

SISTEM DETEKSI HEWAN MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORKS (CNN)

Alifia Mutiara Rahma
Prodi Teknik Komputer
Telkom University
Bandung, Indonesia

alifiamutiaraahma@student.telkomuniversity.ac.id

Galih Karya Gemilang
Prodi Teknik Komputer
Telkom University
Bandung, Indonesia

galihkaryagemilang@student.telkomuniversity.ac.id

Abstract— Klasifikasi hewan menggunakan algoritma *Convolutional Neural Networks (CNN)* telah menjadi topik yang menarik dalam bidang visi komputer dan pemrosesan citra. Dalam penelitian ini, kami mengusulkan dan mengimplementasikan sebuah sistem deteksi hewan berbasis *CNN* dengan tujuan untuk secara otomatis mengklasifikasikan spesies hewan berdasarkan gambar digital. Melalui pelatihan dan evaluasi model *CNN* pada dataset yang luas dan beragam, kami mencapai tingkat akurasi yang tinggi dalam mengenali dan membedakan spesies hewan yang sulit secara visual. Penelitian ini menunjukkan potensi besar algoritma *CNN* dalam aplikasi deteksi dan klasifikasi hewan secara otomatis, seperti pemantauan satwa liar, konservasi alam, atau penelitian ekologi. Tantangan yang perlu diatasi di masa depan termasuk keterbatasan data dan peningkatan efisiensi komputasi untuk pengembangan yang lebih lanjut. Hasil penelitian ini memberikan kontribusi yang berarti dalam pengembangan sistem deteksi hewan yang efektif dan otomatis menggunakan algoritma *CNN*.

Keywords— *Convolutional Neural Networks (CNN)*, Deteksi hewan, klasifikasi hewan, pemrosesan citra.

I. PENDAHULUAN

Dewasa ini, penggunaan teknologi untuk mendeteksi dan mengidentifikasi objek atau sebuah entitas tertentu menjadi semakin penting karena banyak dibutuhkan. Lebih spesifiknya dalam pendeteksian hewan. Hal ini menjadi bahasan yang menarik karena dapat digunakan dalam penitipan hewan, konservasi satwa liar, pengawasan lingkungan, bahkan penelitian biologi.

Sistem deteksi hewan dengan menggunakan algoritma *Convolutional Neural Network* atau *CNN* memanfaatkan jaringan saraf tiruan (*artificial neural networks*) yang dirancang khusus untuk memproses dataset gambar. *Convolutional Neural Network (CNN)* adalah pengembangan dari *Multiplayer Perceptron (MLP)* yang didesain untuk mengolah data dua dimensi [5]. Algoritma ini terinspirasi dari struktur visual korteks otak manusia dan bisa mengenali pola serta fitur visual dalam gambar dengan tingkat akurasi yang tinggi.

Menurut Suartika dkk. (2016, 2) sebuah *CNN* terdiri dari beberapa *layer*. Berdasarkan arsitektur LeNet5 [5], terdapat beberapa macam *layer* utama pada sebuah *CNN* antara lain *Convolution Layer* (lapisan utama dalam *CNN*), *Pooling*

Layer (lapisan pengambilan sampel), *Batch Normalization Layer* (lapisan normalisasi batch), *Fully Connected Layer* (lapisan klasifikasi). Setiap lapisan dalam *CNN* mempunyai peran yang penting dalam pemrosesan dan ekstraksi fitur dari gambar yang diinputkan. Selain itu, dalam sistem deteksi ini menggunakan fungsi aktivasi yakni menggunakan *Rectified Linear Unit (ReLU)*. Lapisan *ReLU* mengikuti lapisan konvolusi untuk menerapkan fungsi aktivasi *ReLU* pada setiap elemen.

Dataset yang digunakan dalam sistem deteksi hewan yaitu menggunakan dataset CIFAR-10 dari tensorflow. Dataset ini terdiri dari 60000 citra berwarna 32x32 dalam 10 kelas, dengan 6000 citra per kelasnya. Dalam dataset CIFAR-10 sudah diberi label pada masing-masing kelasnya. Seperti Namanya CIFAR-10 ini mempunyai 10 kelas dengan label nama pesawat, mobil (bukan truk atau *pick up*), burung, kucing, rusa, anjing, katak, kuda, kapal, dan truk. Dengan banyaknya variasi kelas hewan yang beragam, dataset ini memungkinkan pengembangan dan evaluasi model yang bisa mengenali sekaligus mengklasifikasikan objek hewan dengan akurasi yang tinggi.

Penggunaan sistem deteksi hewan dapat menghasilkan manfaat yang luas. Pemanfaatan sistem ini dapat diterapkan dalam penitipan hewan untuk membantu staf dalam mengidentifikasi atau mengklasifikasikan hewan-hewan yang dititipkan dengan cepat dan akurat. Selain itu, sistem ini juga bisa bermanfaat di bidang konservasi satwa liar apabila programnya dibuat lebih detail. Hal ini dapat membantu dalam survei populasi, melacak sekaligus mengidentifikasi spesies hewan yang terancam punah, serta mendukung Tindakan perlindungan. Bahkan sistem ini dapat dikembangkan juga untuk mendeteksi kehadiran hewan liar yang bisa berpengaruh terhadap ekosistem. Namun, dalam program yang dibuat baru sampai mengidentifikasi jenis hewan yang diinputkan kemudian akan dibaca dalam dataset, gambar tersebut termasuk dalam kelas tertentu.

II. PENELITIAN YANG TERKAIT

A. *Model prediksi jenis hewan dengan metode Convolution Neural Network (Harry Dhika, Nia Rahma Kurnianda, Puput Irfansyah, Wisnu Ananta, 2020)*

Pada era teknologi maju saat ini, teknologi pengenalan citra atau gambar menjadi salah satu inputan yang banyak diterapkan pada berbagai bidang karena manfaatnya begitu luas. Salah satu potret fitur pengenalan citra yaitu digunakan untuk mengenali dan memprediksi jenis objek dalam citra. Salah satu metode yang digunakan adalah *CNN*. *CNN*

merupakan metode efektif dalam pengenalan citra dan klasifikasi. Pada penelitian ini juga menggunakan *CNN* untuk melatih komputer mengenali dan memprediksi jenis binatang berdasarkan gambar yang diinputkan. Hasil penelitian menunjukkan tingkat akurasi *training* sebesar 97,56%. Data sampel yang digunakan adalah gambar kucing dan anjing dengan dimensi 64x64 RGB [3].

B. Klasifikasi ras pada kucing menggunakan algoritma Convolutional Neural Network (CNN) (Muhammad Afif Amanullah Fawwaz, Kurniawan Nur Ramadhani, S.T., Febryanti Sthevanie, S.T., M.T., 2020)

Kucing adalah hewan peliharaan yang bisa dimiliki semua orang. Namun, terkadang tidak semua pemilik kucing mengetahui ras kucingnya. Padahal perawatan kucing sebaiknya dilakukan dengan perawatan khusus untuk menjamin kesehatan kucing disesuaikan dengan ras yang dimiliki kucing tersebut. Untuk itu dalam penelitian ini akan mengklasifikasikan ras kucing menggunakan algoritma *Convolutional Neural Network (CNN)*. Program ini dapat mengidentifikasi ras kucing berdasarkan gambar. Penelitian ini membandingkan empat model *CNN*, yaitu VGG16, InceptionV3, ResNet50, dan Xception, untuk mencari model terbaik. *Preprocessing* dilakukan sebelum melatih dan menguji model. *CNN* bekerja dengan menggunakan operasi konvolusi untuk mengekstrak fitur dari gambar. Model VGG16, InceptionV3, ResNet50, dan Xception dipilih karena keakuratannya dalam klasifikasi. Penelitian sebelumnya juga telah menggunakan model-model tersebut dengan hasil yang baik. Setiap model memiliki arsitektur yang berbeda, seperti jumlah *convolutional layer*, *fully connected layer*, dan teknik seperti *dropout* dan *residual block*. Hasil penelitian ini dievaluasi menggunakan metrik akurasi [4].

C. Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101 (I Wayan Suartika E.P., Arya Yudhi Wijaya, Rully Soelaiman, 2016)

Deep Learning adalah bidang *Machine Learning* yang berkembang pesat berkat teknologi *GPU acceleration*. Salah satu aplikasinya adalah klasifikasi objek pada citra menggunakan *Convolutional Neural Network (CNN)*. Metode *CNN* terdiri dari tahap klasifikasi citra menggunakan *feedforward* dan tahap pembelajaran dengan *backpropagation*. Sebelum klasifikasi, dilakukan praproses dengan *wrapping* dan *cropping* untuk fokus pada objek. Penggunaan *GPU acceleration* menjadi penting dalam pengembangan *Deep Learning* karena mempercepat proses pelatihan model. *Training* dilakukan dengan *feedforward* dan *backpropagation*, dan terakhir klasifikasi menggunakan *feedforward* dengan bobot dan bias yang diperbarui. Uji coba dilakukan pada dataset *Caltech 101* dengan variasi tingkat *confusion*, dan hasilnya menunjukkan rata-rata nilai akurasi yang tinggi. Dengan demikian, dapat disimpulkan bahwa metode *CNN* yang digunakan dalam penelitian ini mampu melakukan klasifikasi objek pada citra dengan baik [5].

D. Klasifikasi Unggas Ordo Anseriformes Berdasarkan Citra Menggunakan Metode Deep Learning Dengan Algoritma Convolutional Neural Network (CNN) (Jean Baptista Robert Openg, Marselinah Endah H., Hamzah, 2022)

Unggas ordo *Anseriformes* merupakan jenis burung (*aves*) air (akuatik) yang jenis spesiesnya banyak tersebar di Indonesia, namun masih banyak masyarakat khususnya para peternak unggas yang belum mengetahui perbedaan dari masing – masing spesies yang tergabung dalam ordo *Anseriformes*. Berdasarkan masalah yang dibahas diatas maka dibuat sebuah aplikasi Klasifikasi Unggas Ordo *Anseriformes* berdasarkan Citra menggunakan Algoritma *Convolution Neural Network* untuk mengklasifikasikan 3 spesies unggas yang hampir serupa tapi berbeda yaitu angsa (*swan*), angsa leher pendek (*goose*), itik Melewar (*mallard duck*) dengan melakukan proses pada model *Training* dan *Testing* lalu diimplementasikan menjadi aplikasi dengan menggunakan *Android Studio*. Hasil dari penelitian ini yaitu aplikasi klasifikasi unggas ordo *Anseriformes* menggunakan algoritma *Convolution Neural Network*. Aplikasi memperoleh hasil akurasi pada kisaran angka 85% - 90% dari tiga kategori spesies unggas ordo *Anseriformes* [6].

E. PENDETEKSIAN OBJEK PADA CITRA HEWAN KARNIVORA DAN HERBIVORA MENGGUNAKAN FASTER R-CNN (Sherien Trisnawaty Eka Putri, Achmad Fahrurrozi, 2022)

Pengelompokkan hewan berdasarkan jenis makanannya terbagi menjadi 3 kelompok besar, yaitu kelompok hewan herbivora, karnivora, dan omnivora. Hewan yang masuk ke dalam kelompok hewan herbivora yaitu hewan-hewan pemakan tumbuhan. Kelompok hewan karnivora yaitu hewan pemakan daging atau pemakan hewan lain seperti Harimau, Macan Tutul, Singa, dan Buaya. Sementara kelompok hewan omnivora yaitu hewan pemakan tumbuhan dan pemakan hewan lain. Pendeteksi objek (*object detection*) adalah suatu teknologi untuk mengenali suatu *object* dari suatu gambar maupun video. Pendeteksian objek merupakan teknologi komputer yang berkaitan dengan *computer vision* dan *image processing* yang berhubungan dengan mendeteksi suatu objek dalam citra digital berdasarkan warna maupun bentuk objek. Algoritma *object detector* modern yang banyak digunakan dalam pengenalan citra adalah *Convolutional Neural Network (CNN)* dan pengembangannya. Penelitian ini menggunakan sistem pendeteksian objek berbasis citra dengan menggunakan *Faster R-CNN* untuk mengklasifikasikan jenis hewan karnivora dan herbivora. Pembuatan program menggunakan arsitektur Inception V2 dan Google Collab pada tahap pelatihan. Data yang digunakan adalah 2000 citra hewan yang dilabelkan secara manual. Sistem pendeteksian objek ini melalui tahap pelatihan, uji coba, dan pengukuran performa. Hasilnya menunjukkan bahwa sistem ini dapat mengklasifikasikan jenis hewan karnivora dan herbivora dengan total *loss model* 0.06 dan tingkat akurasi rata-rata 89%. Sistem ini juga memiliki performa yang baik dengan nilai *Recall* 100% pada beberapa kelompok hewan seperti *Cheetah*, *Eagle*, Komodo, *Shark*, *Tiger*, *Bull*, *Guineapig*, dan Zebra [7].

F. Klasifikasi Ras Anjing Berdasarkan Citra Menggunakan Convolutional Neural Network (Axel Leovinent, Yoannita, 2023)

Anjing mempunyai banyak ras dan mempunyai karakteristik yang berbeda-beda. Karena begitu banyaknya ras anjing terkadang sulit bagi orang untuk mengenali dan memutuskan untuk memelihara ras yang diinginkan. Hal ini berdampak pada kurangnya pengetahuan pemilik anjing tentang cara merawat anjingnya sendiri sebab tidak mengetahui ras anjing yang dipelihara secara spesifik. Maka dari itu pada penelitian ini membahas klasifikasi ras anjing menggunakan Convolutional Neural Network (CNN) dengan model arsitektur *ResNet-50* dan *optimizer Adam*. Dataset yang digunakan terdiri dari 20580 citra anjing yang terbagi menjadi data latih, data validasi, dan data uji dengan perbandingan 60:20:20. Citra-citra tersebut diresize menjadi ukuran 224x224 piksel. Penelitian ini mencapai akurasi sebesar 99,35%. Studi literatur dilakukan untuk memperkuat landasan penelitian, dan data uji dan data latih dikumpulkan dari dataset ras anjing yang terdiri dari 120 ras dengan total 20580 citra. Penelitian ini memiliki tujuan untuk menerapkan metode Convolutional Neural Network dengan arsitektur *ResNet-50* untuk mengklasifikasikan ras anjing [8].

III. METODE PENELITIAN

A. Deep Learning

Machine learning tradisional memiliki batasan-batasan tertentu. Salah satunya adalah ketika kelas yang ingin dipelajari terlalu rumit sehingga teknik sederhana tidak efektif. Selain itu, terlalu banyak detail atribut dapat menyembunyikan informasi penting tentang data. Untuk mengatasi hal ini, diperlukan teknik yang lebih canggih, dan itulah mengapa *deep learning* diciptakan.

Deep learning bergantung pada jaringan saraf yang sekarang diimplementasikan dengan cara yang membuat *MLP (Multi-Layer Perceptron)* yang lama terlihat usang. Berbagai fungsi aktivasi dan cara yang berbeda dalam mengukur kesalahan digunakan. Jumlah lapisan tersembunyi telah berkembang secara signifikan. Dan operator transformasi data yang canggih, yang dipinjam dari bidang penglihatan komputer, umum digunakan.

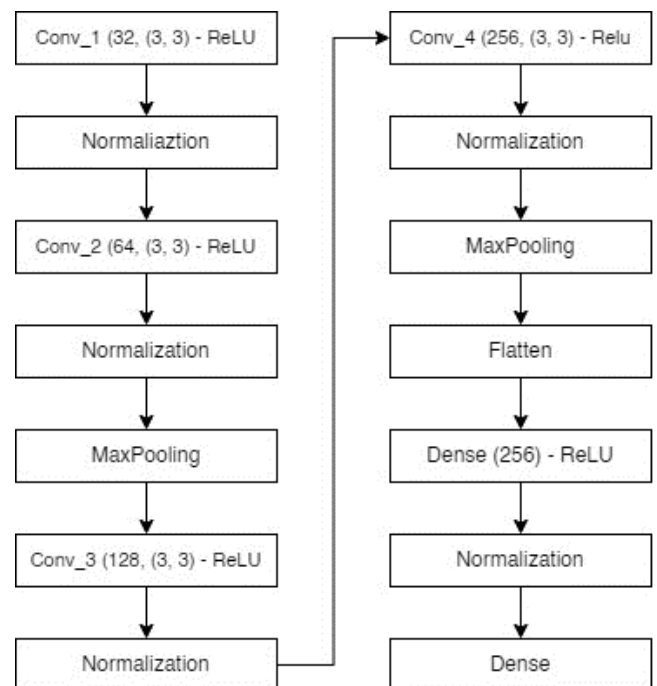
B. Classification

Pengklasifikasi adalah alat yang sangat berharga untuk banyak tugas saat ini, seperti prediksi medis atau genomik, deteksi spam, pengenalan wajah, dan keuangan. Banyak dari aplikasi ini menangani data sensitif [WGH12, SG11, SG13], sehingga penting agar data dan pengklasifikasi tetap bersifat pribadi. [Machine Learning Classification over Encrypted Data]

C. Convolutional Neural Network

Jaringan saraf konvolusional (Convolutional Neural Networks/CNN) adalah tipe struktur jaringan saraf yang sering digunakan untuk pemrosesan data gambar, terutama dalam tugas-tugas seperti mengenali angka tulisan tangan. CNN dirancang untuk memasukkan sifat invarian ke dalam struktur jaringan. Mereka memanfaatkan fakta bahwa piksel-

piksel yang berdekatan dalam sebuah gambar memiliki korelasi yang kuat. Dengan menggunakan lapangan reseptif lokal, pembagian bobot, dan *subsampling*, CNN dapat mengekstraksi fitur-fitur lokal yang penting dari gambar dan mempertahankan invariansi terhadap transformasi tertentu seperti translasi dan distorsi. Dalam arsitektur CNN, terdapat beberapa lapisan konvolusi dan subsampling yang bertujuan untuk meningkatkan invariansi terhadap transformasi input. Selama pelatihan, CNN menggunakan algoritma *backpropagation* untuk mengurangi kesalahan, dan bobot-bobot yang dibagikan dalam CNN mengurangi jumlah parameter yang perlu dipelajari. Dengan demikian, CNN efektif dalam mengenali angka tulisan tangan dan tugas-tugas pemrosesan gambar lainnya. [pattern recognition and machine learning]



Gambar 1 lapisan CNN

D. ReLU Activation

Rectified Linear Unit (ReLU) adalah fungsi aktivasi yang digunakan dalam jaringan saraf dalam pembelajaran mendalam (*deep learning*). Ini adalah fungsi non-linear yang banyak digunakan dalam berbagai aplikasi karena kesederhanaan dan efektivitasnya. ReLU didefinisikan sebagai maksimum antara nol dan nilai input. Jika nilai input negatif, maka outputnya adalah nol, dan jika nilai input positif, maka outputnya adalah nilai input itu sendiri. [Deep Learning Neural Networks for Emotion Classification from Text: Enhanced Leaky Rectified Linear Unit Activation and Weighted Loss]

E. Tensorflow

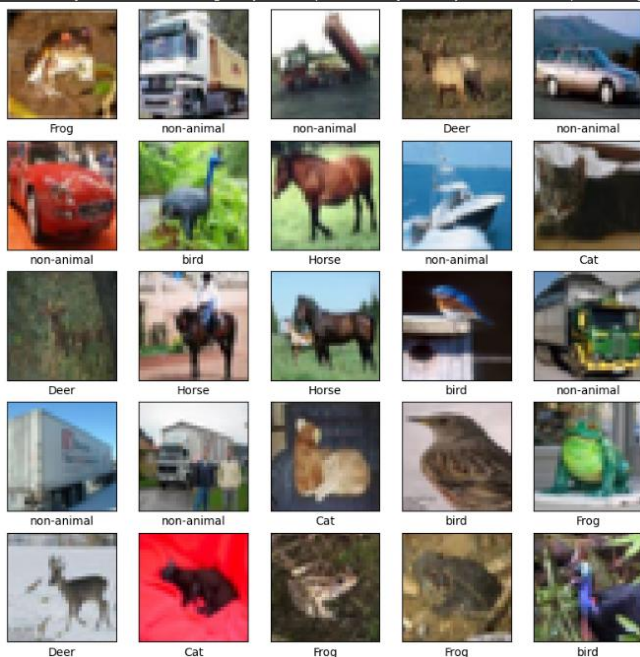
TensorFlow adalah antarmuka untuk mengekspresikan algoritma pembelajaran mesin dan implementasi untuk menjalankan algoritma-algoritma tersebut. Komputasi yang diekspresikan menggunakan TensorFlow dapat dijalankan dengan sedikit atau tanpa perubahan pada berbagai sistem heterogen, mulai dari perangkat mobile seperti ponsel dan tablet hingga sistem

terdistribusi dalam skala besar yang terdiri dari ratusan mesin dan ribuan perangkat komputasi seperti kartu GPU. [TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems] TensorFlow telah digunakan untuk melakukan penelitian dan implementasi sistem pembelajaran mesin dalam lebih dari dua belas bidang ilmu komputer dan bidang lainnya, termasuk pengenalan suara, visi komputer, robotika, pencarian informasi, pemrosesan bahasa alami, ekstraksi informasi geografis, dan penemuan obat secara komputasional.

F. CIFAR10

CIFAR10 adalah dataset komputer visi yang populer yang terdiri dari 60.000 gambar berukuran 32x32 piksel dengan warna, yang dibagi ke dalam 10 kelas, dengan 6.000 gambar per kelas. Kelas-kelasnya adalah: pesawat terbang, mobil, burung, kucing, rusa, anjing, katak, kuda, kapal, dan truk. [CIFAR10 to Compare Visual Recognition Performance between Deep Neural Networks and Humans]

Dataset ini telah digunakan secara luas dalam penelitian tentang pengenalan objek, dan telah diubah menjadi dataset aliran acara (*event-stream*) untuk digunakan dalam penelitian penglihatan neuromorfik. [CIFAR10-DVS: An Event-Stream Dataset for Object Classification] Dataset ini juga telah digunakan untuk mengevaluasi kinerja jaringan saraf dalam perbandingan dengan pengenalan visual manusia. [CIFAR10 to Compare Visual Recognition Performance between Deep Neural Networks and Humans]

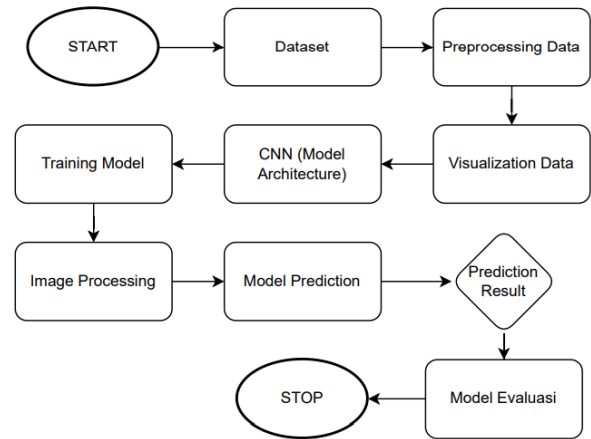


Gambar 2 isi dataset CIFAR10

IV. PERENCANAAN SISTEM

1. Desain Sistem

Dalam sistem deteksi hewan menggunakan algoritma CNN ini menggunakan berbagai proses, diantaranya proses pengambilan data dari dataset, *preprocessing data*, *data cleaning*, pembuatan model, hingga evaluasi model. Program ini dapat digunakan untuk mengklasifikasikan jenis hewan berdasarkan kelompok hewan yang ada pada label dataset CIFAR-10. Berikut diagram blok perencanaan sistemnya:



Gambar 3 diagram blok desain sistem

2. Preprocessing Data

Preprocessing data adalah langkah yang dilakukan untuk mempersiapkan data mentah sebelum diolah atau dianalisis menggunakan model algoritma tertentu. Dengan *preprocessing data*, data yang nantinya digunakan dapat memuat informasi yang lebih bersih dan mudah diakses untuk pemrosesan lebih lanjut, hal ini biasanya dapat berdampak pada hal akhir. Tujuan dari *preprocessing data* antara lain untuk membersihkan data, mentransformasikan data mentah, memformat data, mengisi data kosong (NaN), dan *labelling* data mentah menjadi bentuk data yang cocok untuk diproses nantinya.

2.1. Cleaning data

Cleaning data adalah suatu proses awal sebelum data diproses menggunakan metode atau algoritma tertentu. Biasanya dalam *cleaning data* akan diidentifikasi data yang terdapat dalam dataset. Hal ini karena, pasti ada masalah seperti data yang tidak lengkap, tidak akurat, tidak konsisten atau tidak relevan. Sehingga dengan *cleaning data* ini akan dipastikan kualitas datanya baik untuk digunakan dalam analisis dan pemrosesan data nantinya.

Cleaning data dalam sistem deteksi hewan dengan *labeling class*. *Labeling* adalah suatu proses memberikan label atau kategori pada data. Dalam dataset CIFAR-10 terdapat 10 kelas dengan jenis kelas yang beragam, diantaranya pesawat terbang, mobil, burung, kucing, rusa, anjing, katak, kuda, kapal, dan truk. Sesuai judulnya "sistem deteksi hewan", maka data yang dibutuhkan ialah kelas hewan, yaitu burung, kucing, rusa, anjing, katak, dan kuda. Sehingga untuk kelas yang bukan hewan, diberi label "non-animal". Jadi, model algoritma nantinya akan mengenali hewan untuk mendeteksi hewan apa yang akan diinputkan.

2.2. Normalization Data

Normalisasi data adalah proses mengubah skala data menjadi rentang yang normal untuk memastikan bahwa setiap variabel dalam dataset memiliki nilai yang sebanding dalam proses analisis atau pemodelannya. Dalam sistem deteksi hewan, normalisasi data mengubah nilai piksel dari 0 hingga 255 menjadi 0 – 1 dengan membagi setiap nilai piksel dengan 255. Hal ini dilakukan agar data piksel dalam dataset memiliki rentang yang lebih kecil. Sehingga membantu dalam proses pelatihan model *neural network*.

3. Visualization Data

Setelah dataset CIFAR-10 dilakukan *preprocessing data*, dilanjutkan dengan memvisualisasikan data setelahnya. Dalam hal ini ditampilkan beberapa contoh gambar dari data *training* menggunakan ‘matplotlib.pyplot’. Pertama gambar asli dalam dataset ditampilkan dengan ukuran 10x10 inci. Lalu gambar yang ingin ditampilkan ada 40, sehingga dilakukan iterasi sebanyak 40 kali. Kemudian dibuat subplot dengan ukuran 5 baris dan 8 kolom, ditambah juga label pada sumbu x untuk menunjukkan kelas dari masing-masing contoh gambar yang telah di *preprocessing* dengan *labeling*.

4. CNN (Model Architecture)

Dalam memodelkan data, didefinisikan dulu arsitektur modelnya menggunakan ‘keras.Sequential’. Pada tahapan *preprocessing* dataset telah dinormalisasikan nilainya menjadi 0 dan 1, selanjutnya dibangun model CNN dengan menggunakan beberapa lapisan berikut:

Tabel 1 *layer* sistem model arsitektur

Lapisan	Penjelasan
Con2D	<ul style="list-style-type: none"> Lapisan ini merupakan lapisan konvolusi 2D dengan 32 filter ukuran 3x3. Fungsi aktivasi yang digunakan adalah ReLU (Rectified Linear Unit). input_shape=(32, 32, 3) menentukan dimensi input untuk lapisan ini, yaitu gambar berukuran 32x32 piksel dengan 3 saluran warna (RGB).
BatchNormalization	<ul style="list-style-type: none"> Lapisan ini digunakan untuk normalisasi batch, yaitu mengatur ulang distribusi nilai di setiap batch agar lebih stabil. Normalisasi batch membantu dalam proses pelatihan dan konvergensi model.
Conv2D	<ul style="list-style-type: none"> Lapisan konvolusi 2D dengan 64 filter ukuran 3x3. Fungsi aktivasi yang digunakan adalah ReLU.
BatchNormalization	Lapisan normalisasi batch.

MaxPooling2D	<ul style="list-style-type: none"> Lapisan pemampatan (pooling) maksimum dengan ukuran jendela 2x2. Pemaksimalan dilakukan pada setiap jendela 2x2, sehingga mengurangi dimensi spasial gambar.
Conv2D	<ul style="list-style-type: none"> Lapisan konvolusi 2D dengan 128 filter ukuran 3x3. Fungsi aktivasi yang digunakan adalah ReLU.
BatchNormalization	Lapisan normalisasi batch.
Conv2D	<ul style="list-style-type: none"> Lapisan konvolusi 2D dengan 256 filter ukuran 3x3. Fungsi aktivasi yang digunakan adalah ReLU.
BatchNormalization	Lapisan normalisasi batch.
MaxPooling	Lapisan pemampatan maksimum dengan ukuran jendela 2x2.
Flatten	<ul style="list-style-type: none"> Lapisan ini digunakan untuk meratakan output dari lapisan sebelumnya menjadi vektor satu dimensi. Output dari lapisan ini akan menjadi input untuk lapisan-lapisan densitas selanjutnya.
Dense	<ul style="list-style-type: none"> Lapisan densitas dengan 256 unit neuron. Fungsi aktivasi yang digunakan adalah ReLU.
BatchNormalization	Lapisan normalisasi batch.
Dense	<ul style="list-style-type: none"> Lapisan densitas dengan 10 unit neuron. Tidak ada fungsi aktivasi yang ditentukan secara eksplisit di sini, sehingga output dari lapisan ini berupa nilai-nilai linier. Lapisan ini merupakan output dari model dan akan menghasilkan probabilitas kelas menggunakan fungsi softmax.

Arsitektur model ini juga dikompilasi dengan menggunakan optimizer Adam (*Adaptive Moment Estimation*), *loss function* *Sparse Categorical Crossentropy* untuk klasifikasi dengan banyak kelas diskrit dan metrik akurasi. Dalam optimizer, digunakan juga beberapa optimizer yang lain seperti RMSprop (*Root Mean Square Propagation*), SGD (*Stochastic Gradient Descent*), Adagrad (*Adaptive Gradient*), dan AdaDelta.

5. Training Model

Lanjutan dari proses sebelumnya, model di-fit ke data pelatihan selama sejumlah *epochs* (misalnya 20-50). Semakin banyak *epoch* yang dilakukan, berarti semakin banyak juga model berlatih menggunakan training model dataset dan dapat belajar pola yang lebih kompleks. Namun, terlalu banyak *epoch* juga bisa mengakibatkan *overfitting*. Sehingga sebaiknya jumlah *epoch* yang digunakan tepat tergantung pada karakteristik dataset dan kompleksitas model. Selama proses pelatihan, model akan terus mengoptimalkan parameter internalnya dengan menggunakan optimizer Adam dan menghitung *loss* untuk mengukur sejauh mana prediksi model yang cocok dengan label yang sesungguhnya. Evaluasi pada dataset val (validasi) dilakukan untuk mengukur performa model pada data yang belum pernah dilihat.

6. Image Processing

Dalam *image processing*, gambar yang diinputkan diubah ukurannya menjadi 32x32 piksel. Kemudian gambar diubah menjadi array numpy dan dinormalisasi dengan membagi setiap pikselnya dengan 255.0. lalu *array* pada gambar direshape menjadi bentuk 1,32,32,3 untuk menyesuaikan dengan format input yang diterima oleh model. Sehingga gambar siap untuk dimasukkan dalam model untuk memprediksi kelas yang diinginkan.

7. Model Prediction

Setelah gambar diproses, gambar dimasukkan ke dalam model prediksi yang menghasilkan probabilitas kelas untuk gambar yang telah diinputkan dan diolah dalam *image processing*. Kemudian diambil indeks kelas dengan probabilitas yang paling tinggi menggunakan fungsi 'np.argmax()'. Barulah dapat ditentukan label kelas yang sesuai berdasarkan indeks yang ada dan gambar yang dihasilkan diprediksi menggunakan model yang telah dibuat sebelumnya.

8. Prediction Result

Prediction Result berisi label yang diprediksi atau hasil prediksi dari model yang telah dibangun sebelumnya.

9. Model Evaluation

Dalam *model evaluation*, terdapat hasil nilai *loss* dan akurasi pada test set menggunakan model evaluasi. Kemudian hasil dari evaluasi ditampilkan dalam bentuk angka. Semakin besar hasil evaluasi, maka tingkat akuransinya atau kebenaran hasil prediksinya juga semakin baik. Hal ini, dapat diketahui sejauh mana model dapat memprediksi dengan akurat pada data atau gambar yang belum ada dalam dataset sebelumnya atau dalam data trainingnya.

V. HASIL

1. Deskripsi Dataset

Dataset yang digunakan dalam penelitian ini adalah dataset CIFAR-10, yang terdiri dari 60.000 gambar berukuran 32x32 piksel, yang dibagi menjadi sepuluh kelas yang berbeda, dengan enam ribu gambar per-kelasnya. Dataset ini telah dibagi menjadi 50.000 gambar latih dan 10.000 gambar uji.

2. Metrik Evaluasi

Untuk mengevaluasi kinerja model klasifikasi gambar, kami menggunakan metrik akurasi, yaitu proporsi gambar yang diklasifikasikan dengan benar dibandingkan dengan total gambar dalam subset pengujian.

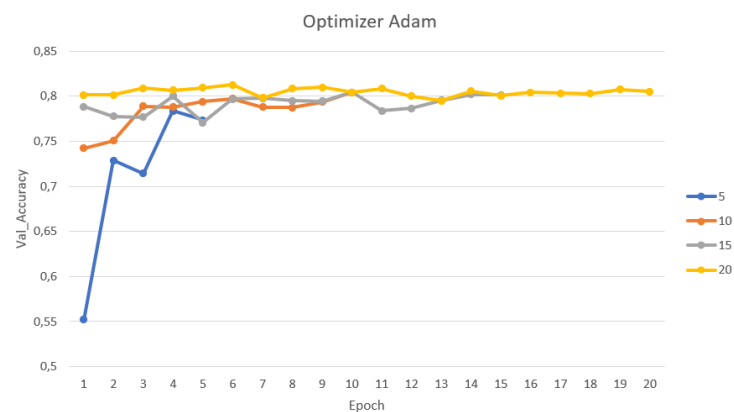
2.1. Implementasi Model

Kami mengimplementasikan model Convolutional Neural Network (CNN) menggunakan kerangka kerja TensorFlow. Model kami terdiri dari beberapa lapisan konvolusi dan pooling, diikuti oleh beberapa lapisan terhubung penuh dan lapisan dropout untuk mengurangi *overfitting*. Kami menggunakan fungsi aktivasi ReLU di setiap lapisan konvolusi dan lapisan terhubung penuh.

2.2. Hasil Eksperimen

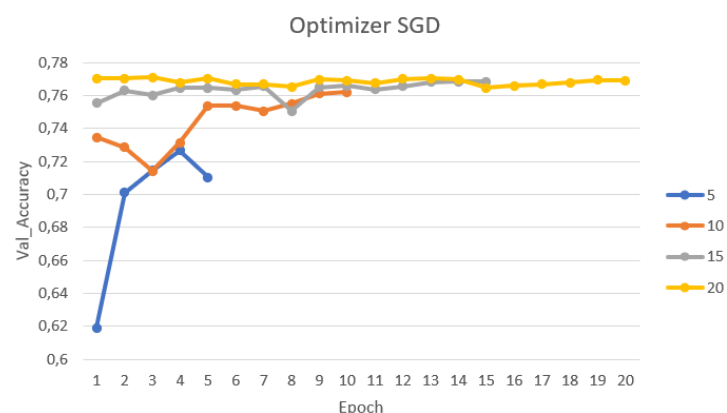
Dalam pelatihan model sistem deteksi hewan menggunakan algoritma CNN, kami menggunakan optimizer yang berbeda-beda dengan *epoch* yang berbeda pula dengan set *epoch* 5, 10, 15, dan 20. Berikut adalah hasil dari evaluasi model kami:

2.2.1. Menggunakan optimizer Adam, berikut diagram garisnya:



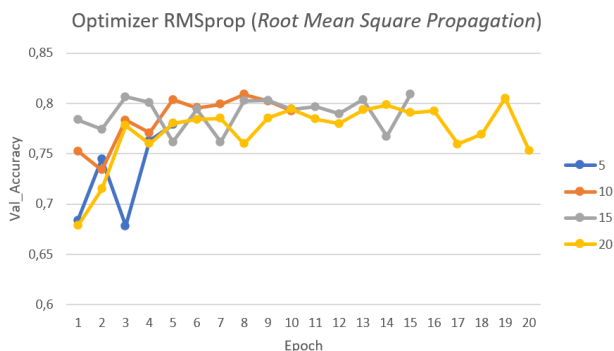
Gambar 4 grafik akurasi optimizer adam terhadap epoch

2.2.2. Menggunakan optimizer SGD, berikut diagram garisnya



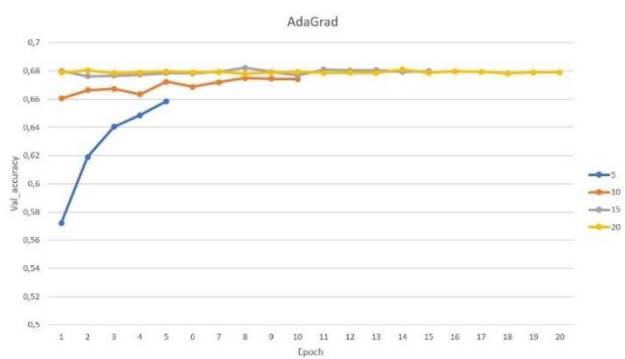
Gambar 5 grafik akurasi optimizer SGD terhadap epoch

2.2.3. Menggunakan optimizer RMSprop (Root Mean Square Propagation), berikut diagram garisnya



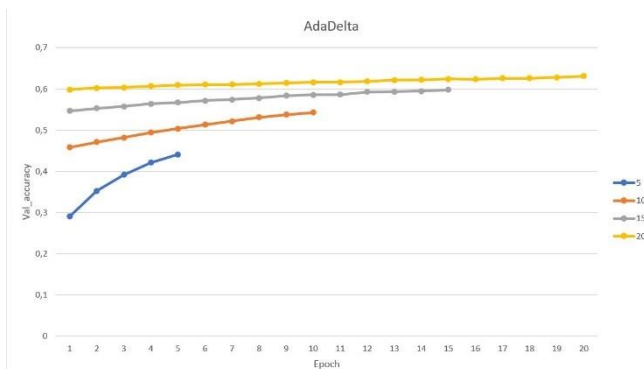
Gambar 6 grafik akurasi optimizer RMSprop terhadap epoch

2.2.4. Menggunakan optimizer AdaGrad, berikut diagram garisnya:



Gambar 7 grafik akurasi optimizer AdaGrad terhadap epoch

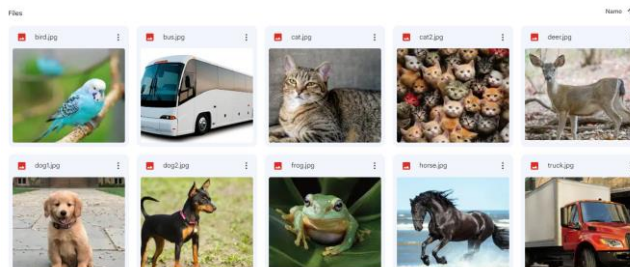
2.2.5. Menggunakan optimizer AdaDelta, berikut diagram garisnya:



Gambar 8 grafik akurasi optimizer AdaDelta terhadap epoch

Dari beberapa pengujian dengan menggunakan beberapa optimizer dengan *epoch* yang berbeda-beda, didapatkan hasil bahwa optimizer yang paling stabil dan mendapatkan *val_accuracy* yang paling tinggi adalah optimizer Adam. Optimizer RMSprop hasil *val_accuracy* nya juga tinggi, tapi hasilnya tidak stabil. Sehingga pengujian selanjutnya akan digunakan optimizer Adam dengan *epoch* yang berbeda-beda.

Pengujian selanjutnya menguji dengan 10 gambar yang berbeda-beda dan menentukan seberapa besar total akurasi dari 10 gambar tersebut. Berikut 10 gambar yang akan diujikan dalam sistem deteksi ini:



Gambar 9 gambar pengujian di luar dataset CIFAR10

Untuk pengujian ini menggunakan optimizer Adam dengan *epoch* yang dibedakan. Berikut hasil pengujiannya:

Tabel 2 hasil pengujian gambar di luar dataset CIFAR10

Objek	Epoch			
	5	10	15	20
Bird	v	v	v	v
Bus	v	v	v	v
Cat	v	v	v	v
Cat_2	x	x	x	x
Deer	x	v	v	x
Dog	v	v	v	v
Dog_2	v	x	x	x
Frog	v	v	v	v
Horse	v	x	v	v
Truck	v	v	v	v
Accuracy	0.8	0.7	0.8	0.7

2.3. Analisis Hasil

Dari hasil eksperimen, kami melihat *val_accuracy* pada model arsitektur dengan menggunakan 5 optimizer yang berbeda. Hasilnya menunjukkan bahwa pada penggunaan optimizer Adam dengan berbagai konfigurasi *epoch*, diperoleh hasil akurasi validasi (*val_accuracy*) yang bervariasi. Dalam skenario ini, pada penggunaan 5 *epoch*, akurasi validasi meningkat dari 0.5521 menjadi 0.7734. Pada penggunaan 10 *epoch*, akurasi validasi terus meningkat dari 0.7424 menjadi 0.8043. Penggunaan 15 *epoch* menghasilkan akurasi validasi yang fluktuatif antara 0.7705 dan 0.8094. Penggunaan 20 *epoch* menunjukkan akurasi validasi yang stabil pada kisaran 0.8002 hingga 0.8126. Oleh karena itu, dapat disimpulkan bahwa dengan peningkatan jumlah *epoch*, akurasi validasi dapat meningkat secara bertahap, meskipun pada titik tertentu peningkatan tersebut tidak signifikan.

Pada penggunaan optimizer SGD dengan berbagai konfigurasi *epoch*, diperoleh hasil akurasi validasi (*val_accuracy*) yang berbeda-beda. Pada penggunaan 5 *epoch*, akurasi validasi meningkat dari 0.6189 menjadi 0.7102. Pada penggunaan 10 *epoch*, akurasi validasi terus meningkat dari 0.7347 menjadi 0.7621. Penggunaan 15 *epoch* menunjukkan akurasi validasi yang fluktuatif antara 0.7554 dan 0.7685. Penggunaan 20 *epoch* menunjukkan akurasi validasi yang stabil pada kisaran 0.7705 hingga 0.7691. Oleh karena itu,

dapat disimpulkan bahwa dengan peningkatan jumlah *epoch*, akurasi validasi cenderung meningkat secara bertahap, namun pada titik tertentu peningkatan tersebut cenderung stagnan atau bahkan menurun.

Dalam penggunaan optimizer RMSprop pada model tersebut, terdapat hasil *val_accuracy* yang bervariasi selama beberapa *epoch*. Pada *epoch* pertama, *val_accuracy* mencapai 0.6837. Pada *epoch* kedua, *val_accuracy* meningkat menjadi 0.7447. Namun, pada *epoch* ketiga, *val_accuracy* mengalami penurunan menjadi 0.6782. Pada *epoch* keempat, *val_accuracy* kembali meningkat menjadi 0.7631. Pada *epoch* kelima, *val_accuracy* mencapai 0.7787. Pada *epoch* kesepuluh, *val_accuracy* meningkat menjadi 0.7523. Pada *epoch* kelima belas, *val_accuracy* mencapai 0.7836. Pada *epoch* kedua puluh, *val_accuracy* menurun menjadi 0.7533. Terlihat bahwa *val_accuracy* pada model dengan optimizer RMSprop ini cenderung naik turun dan tidak menunjukkan tren yang konsisten selama *epoch* yang diamati.

Dalam penggunaan optimizer AdaGrad, terdapat peningkatan *val_accuracy* dari 5 *epoch* ke *epoch* berikutnya. Pada *epoch* pertama, *val_accuracy* mencapai 0.5722, namun mengalami peningkatan menjadi 0.6586 pada *epoch* kelima. Peningkatan tersebut berlanjut hingga *epoch* kesepuluh dengan *val_accuracy* mencapai 0.6743. Setelah itu, terdapat fluktuasi kecil dalam *val_accuracy*, namun tetap dalam kisaran yang sama. Pada *epoch* terakhir, yaitu *epoch* kedua puluh, *val_accuracy* mencapai 0.6791. Hal ini menunjukkan bahwa model yang menggunakan optimizer AdaGrad memiliki performa yang baik dalam memprediksi data yang belum pernah dilihat sebelumnya. Akan tetapi, nilai *val_accuracy* nya tidak setinggi optimizer sebelumnya.

Dalam eksperimen menggunakan optimizer AdaDelta, terdapat peningkatan *val_accuracy* seiring dengan peningkatan jumlah *epoch*. Pada *epoch* pertama, *val_accuracy* sebesar 0.2906. Kemudian, *val_accuracy* mengalami kenaikan setiap *epoch*nya, mencapai 0.6309 pada *epoch* kedua puluh. Hal ini menunjukkan bahwa model yang menggunakan optimizer AdaDelta mampu belajar dan meningkatkan kinerjanya dalam mengklasifikasikan data validasi. Namun, nilai *val_accuracy* nya tidak setinggi optimizer sebelumnya.

Berdasarkan pengujian gambar yang digunakan dari luar dataset CIFAR10, akurasi yang dihitung secara manual nilainya tidak jauh berbeda dengan akurasi testing yang dihitung menggunakan algoritma *CNN*. Jika akurasi yang dihitung program pada lima *epoch* mencapai 0,7734, pada pengujian aslinya mencapai 0,8. Terdapat miss pada gambar *cat_2* dan *deer*. Kemudian pada sepuluh *epoch*, akurasi yang terhitung menggunakan program mencapai 0,8043, sedangkan pengujian aslinya hanya 0,7. Terdapat miss pada gambar *cat_2*, *dog_2*, dan *horse*. Kemudian pada lima belas *epoch*, akurasi yang terhitung menggunakan program mencapai 0,8014, sedangkan pengujian aslinya mencapai 0,8. Terdapat miss pada gambar *cat_2* dan *dog_2*. Terakhir, pada dua puluh *epoch*, akurasi yang terhitung menggunakan program mencapai

0,8051, sedangkan pengujian aslinya hanya mencapai 0,7. Terdapat miss pada gambar *cat_2*, *deer*, dan *dog_2*.

2.4. Pembahasan

Hasil eksperimen kami menunjukkan bahwa model *CNN* yang diimplementasikan dapat mencapai kinerja yang baik dalam tugas klasifikasi gambar CIFAR-10. Namun, terdapat ruang untuk peningkatan performa pada beberapa kelas tertentu. Faktor-faktor yang dapat mempengaruhi kinerja model meliputi ukuran dataset pelatihan, arsitektur model, dan hiperparameter yang digunakan.

KESIMPULAN

Dalam proyek tugas besar ini, kami berhasil mengembangkan dan menerapkan sistem deteksi hewan menggunakan algoritma *Convolutional Neural Networks* (*CNN*) yang mampu mengklasifikasikan spesies hewan secara otomatis berdasarkan gambar digital. Eksperimen kami menunjukkan bahwa model *CNN* yang kompleks yang kami gunakan yang cukup baik, sekitar 80. Hasil penelitian ini menunjukkan potensi besar sistem deteksi hewan berbasis *CNN* dalam aplikasi deteksi dan klasifikasi hewan secara otomatis, seperti pemantauan satwa liar, konservasi alam, atau penelitian ekologi. Meskipun tantangan seperti keterbatasan data dan peningkatan efisiensi komputasi perlu diatasi, penelitian mendatang diharapkan dapat memperluas aplikasi sistem deteksi hewan ini dengan lebih baik.

REFERENSI

- [1] H. Dhika, N. R. Kurnianda, P. Irfansyah and W. Ananta, "Model Prediksi Jenis Hewan dengan Metode Convolution Neural Network," vol. 9, p. 1, 2020.
- [2] M. A. A. Fawwaz, S. M. K. N. Ramadhani and S. M. F. Sthevanie, "Klasifikasi Ras pada Kucing menggunakan Algoritma Convolutional Neural Network (CNN)," *Jurnal Tugas Akhir Fakultas Informatika*, vol. 8, no. 2355-9365, p. 715, 2021.
- [3] I. W. Suartika, A. Y. Wijaya and R. Soelaiman, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101," *Jurnal Teknik ITS*, vol. 5, no. 1, pp. A65-A69, 2016.
- [4] J. B. J. R. Openg, M. Endah H. and H. , "Klasifikasi Unggas Ordo Anseriformes Berdasarkan Citra Menggunakan Metode Deep Learning Dengan Algoritma Convolutional Neural Network (CNN)," *Jurnal SINTaKS*, 2022.
- [5] S. T. E. Putri and A. Fachrurrozi, "PENDETEKSIAN OBJEK PADA CITRA HEWAN KARNIVORA DAN HERBIVORA MENGGUNAKAN FASTER R-CNN," *Jurnal Ilmiah Informatika Komputer*, vol. 27, no. 1, p. 2, 2022.
- [6] A. Leovincen and Y. , "Klasifikasi Ras Anjing Berdasarkan Citra Menggunakan Convolutional Neural Network," *Jurnal Algoritme*, vol. 3, no. 2, pp. 160-169, 2023.

- [7] R. Bost, R. A. Popa, S. Tu and S. Goldwasser, Machine Learning Classification over Encrypted Data.
- [8] H. Yang, A. Alsadoon, P. Prasad, T. Al-Dala'in, T. A. Rashid, A. Maag and O. H. Alsadoon, "Deep Learning Neural Networks for Emotion Classification from Text: Enhanced Leaky Rectified Linear Unit Activation and Weighted Loss," 2022.
- [9] J. Dean and R. Monga, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *Preliminary White Paper*, pp. 1-19, 2015.
- [10] T. Ho-Phuoc, "CIFAR10 to Compare Visual Recognition Performance between Deep Neural Networks and Humans," pp. 1-10.
- [11] H. Li, H. Liu, X. Ji, G. Li and L. Shi, "CIFAR10-DVS: An Event-Stream Dataset for Object Classification," *frontiers in Neuroscience*, vol. 11, pp. 1-10, 2017.
- [12] F. Hasyim, K. Malik, F. Rizal and Y. , "Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi Batik," *Jurnal Kecerdasan Buatan, Komputasi, dan Teknologi Informasi*, vol. X, no. X, pp. 1-8, 20XX.
- [13] C. M. Bishop, Pattern Recognition and Machine Learning, Springer New York, 2006.