

RANCANG BANGUN APLIKASI IDENTIFIKASI BURUNG DILINDUNGI MELALUI CITRA DIGITAL MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN)

Danny Gunawan

Teknik Informatika
Universitas Ma Chung, Villa Puncak Tidar N-1 Malang
email: 311610003@student.machung.ac.id

Abstrak

Indonesia memiliki keanekaragaman flora dan fauna. Salah satu fauna yang dilindungi yaitu fauna burung. Di Indonesia terdapat 1.777 spesies burung dimana beberapa dari spesies tersebut adalah spesies endemik Indonesia. Perburuan dan perdagangan ilegal diduga menjadi penyebab utama kepunahan beberapa spesies langka sehingga oleh negara spesies tersebut dimasukkan dalam kategori dilindungi negara. Terdapat 532 spesies burung yang dilindungi negara pada tahun 2018. Penyebaran informasi yang baik dapat menjadi salah satu cara untuk membatasi ruang gerak perdagangan ilegal spesies-spesies tersebut. Dalam penelitian ini akan dirancang dan dikembangkan sebuah prototipe aplikasi berbasis perangkat bergerak. Aplikasi ini ditujukan untuk mengenali spesies-spesies burung dilindungi negara menggunakan Convolutional Neural Network (CNN) dengan arsitektur LeNet dan AlexNet. Pada arsitektur yang sudah dirancang menggunakan fungsi optimasi Adam dan Nadam. Dari hasil penelitian yang sudah dilakukan, aplikasi yang dirancang dapat bekerja dengan baik dengan tingkat akurasi mencapai 92.65% dalam waktu 17 menit. Dataset yang digunakan pada penelitian ini berjumlah 2.925 citra burung dilindungi dimana 80% dari data tersebut digunakan untuk proses training dan sisanya 20% digunakan untuk validation.

Kata kunci: Burung, Dilindungi, AlexNet, LeNet, Adam, Nadam

Abstract

Indonesia has a diversity of flora and fauna. One of endangered fauna is the bird fauna. In Indonesia there are 1,777 species of birds with several species being Indonesian endemic species. Because of the beauty of these birds, poaching and trade happen so birds are endangered and then a lot of their status become protected by the state. Therefore, the author designed an application based on mobile devices that can recognize the bird which protected by the State. This application aims to be consideration to help bird lovers for maintaining birds that status is protected. In this study using the Convolutional Neural Network (CNN) method with LeNet and AlexNet architecture. The optimization function used is Adam and Nadam. Based on the results of the discussion obtained the level of accuracy using the LeNet architecture and the Nadam optimization function of 99.83% in the training process and 92.65% in the testing process. The training process takes 17 minutes. The dataset used was 2,925 bird images protected with 80% scenarios for training and 20% for validation.

Keywords: Bird, Protected, AlexNet, LeNet, Adam, Nadam

1. Pendahuluan

Indonesia merupakan negara kepulauan yang sangat luas. Membuat Indonesia memiliki keanekaragaman flora dan fauna dengan ciri khas masing-masing di setiap daerah. Salah satu fauna yang ada di Indonesia adalah fauna burung dengan total 1.777 spesies pada tahun 2019 (Edwin, 2019) dimana beberapa dari spesies tersebut adalah spesies endemik Indonesia. Fauna burung juga termasuk fauna yang dilindungi negara. Perburuan dan perdagangan ilegal diduga menjadi penyebab utama kepunahan beberapa spesies langka sehingga oleh negara spesies tersebut dimasukkan dalam kategori dilindungi negara. Maka untuk memudahkan masyarakat mengenali jenis burung dilindungi diperlukan suatu aplikasi yang memanfaatkan gambar atau citra. Dengan gambar tentu banyak cara untuk menyampaikan kepada masyarakat contohnya seperti majalah tetapi dinilai kurang efisien, ada metode yang lebih modern untuk klasifikasi citra.

Metode yang bisa digunakan untuk melakukan klasifikasi pada citra digital, salah satunya dengan *machine learning*. *Machine learning* merupakan bidang studi yang memanfaatkan prinsip ilmu komputer dan statistika untuk membuat model statistik (Sofia, 2018). Bertujuan dapat melakukan prediksi yang baik di masa depan berdasarkan kumpulan data. Salah satu metode machine learning yaitu Convolutional Neural Network (CNN). CNN merupakan pengembangan dari multilayer perceptron (MLP) yang didesain untuk mengolah data dua dimensi dalam bentuk citra. CNN berusaha meniru sistem pengenalan pada visual cortex manusia, sehingga memiliki kemampuan mengolah informasi citra (Putra et al., 2016). CNN adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Masukan (input) dan keluaran (output) dari setiap tahap adalah terdiri dari beberapa array yang biasa disebut feature map (Nurhikmat, 2018). CNN dapat diterapkan untuk melakukan klasifikasi gambar dengan mengekstrak fitur-fitur yang spesifik dari gambar dalam arsitekturnya secara otomatis. Nantinya akan menghasilkan sebuah pola umum yang penting dalam membedakan setiap objek.

2. Tinjauan Pustaka

2.1 Burung

Burung adalah anggota kelompok hewan bertulang belakang yang memiliki bulu dan sayap. Perburuan dan perdagangan ilegal diduga menjadi penyebab utama

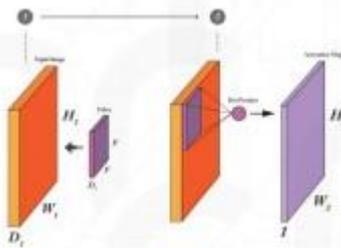
kepunahan beberapa spesies langka sehingga oleh negara spesies tersebut dimasukkan dalam kategori dilindungi negara. Menurut lampiran peraturan Menteri Lingkungan Hidup dan Kehutanan Republik Indonesia Nomer P.20/MENLHK/SETJEN/KUM.1/6/2018 tentang penetapan jenis tumbuhan dan satwa yang dilindungi ada 532 jenis burung dengan status dilindungi. Satwa yang dilindungi oleh pemerintah diatur dalam Undang-Undang Nomer 5 Tahun 1990 tentang Konservasi Sumber Daya Alam Hayati dan Ekosistemnya. Undang-undang tersebut menjelaskan bahwa setiap orang dilarang untuk menangkap, melukai, membunuh, menyimpan, memiliki, memelihara, mengangkut, dan memperniagakan satwa yang dilindungi dalam keadaan hidup. Jika melanggar terhadap ketentuan yang sudah ada maka akan dikenakan sanksi pidana penjara paling lama 5 tahun dan dikenakan denda paling banyak Rp 100 juta rupiah yang dijelaskan dalam Pasal 40 ayat (2) UU 5/1990.

2.2 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah salah satu metode machine learning dari pengembangan *Multi Layer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena dalamnya tingkat jaringan dan banyak diimplementasikan dalam data citra (Sofia, 2018). CNN adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Masukan (*input*) dan keluaran (*output*) dari setiap tahap adalah terdiri dari beberapa array yang biasa disebut *feature map* (Nurhikmat, 2018). CNN menjadi dua bagian penting yaitu *feature learning* dan *classification*. Pada bagian *feature learning* terdiri dari dua tahapan yaitu *convolutional layer* dan *pooling layer*.

2.3 Convolutional Layer

Convolutional layer adalah tahap awal setelah input layer pada arsitektur CNN. Proses ini memanfaatkan apa yang disebut dengan filter. Filter ini akan bergeser ke seluruh bagian dari gambar. Pergeseran tersebut akan melakukan operasi dot antara *input* dan nilai dari filter tersebut sehingga akan menghasilkan *output* yang disebut sebagai *activation map* atau *feature map* (Nurfita, 2018).



Gambar 1. Proses Pada Convolutional Layer

(sumber:

<https://medium.com/nodeflux/mengenalconvolutional-layer-dan-pooling-layer-3c6f5c393ab2>)

$$\text{Output Volume} = \frac{W - F + 2P}{S} + 1 \quad (1)$$

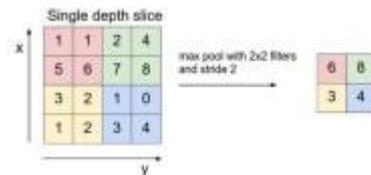
Berdasarkan persamaan di atas, dapat dihitung ukuran spasial dari volume *output* di mana *hyperparameter* yang dipakai adalah ukuran volume (W), filter (F), Stride yang diterapkan (S) dan jumlah padding nol yang digunakan (P). Stride merupakan nilai yang digunakan untuk menggeser filter melalui *input* citra dan *zero padding* adalah nilai untuk mendapatkan angka nol di sekitar border citra.

2.4 Aktivasi ReLU

Aktivasi ReLU (*Rectified Linear Unit*) merupakan lapisan aktivasi pada model CNN. Fungsi ini melakukan *thresholding* dengan nilai nol terhadap nilai piksel pada *input* citra. Aktivasi ini membuat seluruh nilai piksel yang bernilai kurang dari nol pada suatu citra akan dijadikan 0.

2.5 Operasi Pooling

Pooling ini adalah untuk mereduksi *input* secara spasial (mengurangi jumlah parameter) dengan operasi *downsampling*. Metode pooling yang biasanya digunakan adalah *maxpooling* atau mengambil nilai terbesar. Metode lainnya yang bisa digunakan seperti *average-pooling* atau mengambil nilai rata-ratanya.



Gambar 2. Operasi Max-Pooling (sumber:

[https://medium.com/@samuelsena/pengenalandeepl earning-part-7-](https://medium.com/@samuelsena/pengenalandeepl earning-part-7-convolutional-neural-network-cnnb003b477dc94)

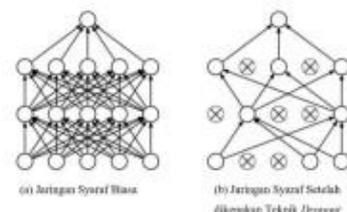
[convolutional-neural-network-cnnb003b477dc94](https://medium.com/@samuelsena/pengenalandeepl earning-part-7-convolutional-neural-network-cnnb003b477dc94))

2.5 Fully Connected Layer

Fully connected layer adalah sebuah lapisan di mana semua neuron aktivasi dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya. Feature map yang dihasilkan dari feature extraction layer masih berbentuk multidimensional array, sehingga perlu dilakukannya *flatten* atau *reshape* feature map menjadi sebuah vector agar bisa digunakan sebagai input dari fully connected layer.

2.6 Dropout Regulation

Dropout merupakan proses mencegah terjadinya *overfitting* dan juga mempercepat proses *learning*. *Dropout* adalah teknik regularisasi jaringan saraf di mana beberapa neuron akan dipilih secara acak dan tidak dipakai selama pelatihan.



Gambar 3. Implementasi Dropout Regulation

(sumber:

<https://medium.com/@amarbudhiraja/httpsmedium-comamarbudhiraja-learning-less-to-learnbetter-dropout-in-deep-machinelearning-74334da4bfc5>)

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Gambar 4. Confusion Matrix

2.7 Softmax Classifier

Fungsi *softmax* digunakan untuk metode klasifikasi dengan jumlah kelas yang banyak. *Output* dari *softmax* ini dapat digunakan untuk mewakili distribusi sebuah kategori. *Softmax* adalah sebuah fungsi yang mengubah Kdimensi vektor 'x' yang berupa nilai sebenarnya menjadi vektor dengan bentuk yang sama namun dengan nilai dalam rentang 0-1, yang jumlahnya 1.

2.8 Adam Optimizer

Adaptive Moment Estimation (Adam) adalah metode lain yang menghitung tingkat pembelajaran adaptif untuk setiap parameter. Adam merupakan algoritma optimasi yang dikembangkan dengan memanfaatkan kelebihan dari algoritma *Adaptive Gradient* (AdaGrad) dan *Root Mean Square Propagation* (RMSProp). Adam mengimplementasikan *moving average* eksponensial dari gradien untuk skala tingkat pembelajaran. Adam menggunakan nilai kedua dari gradien. Hal ini secara khusus dapat menghitung rata-rata hitung eksponensial dari gradien, gradien kuadrat, dan parameter beta 1 dan beta 2 mengontrol rata-rata *decay*.

2.9 Nadam Optimizer

Nesterov-accelerated Adaptive Moment Estimation (Nadam) adalah gabungan dari Adam dan *Nesterov Accelerated Gradient* (NAG). Dengan menggunakan metode momentum, proses pembelajaran dipercepat dengan merangkum rata-rata dari pergerakan eksponensial dari momentum sebelumnya hingga momentum saat ini. Secara intuitif, metode momentum menggabungkan tandatanda kebalikan dari gradien dalam arah kelengkungan tinggi dengan kecepatan yang lebih tinggi untuk meredam fluktuasi.

2.10 Confusion Matrix

Confusion matrix juga sering disebut *error matrix*. Pada dasarnya *confusion matrix* memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya. *Confusion matrix* berbentuk tabel matriks yang menggambarkan kinerja model klasifikasi pada serangkaian data uji yang nilai sebenarnya diketahui. Penentuan baik atau tidaknya performa suatu model klasifikasi dapat dilihat dari parameter pengukuran performanya, yaitu tingkat akurasi, *recall*, dan *precision*.

Terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada *confusion matrix*. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). TP merupakan data positif yang diprediksi benar, TN adalah data negatif yang diprediksi benar, FP merupakan data negatif namun diprediksi sebagai data positif, dan FN adalah data positif namun diprediksi sebagai data negatif. *Accuracy* menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. Maka, *accuracy* merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Nilai *accuracy* dapat diperoleh dengan persamaan di bawah.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (2)$$

Precision menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model. *Precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Nilai *precision* dapat diperoleh dengan persamaan berikut:

$$Precision = \frac{(TP)}{(TP + FP)} \quad (3)$$

Recall menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. Maka, *recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Nilai *recall* dapat diperoleh dengan persamaan berikut:

$$Recall = \frac{(TP)}{(TP + FN)} \quad (4)$$

3. Analisis dan Perancangan Sistem

3.1 Analisis Kebutuhan

Analisis kebutuhan adalah menganalisis kebutuhan user dan sumber daya yang akan digunakan sehingga dapat ditentukan kompatibilitas aplikasi yang dibangun terhadap sumber daya yang ada. Kebutuhan user yaitu diperlukannya koneksi internet dan fasilitas seperti kamera yang terdapat pada handphone. Kamera untuk mengambil gambar burung yang akan dikenali dan koneksi internet untuk mengirim gambar burung ke web service. Penggunaan web service bertujuan untuk meringankan aplikasi jika model CNN yang digunakan terlalu besar ukurannya.

Menurut lampiran peraturan Menteri Lingkungan Hidup dan Kehutanan Republik Indonesia terdapat 532 spesies yang dilindungi. Untuk penelitian ini dipilih sebanyak 15 spesies yaitu burung elang jawa, julang emas, gelatik jawa,

merak hijau, nuri raja ambon, kasturi ternate, kasturi kepala hitam, perkici pelangi, nuri tanau, jalak putih sayap hitam, kacamata jawa, madu sepah raja, kakatua raja, kakatua jambul kuning, dan kakatua putih. Pemilihan ini berdasarkan burung yang sering ditemui karena dari beberapa spesies tersebut sulit untuk ditemukan. Untuk proses pengenalan citra burung dilindungi membutuhkan spesifikasi perangkat keras dan perangkat lunak tertentu. Mengenai spesifikasi perangkat keras dan perangkat lunak pada penelitian ini, ditunjukkan pada Tabel 1.

Tabel 1. Spesifikasi Perangkat Keras dan Lunak

Perangkat Keras	Intel Core i7-9750H @ 2.60GHz
	NVIDIA GeForce 1650 4GB GDDR5
	8.00 GB RAM DDR4
Perangkat Lunak	Windows 10 Home Single Language
	Python 3.7.7
	Sublime Text 3 dan Android Studio

3.2 Pengumpulan Data

Pengumpulan data dilakukan dengan menggunakan salah satu ekstensi dari browser Google Chrome yaitu “Fatkun batch unduh gambar”. Pengumpulan data sebanyak 3.000 citra dari 15 kelas dimana setiap kelas berjumlah 200 citra. Nantinya dataset akan dibagi menjadi dua dataset. Dataset pertama untuk melakukan proses training dimana setiap kelas berjumlah 195 citra dengan total keseluruhan 2.925 citra. Berikutnya dataset kedua untuk melakukan uji coba model dengan citra baru yang diambil secara acak sebanyak 5 citra setiap kelas dengan total keseluruhan dataset uji coba sebanyak 75 citra.

3.3 Preprocessing Data

Preprocessing merupakan tahap awal yang dilakukan sebelum melakukan metode CNN. Secara keseluruhan data memiliki ukuran yang berbeda-beda. Oleh karena itu, *preprocessing* citra dilakukan untuk menyiapkan citra kemudian diproses lebih lanjut. *Preprocessing* yang dilakukan adalah mengubah ukuran (*resize*) citra burung. *Resize* citra dilakukan untuk menyamakan semua data agar satu ukuran. Selanjutnya citra akan menggunakan *augmentation*. *Augmentation* adalah salah satu teknik yang dapat menambah dataset tanpa memerlukan citra baru yaitu dengan cara menggandakan gambar dengan beberapa variasi sehingga model dapat belajar dari lebih banyak contoh. Terdapat lima teknik *augmentation* yang bisa digunakan yaitu: untuk menggeser gambar dapat digunakan argumen *width_shift_range* dan *height_shift_range*, untuk membalikkan gambar bisa menggunakan argumen *horizontal_flip* dan *vertical_flip*, memutar gambar melalui argumen *rotating_range*, untuk mengatur kecerahan gambar menggunakan argumen *brightness_range*, dan pembesaran gambar menggunakan argumen *zoom_range*.

3.4 Pemilihan Arsitektur CNN

Pada penelitian ini, arsitektur CNN yang akan digunakan adalah AlexNet dan LeNet. Untuk membandingkan arsitektur mana yang menghasilkan tingkat akurasi lebih baik. AlexNet memiliki lima *convolution layers* dengan tiga *max pooling layers*. LeNet memiliki dua *convolution layers* diikuti dengan dua *max pooling*.

3.5 Alur dan Perancangan Antarmuka Aplikasi

Alur aplikasi merupakan urutan proses jalannya aplikasi mulai dari pengguna memasukan gambar hingga gambar tersebut dapat dikenali. Pada pertama kali pengguna akan mengambil citra menggunakan kamera yang terdapat pada *smartphone* atau bisa menggunakan citra yang sudah tersimpan di dalam *gallery*. Setelah itu akan dilakukan ke tahap berikutnya yaitu mengatur bagian mana yang akan dikenali dengan cara *crop* selanjutnya citra yang sudah siap untuk dikenali akan dikirimkan ke *web service* untuk masuk ke dalam model CNN untuk melakukan pengenalan. Jika sudah selesai maka akan mengembalikan *response* berupa nama burung dengan tingkat keyakinan paling tinggi.

3.6 Pengujian Program

Untuk mengetahui kecepatan *response* saat mengirim citra lalu mengembalikan sebuah informasi dengan menggunakan *web service* maka dibutuhkan pengujian. Pengujian lain juga dibutuhkan untuk mengetahui apakah fitur-fitur yang terdapat pada aplikasi berjalan semua tanpa ada kendala. Untuk mencari tahu keakuratan aplikasi, salah satu evaluasi yang sering digunakan yaitu mencari nilai akurasi, *recall*, dan presisi. Untuk menghitung faktor-faktor tersebut diperlukan sebuah matriks yang biasa disebut *confusion matrix*.

4. Hasil dan Pembahasan

4.1 Pengumpulan Data

Hasil pengumpulan gambar dengan total berjumlah 3000 data citra jenis burung yang sudah ditetapkan dengan jumlah masing-masing jenis yaitu 200 citra burung. Diambil secara acak berjumlah 5 citra dari setiap kelas untuk dipisahkan sebagai dataset uji coba data baru. Dengan jumlah akhir sebanyak 195 citra burung setiap kelas yang akan digunakan sebagai dataset *training*.

4.2 Hasil Pelatihan Arsitektur

Dalam algoritma CNN pembentukan arsitektur jaringan dan parameter yang digunakan dapat mempengaruhi hasil dari akurasi model. Arsitektur yang digunakan yaitu AlexNet dan LeNet. Setelah dataset terkumpul dengan *folder* sesuai kelas, selanjutnya dataset diberi label sesuai kelas atau kategori. Pelabelan dimulai dari label 0 hingga 14. Sehingga menghasilkan label dan kelas sebagai berikut:

Tabel 2 Pembagian Kelas Berdasarkan Label

Kelas	Label
0	Elang Jawa
1	Gelatik Jawa
2	Jalak Putih Sayap Hitam
3	Julang Emas
4	Kacamata Jawa
5	Kakatua Jambul Kuning
6	Kakatua Jambul Putih
7	Kakatua Raja
8	Kasturi Kepala Hitam
9	Kasturi Ternate
10	Madu Sepah Raja
11	Merak Hijau
12	Nuri Raja Ambon
13	Nuri Tanau
14	Perkici Pelangi

Dataset yang digunakan pada penelitian ini berjumlah 2.925 citra burung dilindungi dimana 80% dari data tersebut digunakan untuk proses *training* dan sisanya 20% digunakan untuk *validation*. Selanjutnya adalah penentuan dari beberapa parameter yang dibutuhkan oleh model CNN. Untuk model dengan menggunakan arsitektur AlexNet dengan *input* citra yang berukuran 227x227x3. Sedangkan untuk *input* gambar pada model dengan menggunakan arsitektur LeNet adalah 64x64x3 dan 32x32x3. Angka tiga yang dimaksud adalah sebuah citra yang memiliki 3 channel yaitu *Red*, *Green*, dan *Blue* (RGB). Untuk *batch size* berukuran 32. *Batch size* adalah jumlah sampel yang disebarkan ke dalam arsitektur *neural network*. Kemudian *learning rate* nya membandingkan nilai *learning rate* 0.001 dan 0.0001. Arsitektur AlexNet menggunakan *dropout regulation* dengan nilai 0.4 dan arsitektur LeNet menggunakan *dropout regulation* dengan nilai 0.6. Kedua arsitektur menggunakan optimasi Adam dan Nadam. Proses *augmentation* menggunakan

ImageDataGenerator dari *library Keras*. Proses ini biasa disebut pembangkitan data. Untuk *rotation* ini digunakan untuk merotasi gambar searah jarum jam dengan nilai derajat 0 hingga 30. *Width shift* dan *height shift* digunakan untuk memindahkan semua piksel gambar dalam satu arah, seperti horizontal atau vertikal dengan menjaga dimensi gambar tetap sama. *Shear* dan *zoom* digunakan untuk merotasi ke arah berlawanan dengan arah jarum jam dan juga memperbesar gambar. Berikut hasil perbandingan arsitektur LeNet antara menggunakan *augmentation* dan tidak menggunakan *augmentation* dengan parameter yang sama yaitu: optimasi Nadam, *epoch* 500, *learning rate* 0.001, *batch size* 32, dan *input size* 64x64 *pixel*.

Tabel 3. Perbandingan Arsitektur LeNet Dengan Optimasi Nadam Menggunakan Augmentation dan Tidak Menggunakan Augmentation

<i>Epoch</i>	<i>Accuracy Validation</i>	<i>Loss Validation</i>	<i>Time (Minutes)</i>	<i>Image Size (Pixel)</i>	<i>Learning Rate</i>	<i>Augmentation</i>
500	92.65%	0.24	17	64x64	0.001	Ya
500	78.12%	1.19	15	64x64	0.001	Tidak

Berdasarkan Tabel 3 dapat dilihat bahwa dengan menggunakan teknik *augmentation* dapat menambah akurasi *validation*. Karena dengan menggunakan *augmentation* dapat menambah dataset dengan berbagai variasi. Selanjutnya untuk mencari akurasi *training*, akurasi *validation*, dan *loss validation* dari sebuah model dapat menggunakan *model.evaluate* dari library *TensorFlow*. Untuk menyimpan model yang sudah melakukan proses *training* menggunakan *model.save* yang nantinya tersimpan dalam sebuah *directory* dan dapat digunakan untuk mengklasifikasikan data baru. Untuk memanggil grafik proses *training* dengan menggunakan library *Matplotlib*. *Matplotlib* adalah sebuah *visualize tools*, dengan bantuan *tools* ini dapat mempermudah dalam

melihat tingkat *accuracy* dan *loss* model dari data *train* dan *validation*.

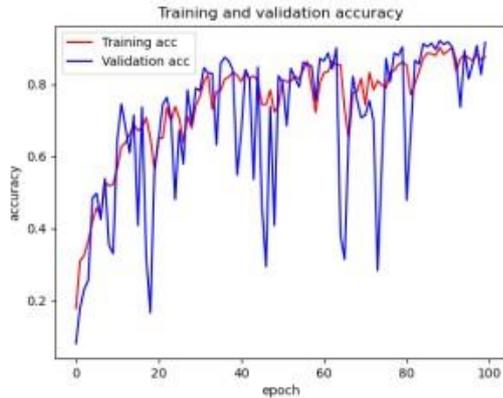
4.3 Arsitektur AlexNet

Untuk arsitektur AlexNet ada beberapa parameter yang digunakan untuk melakukan proses *training* untuk menghasilkan model yang optimal. Parameter yang dimaksud adalah penggunaan optimasi, jumlah *epoch* dan *learning rate*. *Epoch* yang digunakan mulai dari 100, 250, dan 500 sedangkan untuk *learning rate* bernilai 0.001 dan 0.0001. Dengan *input size* sebesar 227x227 *pixel* dan *batch size* 32. Berikut adalah hasil perbandingan jumlah *epoch* dan *learning rate* dari hasil *training*.

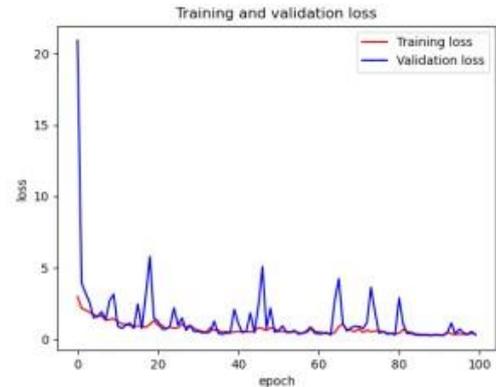
Tabel 4. Hasil Perbandingan Menggunakan Arsitektur AlexNet Optimasi Adam, Epoch dan Learning Rate

<i>Epoch</i>	<i>Accuracy Training</i>	<i>Accuracy Validation</i>	<i>Loss Validation</i>	<i>Time (Minutes)</i>	<i>Learning Rate</i>	<i>Optimasi</i>
100	97.14%	91.62%	0.30	33	0.001	Adam
250	93.12%	84.96%	6.53	72	0.001	
500	95.90%	88.72%	0.40	144	0.001	
100	74.19%	69.40%	1.05	32	0.0001	
250	85.43%	78.97%	0.75	72	0.0001	
500	98.29%	88.89%	0.45	148	0.0001	
100	84.62%	74.70%	0.80	31	0.001	Nadam
250	91.03%	81.71%	0.63	75	0.001	
500	96.75%	88.72%	2.4	148	0.001	
100	95.17%	89.40%	0.38	30	0.0001	
250	99.40%	90.43%	0.37	78	0.0001	
500	97.99%	90.60%	0.38	148	0.0001	

Berdasarkan Tabel 4 di atas model dengan menggunakan optimasi Adam mendapatkan akurasi terbaik dengan *epoch* berjumlah 100, dan menggunakan *learning rate* bernilai 0.001 dengan hasil akurasi sebesar 97.14% untuk training dan 91.62% untuk akurasi validation dalam waktu 33 menit. Hasil grafik accuracy dan loss sebagai berikut:



Gambar 5. Grafik Akurasi Training dan Validation Menggunakan Arsitektur AlexNet Epoch 100 dan Learning Rate 0.001



Gambar 6. Grafik Loss Training dan Validation Menggunakan Arsitektur AlexNet

Tabel 5. Precision dan Recall Hasil Dari Arsitektur AlexNet Optimasi Adam Epoch 100 dan Learning Rate 0.001

Kelas	Precision	Recall	F1-Score	Support
0	0.94	0.85	0.89	39
1	0.97	0.92	0.95	39
2	0.89	0.82	0.85	39
3	0.97	0.82	0.97	39
4	0.93	0.97	0.95	39
5	0.78	0.79	0.78	39
6	0.83	0.90	0.86	39
7	0.88	0.90	0.89	39
8	0.97	0.92	0.95	39
9	0.92	0.92	0.92	39
10	0.95	0.97	0.96	39
11	0.97	0.97	0.97	39
12	0.81	0.90	0.85	39
13	0.95	0.92	0.94	39
14	1.00	1.00	1.00	39

Epoch 100 dan Learning Rate 0.001

Hasil klasifikasi yang didapatkan dari data *validation* ditunjukkan melalui *precision* dan *recall* seperti berikut:

0.79 yang berarti jumlah benar dikenali dari kelas tersebut hanya sebesar 79%.

4.4 Arsitektur LeNet

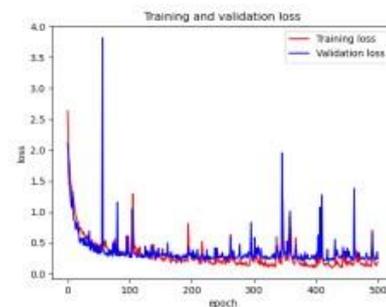
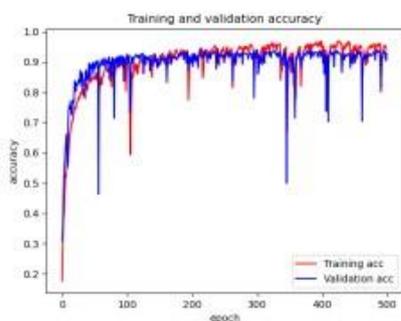
Untuk arsitektur LeNet ada beberapa parameter yang digunakan untuk melakukan proses *training* untuk menghasilkan model yang optimal. Parameter yang dimaksud adalah penggunaan optimasi, jumlah *epoch*, *input size*, dan *learning rate*. *Epoch* yang digunakan mulai dari 100, 250, dan 500 sedangkan untuk *learning rate* bernilai 0.001 dan 0.0001 untuk batch size yaitu 32. Dengan *input size* sebesar 32x32 pixel dan 64x64 pixel. Berikut adalah hasil perbandingan jumlah *epoch* dan *learning rate* dari hasil *training*.

Dari Tabel 5 *precision* dan *recall* di atas menunjukkan bahwa kelas 5 memiliki data yang menyebar bisa dilihat dengan nilai *precision* sebesar 0.78 dan nilai *recall* sebesar

**Tabel 6. Hasil Perbandingan Menggunakan Arsitektur LeNet Optimasi,
*Epoch, Input Size dan Learning Rate***

Epoch	Accuracy Training	Accuracy Validation	Loss Validation	Time (Minutes)	Image Size (Pixel)	Learning Rate	Optimasi
100	97.35%	87.35%	0.42	1.8	32x32	0.001	Adam
100	93.93%	83.76%	0.48	1.8	32x32	0.0001	
100	97.91%	90.60%	0.33	3.5	64x64	0.001	
100	98.63%	89.57%	0.33	3.4	64x64	0.0001	
250	99.23%	91.62%	0.34	4.2	32x32	0.001	
250	98.80%	89.57%	0.40	4.1	32x32	0.0001	
250	98.72%	91.11%	0.31	8.5	64x64	0.001	
250	99.70%	90.26%	0.36	8.8	64x64	0.0001	
500	99.74%	90.60%	0.40	8.1	32x32	0.001	
500	99.91%	90.60%	0.41	8	32x32	0.0001	
500	99.87%	92.13%	0.31	16.7	64x64	0.001	
500	99.70%	92.14%	0.38	16.8	64x64	0.0001	
100	98.08%	89.40%	0.39	1.9	32x32	0.001	Nadam
100	93.80%	84.27%	0.50	1.8	32x32	0.0001	
100	98.25%	88.55%	0.37	3.8	64x64	0.001	
100	98.29%	90.26%	0.35	3.6	64x64	0.0001	
250	99.74%	90.94%	0.38	4	32x32	0.001	
250	98.85%	87.86%	0.41	4	32x32	0.0001	
250	99.40%	92.65%	0.28	8.3	64x64	0.001	
250	99.87%	91.97%	0.31	8.5	64x64	0.0001	
500	99.49%	90.60%	0.39	7.8	32x32	0.001	
500	99.79%	88.89%	0.51	7.8	32x32	0.0001	
500	99.83%	92.65%	0.24	17	64x64	0.001	
500	99.91%	92.42%	0.32	17	64x64	0.0001	

Berdasarkan Tabel 6 di atas model dengan menggunakan arsitektur LeNet optimasi Nadam mendapatkan akurasi terbaik dengan *epoch* berjumlah 500, dan menggunakan *learning rate* bernilai 0.001 dengan hasil akurasi sebesar 99.83% untuk *training* dan 92.65% untuk akurasi *validation* dalam waktu 17 menit. Hasil grafik *accuracy* dan *loss* sebagai berikut:



Gambar 8. Grafik Loss Training dan Validation

Menggunakan Arsitektur LeNet Optimasi Nadam Epoch 500, Input Size 64x64pixel, dan Learning Rate 0.001

Gambar 7. Grafik Akurasi Training dan Validation Menggunakan Arsitektur LeNet Optimasi Nadam Epoch 500, Input Size 64x64pixel, dan Learning Rate 0.001

Tabel 7. Precision dan Recall Hasil Dari Arsitektur LeNet Optimasi Adam Epoch 100 dan Learning Rate 0.001

Kelas	Precision	Recall	F1-Score	Support
0	0.86	0.92	0.89	39
1	0.97	0.90	0.93	39
2	0.78	0.79	0.78	39
3	0.95	0.97	0.96	39
4	0.97	0.85	0.90	39
5	0.81	0.87	0.84	39
6	0.90	0.90	0.90	39
7	0.95	0.92	0.94	39
8	0.92	0.90	0.91	39
9	0.97	0.97	0.97	39
10	1.00	0.97	0.99	39
11	0.95	1.00	0.97	39
12	0.97	0.97	0.97	39
13	0.97	0.97	0.97	39
14	0.95	0.97	0.96	39

Dari Tabel 7 *precision* dan *recall* di atas menunjukkan bahwa kelas 2 memiliki penyebaran data yang banyak bisa dilihat dengan nilai *precision* sebesar 0.78 dan nilai *recall* sebesar 0.79 yang berarti jumlah benar dari kelas tersebut sebesar 79%.

4.5 Perbandingan Keseluruhan Arsitektur

Berdasarkan beberapa percobaan pencarian parameter terbaik dalam mencari akurasi terbaik dengan arsitektur hasil perbandingan akurasi secara keseluruhan.

yang berbeda Maka hasil yang didapat menggunakan arsitektur LeNet dengan optimasi Nadam dan parameter *epoch* 500, *image size* berukuran 64x64 *pixel*, dan *learning rate* 0.001 mendapatkan akurasi terbaik yaitu 99.83% untuk akurasi *training* dan 92.65% untuk akurasi *validation*. Jika dibandingkan dengan arsitektur AlexNet dengan optimasi Nadam parameter yang sama yaitu *epoch* 500 dan *learning rate* 0.001 mendapatkan akurasi 96.75% untuk *training* dan 88.72% untuk *validation*. Berikut adalah

Tabel 8 Perbandingan Untuk Keseluruhan Arsitektur

AlexNet		
Optimasi	Hasil Akurasi (Data Training)	
	Tertinggi	Terendah
Adam	91.62%	69.40%
Nadam	90.60%	74.70%
LeNet		
Optimasi	Hasil Akurasi (Data Training)	
	Tertinggi	Terendah
Adam	92.14%	83.76%
Nadam	92.65%	84.27%

4.6 Pengujian Aplikasi

Pengujian aplikasi untuk mencari kecepatan terhadap *response* yang diberikan oleh *web service* untuk mengembalikan sebuah informasi pengenalan burung dilindungi. Saat pengujian aplikasi provider internet yang digunakan yaitu *IndiHome* dengan kecepatan unduh 20 Mbps. Pengujian dilakukan dengan menggunakan citra berukuran 1.3 MB dan 1.66 MB. Untuk waktu yang didapatkan saat menggunakan citra berukuran 1.3 MB adalah 1.32 detik sedangkan untuk menggunakan citra berukuran 1.66 MB memakan waktu 2.15 detik. Semakin kecil ukuran citra maka semakin cepat *web service* mengembalikan informasi. Untuk fitur-fitur yang terdapat pada aplikasi setelah dicoba menggunakan *handphone* lain tidak ada kendala semua berjalan dengan baik.

4.7 Pengujian Model

CNN membutuhkan proses *training* dan *testing*. Proses *training* ini bertujuan untuk melatih algoritma CNN dalam mengenali datasetnya dan membentuk sebuah model dengan parameter yang paling optimal. Untuk memanggil model yang sudah tersimpan di *directory* menggunakan *load_model*. Langkah pertama adalah membaca gambar yang akan di kenali tentu *input* gambar harus sesuai dengan *input* pada proses *train* maka perlu di *resize* dengan ukuran yang sama. Selanjutnya memanggil model yang sudah dibuat saat *training* data. Terdapat 15 label yang nantinya sebagai hasil dari pengenalan citra untuk kode sumber dapat dilihat pada Lampiran A. Dengan menggunakan data baru sebagai uji model dengan total 75 gambar burung

dengan 5 gambar setiap kelas. Hasil akhir klasifikasi yang didapatkan dari data baru ditunjukkan melalui tabel *confusion matrix* sebagai berikut:

sebesar 99.83% untuk *training* dan 92.65% untuk *validation*.

Tabel 9. Confusion Matrix Data Baru

		Prediksi														
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Actual	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	4	1	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	4	0	0	0	0	0	1	0	0	0	0	0	0
	3	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	4	1	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	4	1	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
	11	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0
	12	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
	13	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5

Berdasarkan Tabel 4.14 di atas hasil pengenalan dari model terhadap data baru menunjukkan hasil yang baik. Pengenalan terhadap kelas 0 di klasifikasikan ke dalam kelas 0 artinya *system* mengenali gambar tersebut dengan benar. Pengenalan pada kelas 1 yang tepat di kelas 1 sebanyak 4 data dan 1 data dikenali sebagai kelas 2. Terdapat juga kelas lainnya yang memiliki kesalahan dalam mengenali suatu gambar yaitu kelas 2 dengan 1 gambar dikenali sebagai kelas 7, kelas 5 dengan 1 gambar dikenali sebagai kelas 6, dan kelas 8 dengan 1 gambar masuk ke dalam kelas 9. Perhitungan akurasi dari keseluruhan *confusion matrix* dapat dicari dengan cara menjumlah klasifikasi dengan benar dibagi dengan jumlah keseluruhan data uji. Maka mendapatkan *overall accuracy* sebagai berikut:

$$\text{Overall accuracy} = \frac{71}{75} = 93\%$$

Jadi akurasi yang dihasilkan oleh model dengan menggunakan arsitektur LeNet dengan menggunakan optimasi Nadam, *input* gambar 64x64 *pixel*, nilai *learning rate* 0.001, dan jumlah *epoch* 500 didapatkan nilai akurasi sebesar 93%.

5. Penutup

5.1 Simpulan

Berdasarkan hasil penerapan metode *Convolutional Neural Network* (CNN) untuk pengenalan citra burung dilindungi, diperoleh beberapa kesimpulan sebagai berikut:

1. Model CNN dengan arsitektur LeNet pada penelitian ini menggunakan *input* gambar berupa 64x64, nilai *learning rate* 0.001, jumlah *epoch* 500, fungsi optimasi Nadam, jumlah data 2.925 dengan pembagian 80% untuk *training* dan 20% untuk *validation*. Menghasilkan tingkat akurasi *training* dan *validation* dalam melakukan klasifikasi gambar burung dilindungi

2. Dengan menggunakan data *testing* baru sebanyak 75 untuk diujikan ke dalam model yang sudah dibuat. Hasil *testing* tersebut mendapatkan akurasi sebesar 95% dalam melakukan klasifikasi burung dilindungi. Dengan begitu bisa memudahkan masyarakat atau pecinta burung untuk mengetahui burung apa saja yang dilindungi oleh negara.

5.2 Saran

Adapun saran yang diberikan pada penelitian ini sebagai berikut:

1. Penelitian selanjutnya diharapkan dapat menambah jenis burung yang dilindungi di Indonesia untuk pengenalannya.
2. Penelitian selanjutnya juga bisa mencoba dengan arsitektur lain tidak hanya menggunakan LeNet dan AlexNet.

6. Daftar Pustaka

Edwin, Y. (2019). *Spesies burung di Indonesia semakin kaya*. Retrieved from <https://lokadata.id/artikel/spesiesburung-di-indonesiasemakin-kaya>

Nurfita, R. D. (2018). Implementasi Deep Learning Berbasis Tensorflow Untuk Pengenalan Sidik Jari [UNIVERSITAS MUHAMMADIYAH SURAKARTA]. In Emitter: Jurnal Teknik Elektro (Vol. 18, Issue 01). doi: 10.23917/emitor.v18i01.6236

Nurhikmat, T. (2018). *Implementasi Deep Learning untuk Image Classification Menggunakan Algoritma Convolutional Neural Network Pada Citra Wayang Golek* (Vol. 2). UNIVERSITAS ISLAM INDONESIA.

Putra, W. S. E., Wijaya, A. Y., & Soelaiman, R. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101. *Jurnal Teknik ITS*, 5(1). doi: 10.12962/j23373539.v5i1.15696

Sofia, N. (2018). *Convolutional Neural Network*.
Retrieved from <https://medium.com/@nadhifasofia/1-convolutional-neural-networkconvolutional-neuralnetworkmerupakan-salah-satu-metodemachine28189e17335b>