

Identifikasi Jenis Anjing Berdasarkan Gambar Menggunakan *Convolutional Neural Network* Berbasis Android

Kevin Oktovio Lauw¹, Leo Willyanto Santoso², Rolly Intan³

Program Studi Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121-131 Surabaya 60236, Indonesia

Telp. (031) 2983455, Fax. (031)8417658

E-mail: kevinlauw13@gmail.com¹, leow@petra.ac.id², rintan@petra.ac.id³

ABSTRAK

Anjing banyak dipelihara oleh masyarakat akan tetapi, untuk memelihara seekor anjing, terdapat beberapa faktor yang harus diperhatikan seperti pakan yang dikonsumsi, intensitas perawatan, serta kebersihan kandang atau lingkungan yang sesuai. Oleh karena itu, diperlukan sebuah aplikasi android untuk mengidentifikasi jenis anjing dan memberikan informasi yang berkaitan dengan jenis anjing tersebut.

Metode yang digunakan adalah *You Only Look Once* untuk mendeteksi objek anjing pada gambar kemudian gambar anjing tersebut dipotong, hasilnya akan diolah oleh *Convolutional Neural Network* untuk mengidentifikasi jenis anjing berdasarkan gambar yang diberikan setelah itu menampilkan hasil identifikasinya pada android.

Hasil pengujian menunjukkan bahwa hasil identifikasi dari *CNN* sangat bergantung pada hasil prediksi dari *YOLO* dikarenakan *input* dari *CNN* merupakan hasil prediksi dari *YOLO*. *YOLO* mempunyai kekurangan dimana akan mendeksi boneka anjing dan bulu anjing sebagai objek anjing. Hasil pengujian menunjukkan akurasi dari *YOLO* untuk mendeteksi anjing sebesar 94,242%, akurasi *CNN* model I sebesar 56,400%, akurasi *CNN* model II sebesar 40,000% dan akurasi *CNN* model III 50,400%.

Kata Kunci: *Convolutional Neural Network, You Only Look Once, Tensorflow, Keras, Identifikasi Anjing, Dog Image.*

ABSTRACT

Dogs are raised by many people however, to maintain a dog, there are several factors that must be considered such as feed consumed, intensity of care, and cleanliness of the cage or the appropriate environment. Therefore, an android application is needed to identify the type of dog and provide information related to the type of dog.

The method we used is You Only Look Once to detect dog objects in an image then the dog image is cropped, the results will be processed by the Convolutional Neural Network to identify the type of dog based on the image given after that displaying the results of its identification on android.

The test results show that the identification results from CNN are very dependent on the results of predictions from YOLO because the input from CNN is the result of predictions from YOLO. YOLO has a disadvantage where it will detect dog dolls and dog fur as dog objects. The test results show the accuracy of YOLO to detect dogs is 94.242%, CNN accuracy of model I is 56.400%, accuracy of CNN model II is 40.000% and accuracy of CNN model III is 50.400%.

Keywords: *Machine Learning, Artificial Neural Network, Convolutional Neural Network, YOLO, Faster R-CNN, Image Recognition, Image*

1. PENDAHULUAN

Anjing merupakan salah satu hewan mamalia yang banyak dipelihara dan dicintai oleh masyarakat. Akan tetapi, untuk memelihara seekor anjing, terdapat beberapa faktor yang harus diperhatikan seperti pakan yang dikonsumsi, intensitas perawatan, serta kebersihan kandang atau lingkungan yang sesuai. Faktor-faktor tersebut menjadi sangat penting karena berkaitan erat dengan kesehatan fisik dari seekor anjing. Seringkali para pemilik anjing mengabaikan hal tersebut karena kurangnya pengetahuan mengenai dampak yang dapat terjadi dan merugikan baik bagi pemilik maupun anjing itu sendiri [12]. *Computer vision* mempunyai tujuan utama untuk membuat keputusan yang berguna tentang objek fisik nyata dan pemandangan (*scenes*) berdasarkan *image* yang didapat dari sensor. Sedangkan, *Image processing* merupakan salah satu jenis teknologi untuk menyelesaikan masalah mengenai pemrosesan gambar. Dalam *image processing* gambar yang ada diolah sedemikian rupa sehingga gambar tersebut lebih mudah untuk diproses [13]. Dibandingkan dengan *face recognition*, mendeteksi anjing memiliki perbedaan antar spesies yang lebih besar dan lebih menantang dalam pengenalan pola [6]. Oleh karena itu, akan dibuat sebuah aplikasi untuk mengidentifikasi jenis anjing, dimana melalui gambar yang diberikan setelah itu akan memberikan informasi tentang jenis anjing tersebut seperti nama jenisnya, sifatnya, dan cara untuk memelihara anjing tersebut, dimana informasi tersebut bisa didapatkan dari pakar anjing atau sumber terpercaya lainnya.

Klasifikasi jenis hewan telah dilakukan menggunakan pendekatan segmentasi dan tanpa segmentasi sebagai tahapan awal. *Context Aware Saliency (CAS)* merupakan metode yang mampu membuat wilayah objek menjadi lebih dominan dibandingkan dengan *background* dalam mode *saliency* sehingga dapat menjadi alternatif pengganti proses segmentasi objek. Fitur bentuk diambil berdasarkan citra hasil *saliency* menggunakan metode *Histogram of Oriented Gradient (HOG)*. Metode *K-Nearest Neighbors (K-NN)* digunakan untuk klasifikasi jenis hewan mamalia berdasarkan fitur *HOG* dari citra *saliency* [9]. Selain itu, algoritma yang juga dapat digunakan untuk pengembangan sistem klasifikasi gambar adalah *Linear Discriminant Analysis (LDA)* dan *Support Vector Machine (SVM)*. *LDA* adalah *feature extraction algorithm* untuk menemukan *hyperplane* yang memisahkan kelas dengan baik. *SVM* adalah algoritma klasifikasi, berdasarkan ide untuk menemukan *hyperplane* yang terbaik membagi dataset ke dalam kelas [10]. Metode *K-Nearest Neighbors* dengan fitur *Context Aware Saliency* dan *Histogram of Oriented Gradient* untuk

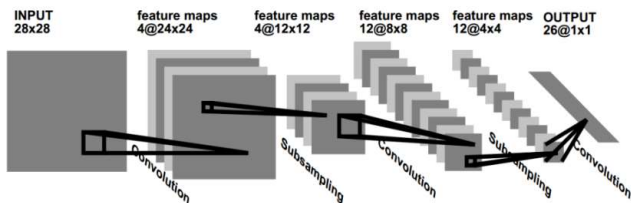
klasifikasi mamalia berdasarkan bentuk wajah menunjukkan bahwa jenis hewan yang dapat dikenali dengan baik adalah kucing dan harimau. Sedangkan, domba, anjing dan babi belum mampu dikenali dengan baik. Algoritma *Linear Discriminant Analysis* dan *Support Vector Machine* dapat mengklasifikasi anjing dan kucing dengan baik akan tetapi tingkat akurasi yang didapat tidak tinggi karena dipengaruhi oleh beberapa faktor seperti jumlah training data, ukuran gambar, kemiripan fitur antara anjing dan kucing, seperti telinga, bentuk tubuh, warna kulit.

Algoritma *Convolutional Neural Network (CNN)* merupakan salah satu metode *Deep learning* yang dapat digunakan untuk mendeteksi dan mengenali sebuah objek pada sebuah citra digital. Kemampuan *CNN* di klaim sebagai model terbaik untuk memecahkan permasalahan *object detection* dan *object recognition* [3]. Selain *CNN*, akan digunakan *You Only Look Once (YOLO)* yang berarti suatu gambar dapat diprediksi objek apa dan di mana mereka pada satu pandangan sekilas [4]. *YOLO* masih tertinggal dibandingkan state-of-the-art detection systems dalam hal akurasi. Akan tetapi *YOLO* dapat dengan cepat mengidentifikasi objek dalam gambar dan secara tepat melokalisasi beberapa objek, terutama yang kecil [8]. Oleh karena itu, pada penelitian ini akan digunakan *YOLO* untuk mengidentifikasi objek dalam gambar yaitu anjing setelah itu, gambar tersebut akan diproses oleh *CNN* untuk mengidentifikasi jenis anjing melalui gambar yang diberikan dan memberikan informasi yang berkaitan seperti nama jenisnya, sifatnya, dan cara untuk memelihara anjing tersebut.

2. DASAR TEORI

2.1 Convolutional Neural Network

Convolutional Neural Network (CNN) termasuk dalam jenis *deep learning* karena kedalaman jaringannya. *Deep learning* adalah cabang dari *machine learning* yang dapat mengajarkan komputer untuk melakukan pekerjaan selayaknya manusia, seperti komputer dapat belajar dari proses *training* [2]. *Convolutional Neural Network* adalah pengembangan dari *Multilayer Perceptron (MLP)* untuk mengolah data dua dimensi. *Multilayer Perceptron* sendiri adalah pengembangan dari *Artificial Neural Network (ANN)* yang ditujukan untuk menutupi kekurangan *Artificial Neural Network (ANN)* dengan *Single-layer Perceptron* dalam menyelesaikan operasi logika yang kompleks. Hal tersebut dimungkinkan dengan menambahkan *hidden layer* yang membuat *ANN* powerful untuk memecahkan operasi logika yang kompleks (*universal approximation*) dan sering digunakan untuk permasalahan-permasalahan klasifikasi, *recognition*, dan prediksi [7]. Model Arsitektur *CNN* dapat dilihat pada Gambar 1.



Gambar 1. Arsitektur *Convolutional Neural Network* [1]

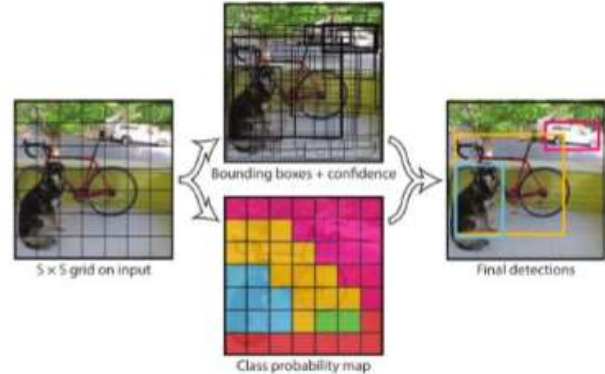
2.2 You Only Look Once

You Only Look Once (YOLO) adalah sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara real-time. Sistem pendeteksian yang dilakukan adalah dengan menggunakan repurpose classifier atau localizer untuk melakukan deteksi. Sebuah model diterapkan pada sebuah citra di beberapa lokasi dan skala.

Daerah dengan citra yang diberi score paling tinggi akan dianggap sebagai sebuah pendeteksian [5].

YOLO menggunakan pendekatan jaringan saraf tiruan untuk mendeteksi objek pada sebuah citra. Jaringan ini membagi citra menjadi beberapa wilayah dan memprediksi setiap kotak pembatas dan probabilitas untuk setiap wilayah. Kotak-kotak pembatas ini kemudian dibandingkan dengan setiap probabilitas yang diprediksi. *Yolo* memiliki beberapa kelebihan dibandingkan dengan sistem yang berorientasi pada classifier, terlihat dari seluruh citra pada saat dilakukan test dengan prediksi yang diinformasikan secara global pada citra [5].

YOLO akan membagi inputan gambar menjadi sebuah $S \times S$ grid. Setiap *grid cell* memprediksi *bounding box* sebanyak B dan *confidence score* untuk setiap *bounding box* tersebut. *Confidence score* merupakan nilai probabilitas *bounding box* yang bersangkutan memiliki sebuah objek. Untuk mendapatkan hasil prediksi berupa *bounding box*, akan dilakukan konvolusi dari inputan gambar, sehingga hasil akhirnya akan mendapat ukuran *bounding box* sebesar $S \times S \times (B * 5 + C)$, dimana B merupakan banyaknya *bounding box* dalam 1 *grid* dan C adalah banyaknya class yang dapat diklasifikasi. Nilai B dikalikan dengan 5 karena dalam sebuah *bounding box* memiliki 5 *value* yang perlu disimpan yaitu: koordinat x , koordinat y , lebar (*width*), tinggi (*height*), dan *confidence score*. Berikutnya akan dilakukan normalisasi untuk setiap atribut pada *bounding box* sehingga nilainya menjadi antara 0 hingga 1. Koordinat x dan y yang dinormalisasi menyesuaikan titik kiri atas dari *grid* yang bersangkutan. Dan *width* dan *height* akan dinormalisasi sesuai dengan ukuran gambar. Nilai koordinat x dan y pada sebuah *bounding box* pada setiap *grid* merupakan titik tengah *grid* yang bersangkutan [11]. Proses *YOLO* dapat dilihat pada Gambar 2.



Gambar 2. *You Only Look Once (YOLO)* [13]

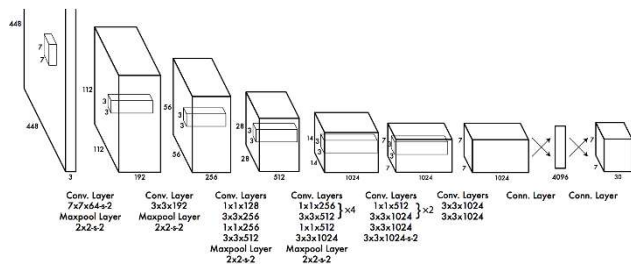
3. DESAIN SISTEM

3.1 Analisis Sistem

Analisis sistem membahas permasalahan bagaimana data yang diberikan dapat diproses sehingga dapat menghasilkan output yang sesuai. Sistem mencakup model arsitektur konvolusi *neural network* untuk *training*

3.1.1 YOLO Network untuk Training Dataset

You Only Look Once (YOLO) network structure mirip dengan model sebuah *Convolutional Neural Network*, dimana *YOLO* mempunyai 24 *convolutional layers* diikuti dengan 2 *fully connected layers (FC)* dan 4 *maxpool layer* untuk mendapatkan *grid-grid* yang digunakan untuk mengklasifikasi *object* atau *class*.

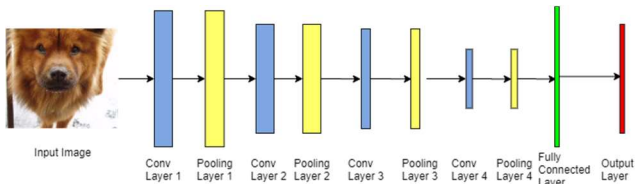


Gambar 3. Arsitektur Konvolusi Neural Network YOLO

Pada arsitektur konvolusi neural network neural network YOLO Gambar 3. dibuat dengan S=7, B=2 dan C=20 sehingga mendapatkan prediksi final output 7 x 7 x 30 tensor. Menggunakan CNN network untuk mengurangi spatial dimension menjadi 7 x 7 dengan output channel di setiap lokasi sebanyak 1024. Beberapa convolution layer menggunakan 1 x 1 reduction layers untuk mengurangi the dept of the features maps. Selain itu, menggunakan 2 fully connected layer sebagai bentuk regresi linier yang menghasilkan output 7 x 7 x 30 dan mengubah bentuk 7 x 7 x 30 menjadi 2 boundary box predictions.

3.1.2 CNN Training Model

CNN menerima inputan gambar yang akan di-resize terlebih dahulu menjadi ukuran 224x224px agar ukuran semua gambar seragam hal ini dilakukan karena CNN hanya dapat menerima input dengan dimensi yang sama. Proses berikutnya dilanjutkan dengan pembuatan model secara sequential. Terdapat empat convolution layers dan max pooling layers serta dua fully connected layers, dimana convolution layer pertama, merupakan layer convolutional 2D menggunakan activation ReLU, zero padding, dan ukuran kernel sebesar 2 serta jumlah filters sebanyak 16 dengan input shape 224x224x3 dimana 224 merupakan width dan height dari gambar input sedangkan 3 merupakan RGB channel. Layer kedua, merupakan max pooling layer dengan pool_size sebesar 2 untuk memastikan tidak kehilangan informasi dalam gambar ketika menurunkan kemungkinan overfitting. Layer berikutnya, merupakan layer convolutional 2D menggunakan activation ReLU, ukuran kernel sebesar 2 dan jumlah filters 32. Layer keempat, merupakan max pooling layer dengan pool_size sebesar 2. Layer kelima, merupakan layer convolutional 2D dengan activation ReLU, ukuran kernel sebesar 2 dan jumlah filters 64. Layer keenam, merupakan max pooling layer dengan pool_size sebesar 2. Layer ketujuh, merupakan layer convolutional 2D dengan activation ReLU, ukuran kernel sebesar 2 dan jumlah filters 128. Layer kedelapan, merupakan max pooling layer dengan pool_size sebesar 2. Dan yang terakhir merupakan 2 fully connected layer dengan activation ReLU dan softmax. Model arsitektur CNN dapat dilihat pada Gambar 4.



Gambar 4. Model Arsitektur CNN

3.2 Desain Aplikasi

Aplikasi dibuat dengan front end menggunakan android serta back end menggunakan bahasa pemrograman python. Front end merupakan aplikasi Android berfungsi sebagai tempat user untuk meng-upload foto anjing baik dari camera maupun gallery mobile phone user. Setelah itu, gambar tersebut akan diupload kedalam server untuk diproses oleh back end. Di back end Script python akan berfungsi sebagai pre-processing image, dimana akan dideteksi apakah gambar yang di upload dari front end terdapat gambar anjing menggunakan YOLO. Jika terdapat gambar anjing selanjutnya gambar tersebut akan di crop gambar anjingnya saja. Selanjutnya, gambar yang sudah di crop akan di resize dan dilakukan testing menggunakan CNN pada gambar anjing yang sudah dideteksi. Setelah gambar diproses, hasil dari proses tersebut akan dikirim ke server dan akan ditampilkan kembali pada front end.

4. IMPLEMENTASI SISTEM

Implementasi sistem dibuat sesuai dengan analisa dan desain sistem. Implementasi sistem dilakukan pada computer dengan spesifikasi GPU NVIDIA GeForce GTX 960M, RAM sebesar 16GB serta Processor Intel® Core™ i7-6700HQ.

Implementasi program menggunakan bahasa pemrograman python dengan versi 3.7.0. Framework yang digunakan untuk sistem ini adalah Tensorflow versi 2.0. Adapun beberapa library yang mendukung sistem ini adalah CV2 dan Numpy

Selain itu aplikasi android dengan menggunakan java untuk meng-upload input gambar ke server dengan menggunakan php akan menjalankan file python menggunakan command exec yang ada pada php untuk melakukan prediksi pada gambar yang di-upload.

5. ANALISA DAN PENGUJIAN

5.1 Pengujian Sistem

Pengujian sistem mengevaluasi performa daripada sistem training dan sistem testing yang telah dibuat. Pencatatan pengujian akan dibedakan berdasarkan jenis anjing yang diprediksi dan diklasifikasikan.

5.1.1 Pengujian Metode YOLO

Pengujian dengan menggunakan metode YOLO dilakukan terhadap 25 class anjing yang berbeda. Dengan jumlah iterasi atau epoch 2000. Untuk mendapatkan akurasi prediksi anjing dapat menggunakan rumus (1). Hasil pengujian dapat dilihat pada Tabel 1.

$$Akurasi A = \frac{Jumlah\ gambar\ anjing\ yang\ berhasil\ terdeteksi}{Total\ jumlah\ gambar\ anjing} \times 100$$

$$Akurasi\ rata - rata = \frac{\sum_{n=1}^{25} Akurasi\ A\ ke - n}{25} \dots\dots\dots (1)$$

Tabel 1. Pengujian prediksi dengan metode YOLO

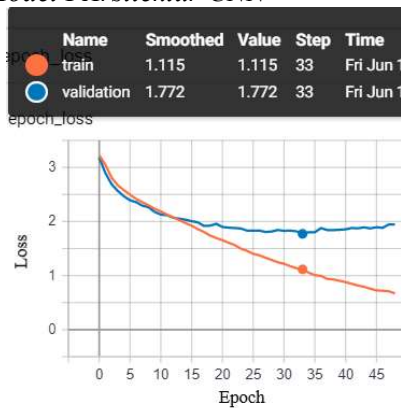
Class	Jumlah Gambar	Terdeteksi	Akurasi A
Airedale	243	235	96,708%
Appenzeller	222	218	98,198%
Basenji	278	260	93,525%
Basset	269	258	95,911%
Beagle	231	227	98,268%

Bostonbull	240	225	93,750%
Boxer	269	253	94,052%
Brittanyspaniel	273	259	94,872%
Cairn	278	269	96,763%
Chihuahua	214	188	87,851%
Chow	215	200	93,023%
Doberman	225	210	93,333%
Goldenretriever	232	219	94,397%
Kuvasz	238	221	92,857%
Maltesedog	312	294	94,231%
Pembroke	241	229	95,021%
Pomeranian	284	263	92,606%
Pug	248	227	91,532%
Redbone	240	220	91,666%
Rottweiler	235	219	93,192%
Samoyed	274	257	93,796%
Shihtzu	287	274	95,470%
Siberianhusky	244	232	95,082%
vizsla	247	232	93,927%
Weimaraner	251	241	96,016%
Rata-rata			94,242%

5.1.2 Pengujian Metode CNN

Pengujian dengan menggunakan metode CNN dilakukan terhadap 25 class anjing yang berbeda untuk mengidentifikasi gambar anjing yang diberikan. Dimana metode CNN ini dibedakan menjadi 3 model arsitektur yang berbeda

5.1.2.1 Model I Arsitektur CNN



Gambar 5. Epoch loss dari train dan validation pada Model I CNN

Gambar 5. merupakan gambar proses dari model yang dibuat dan disimpan untuk digunakan dalam mengidentifikasi 25 class anjing dengan model I arsitektur CNN. Garis orange merupakan nilai loss dari data train sedangkan garis biru merupakan nilai loss dari data validation. Semakin rendah nilai loss maka semakin bagus model tersebut yang akan digunakan untuk mengidentifikasi jenis anjing. Pada epoch yang pertama nilai loss baik train dan validation masih tinggi (diatas 3) dan bergerak turun menjadi lebih baik. Nilai loss

dari validation yang paling bagus terdapat pada epoch ke-34 sebesar 1.772 dan nilai loss dari train sebesar 1.115 dengan menggunakan fitur dari tensorflow yaitu ModelCheckpoint dimana hanya menyimpan model yang bagus dan model pada epoch berikutnya tidak disimpan jika nilai validation loss-nya lebih jelek. Waktu yang dibutuhkan model ke I untuk mencapai epoch ke 50 adalah 1 jam 35 menit 53 detik. Untuk mendapat akurasi mengidentifikasi jenis anjing dengan data testing dapat menggunakan Rumus (2). Hasil pengujian dapat dilihat pada Tabel 2.

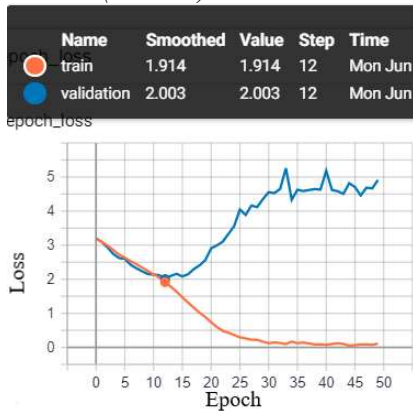
$$Akurasi B = \frac{\text{Jumlah gambar anjing yang berhasil diidentifikasi}}{\text{Total jumlah gambar anjing yang terdeteksi}} \times 100$$

$$Akurasi rata - rata = \frac{\sum_{n=1}^{25} Akurasi B \text{ ke } - n}{25} \dots\dots\dots (2)$$

Tabel 2. Pengujian identifikasi anjing dengan metode CNN model I

Class	Jumlah Gambar	Terdeteksi	Akurasi B
Airedale	10	9	90,000%
Appenzeller	10	7	70,000%
Basenji	10	8	80,000%
Basset	10	3	30,000%
Beagle	10	2	20,000%
Bostonbull	10	8	80,000%
Boxer	10	6	60,000%
Brittanyspaniel	10	7	70,000%
Cairn	10	2	20,000%
Chihuahua	10	1	10,000%
Chow	10	8	80,000%
Doberman	10	7	70,000%
Goldenretriever	10	4	40,000%
Kuvasz	10	5	50,000%
Maltesedog	10	8	80,000%
Pembroke	10	5	50,000%
Pomeranian	10	10	100,000%
Pug	10	3	30,000%
Redbone	10	4	40,000%
Rottweiler	10	10	100,000%
Samoyed	10	6	60,000%
Shihtzu	10	4	40,000%
Siberianhusky	10	5	50,000%
vizsla	10	4	40,000%
Weimaraner	10	5	50,000%
Rata-rata			56,400%

5.1.2.2 Model II (Alexnet) Arsitektur CNN



Gambar 6. Epoch loss dari train dan validation pada Model II CNN

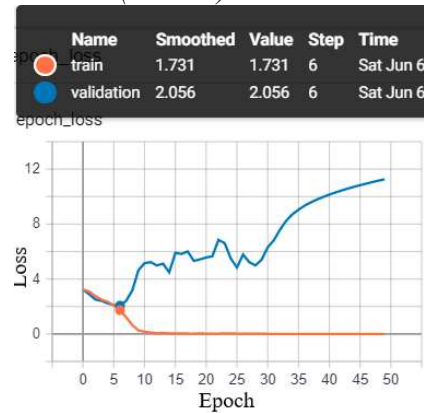
Gambar 6. merupakan gambar proses dari model yang dibuat dan disimpan untuk digunakan dalam mengidentifikasi 25 class anjing dengan model II arsitektur CNN. Pada epoch yang pertama nilai loss baik train dan validation masih tinggi (diatas 3) dan bergerak turun menjadi lebih baik. Nilai loss dari validation yang paling bagus terdapat pada epoch ke-13 sebesar 2.002 dan nilai loss dari train sebesar 1.914. Waktu yang dibutuhkan model ke II untuk mencapai epoch ke 50 adalah 4 jam 32 menit 57 detik. Untuk mendapat akurasi mengidentifikasi jenis anjing dengan data testing dapat menggunakan Rumus (2). Hasil pengujian dapat dilihat pada Tabel 3.

Tabel 3. Pengujian identifikasi anjing dengan metode CNN model II

Class	Jumlah Gambar	Terdeteksi	Akurasi B
Airedale	10	8	80,000%
Appenzeller	10	5	50,000%
Basenji	10	6	60,000%
Basset	10	3	30,000%
Beagle	10	1	10,000%
Bostonbull	10	5	50,000%
Boxer	10	2	20,000%
Brittanyspaniel	10	5	50,000%
Cairn	10	0	0,000%
Chihuahua	10	0	0,000%
Chow	10	7	70,000%
Doberman	10	3	30,000%
Goldenretriever	10	1	10,000%
Kuvasz	10	3	30,000%
Maltesedog	10	7	70,000%
Pembroke	10	4	40,000%
Pomeranian	10	4	40,000%
Pug	10	0	0,000%
Redbone	10	6	60,000%
Rottweiler	10	8	80,000%

Samoyed	10	7	70,000%
Shihtzu	10	3	30,000%
Siberianhusky	10	4	80,000%
vizsla	10	4	50,000%
Weimaraner	10	6	60,000%
Rata-rata			40,000%

5.1.2.3 Model III (VGG16) Arsitektur CNN



Gambar 7. Epoch loss dari train dan validation pada Model III CNN

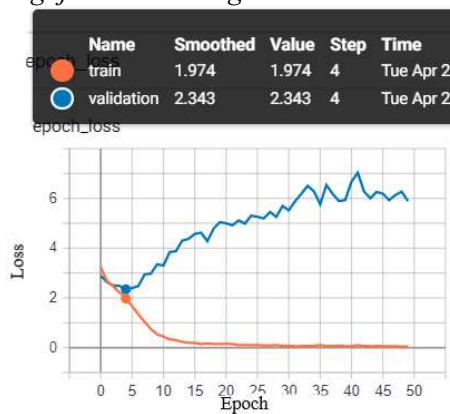
Gambar 7. merupakan gambar proses dari model yang dibuat dan disimpan untuk digunakan dalam mengidentifikasi 25 class anjing dengan model III arsitektur CNN. Pada epoch yang pertama nilai loss baik train dan validation masih tinggi (diatas 3) dan bergerak turun menjadi lebih baik. Nilai loss dari validation yang paling bagus terdapat pada epoch ke-7 sebesar 2.056 dan nilai loss dari train sebesar 1.731. Waktu yang dibutuhkan model ke III untuk mencapai epoch ke 50 adalah 2 hari 13 jam 14 menit 58 detik. Untuk mendapat akurasi mengidentifikasi jenis anjing dengan data testing dapat menggunakan Rumus (2). Hasil pengujian dapat dilihat pada Tabel 4.

Tabel 2. Pengujian identifikasi anjing dengan metode CNN model III

Class	Jumlah Gambar	Terdeteksi	Akurasi B
Airedale	10	8	80,000%
Appenzeller	10	8	80,000%
Basenji	10	1	10,000%
Basset	10	4	40,000%
Beagle	10	2	20,000%
Bostonbull	10	10	100,000%
Boxer	10	5	50,000%
Brittanyspaniel	10	9	90,000%
Cairn	10	1	10,000%
Chihuahua	10	3	30,000%
Chow	10	6	60,000%
Doberman	10	7	70,000%
Goldenretriever	10	6	60,000%
Kuvasz	10	3	30,000%

Maltesedog	10	8	80,000%
Pembroke	10	4	40,000%
Pomeranian	10	7	70,000%
Pug	10	3	30,000%
Redbone	10	1	10,000%
Rottweiler	10	9	90,000%
Samoyed	10	4	40,000%
Shihtzu	10	2	20,000%
Siberianhusky	10	3	30,000%
vizsla	10	9	90,000%
Weimaraner	10	3	30,000%
Rata-rata			50,400%

5.1.3 Pengujian CNN Dengan Penambahan Dataset



Gambar 8. Epoch loss dari train dan validation pada Model I CNN dataset berbeda

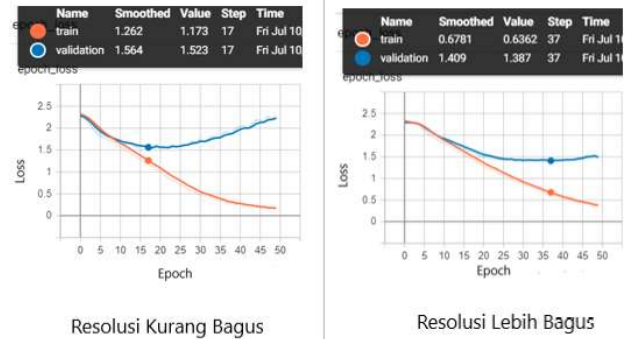
Gambar 8. merupakan gambar proses dari model yang dibuat dan disimpan untuk digunakan dalam mengidentifikasi 25 class anjing dengan model I arsitektur CNN akan tetapi dengan dataset yang belum ditambahkan. Nilai loss dari validation yang paling bagus terdapat pada epoch ke-5 sebesar 2.343 dan nilai loss dari train sebesar 1.974 dengan menggunakan fitur dari tensorflow yaitu ModelCheckpoint dimana hanya menyimpan model yang bagus dan model pada epoch berikutnya tidak disimpan jika nilai validation loss-nya lebih jelek.

Tabel 5. Perbandingan akurasi dengan dataset yang berbeda

Model I	Jumlah Gambar	Validation Loss	Jumlah Testing	Akurasi
Dataset lama	4158	1.974	250	32.800%
Dataset baru	5930	1.772	250	56.400%

Tabel 5. memberikan hasil testing identifikasi anjing dengan dataset yang sudah ditambah memiliki akurasi lebih baik dari pada menggunakan dataset lama. Pengujian dilakukan dengan menambah variasi gambar anjing dari 25 class, dimana mempunyai resolusi gambar yang lebih bagus dari pada gambar anjing dataset lama serta mengurangi overfitting.

5.1.4 Pengujian CNN Dengan Resolusi Dataset yang Berbeda



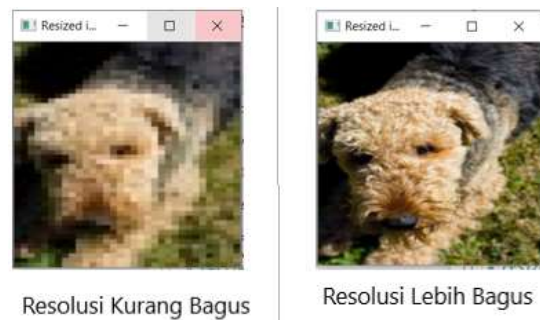
Gambar 9. Grafik Loss dari hasil training CNN

Gambar 9. merupakan gambar proses dari 2 model yang dibuat dan disimpan untuk digunakan dalam mengidentifikasi class anjing menggunakan arsitektur CNN model I dengan menggunakan dataset yang sama tetapi dengan resolusi yang berbeda. Garis orange merupakan nilai loss dari data train sedangkan garis biru merupakan nilai loss dari data validation. Perbandingan akurasi antara dataset dengan resolusi kurang bagus dan resolusi yang lebih bagus dapat dilihat pada Tabel 6.

Tabel 6. Perbandingan akurasi dengan resolusi dataset yang berbeda

Model I	Jumlah Gambar	Validation Loss	Jumlah Testing	Akurasi
Resolusi kurang bagus	2529	1.523	100	54.000%
Resolusi lebih bagus	2529	1.387	100	72.000%

Tabel 6. memberikan hasil testing identifikasi anjing dengan resolusi gambar dataset yang lebih bagus dapat meningkatkan akurasi dalam mengidentifikasi jenis anjing. Pengujian dilakukan pada 10 class jenis anjing dengan gambar yang sama akan tetapi beda resolusinya. Contoh perbandingan resolusi gambar yang sudah di-resize menjadi 224x224 pixel dapat dilihat pada Gambar 10.



Gambar 10. Contoh perbandingan resolusi gambar pada dataset yang sama

5.2 Pengujian Aplikasi

Pengujian aplikasi dilakukan dengan mencoba melakukan prediksi dan klasifikasi berdasarkan *user interface* yang telah dibuat. Pengujian dilakukan pada masing-masing jenis anjing dan gambar yang tidak ada objek anjing untuk melihat bagaimana hasil dari prediksi dan klasifikasi pada *user interface* yang telah dibuat. Tampilan *user interface* dapat dilihat pada Gambar 11.



Gambar 11. Gambar User Interface

6. KESIMPULAN

Setelah dilakukan perancangan sistem, pengimplementasian, dan pengujian terhadap aplikasi yang telah dibuat, dapat ditarik kesimpulan sebagai berikut:

- Berdasarkan hasil pengujian pada sistem dan aplikasi dapat disimpulkan bahwa *CNN* sangat bergantung pada hasil dari prediksi *YOLO* dalam mengenali objek anjing karena *input CNN* berasal dari hasil prediksi *YOLO*.
- Berdasarkan hasil pengujian dapat disimpulkan bahwa dari tiga model arsitektur *CNN* yang digunakan, model I merupakan model yang lebih bagus dari model II dan model III.
- Berdasarkan hasil pengujian dapat disimpulkan bahwa nilai akurasi *YOLO* mempunyai rata-rata sebesar 94,242%, akurasi *CNN* model I sebesar 56,400%, akurasi *CNN* model II sebesar 40,000% dan akurasi *CNN* model III 50,400%.
- Berdasarkan hasil pengujian dapat disimpulkan penambahan *dataset* yang digunakan untuk *training* dengan resolusi gambar yang bagus dapat membuat model yang dipakai lebih baik dan mengurangi *overfitting*.
- Berdasarkan hasil pengujian waktu yang dibutuhkan untuk mengidentifikasi anjing oleh *CNN* model I selama 9 sampai 11 detik, *CNN* model II selama 11 sampai 13 detik dan *CNN* model III selama 44 sampai 49 detik.
- *YOLO* mempunyai kekurangan dimana *YOLO* akan mendeteksi objek boneka anjing dan objek bulu anjing sebagai objek anjing.

7. DAFTAR PUSTAKA

- [1] Abhirawa, H. 2017. Pengenalan Wajah Menggunakan Convolutional Neural Network. *e-Proceeding of Engineering*, 4907-4916.
URI= <http://repositori.usu.ac.id/handle/123456789/15450>
- [2] Arrofiqoh, E. N., & Harintaka. 2018. Implementasi Metode Convolutional Neural Network untuk Klasifikasi Tanaman pada Citra Resolusi Tinggi. *Geomatika*, 61-68.
URI= <http://garuda.ristekbrin.go.id/documents/detail/843166>

- [3] Dewa, C. K. (2018). Convolutional Neural Networks for Handwritten Javanese Character Recognition. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 83-94.
DOI= <https://doi.org/10.22146/ijccs.31144>
- [4] Du, J. 2018. Understanding of Object Detection Based on CNN Family and YOLO. *Journal of Physics: Conference Series*, 1-8.
URI= <https://iopscience.iop.org/article/10.1088/1742-6596/1004/1/012029>
- [5] Jupiyandi, S., Saniputra, F. R., Pratama, Y., & Dharmawan, M. R. 2019. Pengembangan Deteksi Citra Mobil Untuk Mengetahui Jumlah Tempat Parkir Menggunakan Cuda dan Modified YOLO. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 413-419.
URI= <http://jtiik.ub.ac.id/index.php/jtiik/article/view/1275/pdf>
- [6] Ouyang, J., He, H., He, Y., & Tang, H. 2019. Dog recognition in public places based on convolutional neural network. *International Journal of Distributed Sensor Networks*, 1-9.
URI= <https://journals.sagepub.com/doi/10.1177/1550147719829675?icid=int.sj-full-text.similar-articles.2>
- [7] Pangestu, M. A., & Bunyamin, H. 2018. Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan Pre-Trained CNN Model. *Jurnal Teknik Informatika dan Sistem Informasi*, 337-344.
URI= <https://journal.maranatha.edu/index.php/jutisi/article/view/1501>
- [8] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2015. You Only Look Once: Unified, Real-Time Object Detection.
URI= <https://arxiv.org/abs/1506.02640>
- [9] Rivan, M. E., & Yohannes. 2019. Klasifikasi Mamalia Berdasarkan Bentuk Wajah Dengan K-NN Menggunakan Fitur CAS Dan HOG. *Jurnal Teknik Informatika dan Sistem Informasi*, 173-180.
URI= <https://doi.org/10.35957/jatisi.v5i2.139>
- [10] Salim, S. D., & Suryadibrata, A. 2019. Klasifikasi Anjing dan Kucing menggunakan Algoritma Linear Discriminant Analysis dan Support Vector Machine. *ULTIMATICS*, 46-51.
URI= <https://doi.org/10.31937/ti.v1i1.1076>
- [11] Shianto, K. A. 2019. Deteksi Jenis Mobil Menggunakan Metode YOLO Dan Faster R-CNN. *Jurnal Infra Vol 7, No 1*, 157-163.
URI= <http://publication.petra.ac.id/index.php/teknik-informatika/article/view/8065>
- [12] Ulfa, Z., Elfidasari, D., & Sugoro, I. 2016. Identifikasi Khamir Patogen pada Kulit dan Telinga Anjing Peliharaan. *AL-AZHAR INDONESIA SERI SAINS DAN TEKNOLOGI*, 213-220.
URI= <https://jurnal.uai.ac.id/index.php/SST/article/view/236>
- [13] Wirawan, L. V. 2002. Sistem pengenalan plat nomor kendaraan bermotor dengan metode principal components analysis.
URI= <http://jurnalelektro.petra.ac.id/index.php/elk/article/view>