, and A. V. Uriarte-Arcia, "Columnar cactus recognition in aerial images using a deep learning approach," *Ecol. Inform.*, vol. 52, no. May, pp. 131–138, 2019.
- [9] S. S. A.-N. Belal A.M. Ashqar, Bassem S. Abu-Nasser, "Plant Seedlings Classification Using Deep Learning," *Int. J. Acad. Inf. Syst. Res.*, vol. 46, no. 3, pp. 745–749, 2019.
- [10] G. Latif, J. Alghazo, R. Maheswar, V. Vijayakumar, and M. Butt, "Deep learning based intelligence cognitive vision drone for automatic plant diseases identification and spraying," *J. Intell. Fuzzy Syst.*, vol. 39, no. 6, pp. 8103–8114, 2020.
- [11] C. Neural, N. Cnn, M. I. Syahputra, and A. T. Wibowo, "Klasifikasi Genus Tanaman Anggrek berdasarkan Citra Kuntum Bunga Menggunakan Metode Convolutional Neural Network (CNN)," pp. 1–9.
- [12] Institute of Electrical and Electronics Engineers, "Simple Convolutional Neural Network on Image Classification," pp. 721–724, 2017.
- [13] M. B. Herlambang, "Deep Learning: Convolutional Neural Networks," 2019. [Online]. Available: <https://www.megabagus.id/deep-learning-convolutional-neural-networks/4/>.
- [14] M. B. Herlambang, "Deep Learning: Convolutional Neural Networks," p. 5, 2019.
- [15] K. H. Mahmud, Adiwijaya, and S. Al Faraby, "Klasifikasi Citra Multi-Kelas Menggunakan Convolutional Neural Network," *e-Proceeding Eng.*, vol. 6, no. 1, pp. 2127–2136, 2019.
- [16] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *Proc. 2017 Int. Conf. Eng. Technol. ICET 2017*, vol. 2018-Janua, pp. 1–6, 2018.
- [17] Y. Lecun, L. Bottou, Y. Bengio, and P. Ha, "Gradient-based learning applied to document recognition," *Proc. IEEE*, no. November, pp. 1–46, 1998.
- [18] M. B. Herlambang, "Deep Learning: Convolutional Neural Networks," 2019. [Online]. Available: <https://www.megabagus.id/deep-learning-convolutional-neural-networks/7/>.
- [19] M. R. Narasinga Rao, V. Venkatesh Prasad, P. Sai Teja, M. Zindavali, and O. Phanindra Reddy, "A survey on prevention of overfitting in convolution neural networks using machine learning techniques," *Int. J. Eng. Technol.*, vol. 7, no. 2.32 Special Issue 32, pp. 177–180, 2018.
- [20] J. Brownlee, "A Gentle Introduction to k-fold Cross-Validation," 2018. [Online]. Available: <https://machinelearningmastery.com/k-fold-cross-validation/>.
- [21] S. Kumar, "Overview of various Optimizers in Neural Networks," 2020. [Online]. Available: <https://towardsdatascience.com/overview-of-various-optimizers-in-neural-networks-17c1be2df6d5>.
- [22] C. Ranjan, "Rules-of-thumb for building a Neural Network," 2019.
- [23] G. Latif, J. Alghazo, R. Maheswar, V. Vijayakumar, and M. Butt, "Deep learning based intelligence cognitive vision drone for automatic plant diseases identification and spraying," *J. Intell. Fuzzy Syst.*, pp. 1–12, 2020.