

DETEKSI DAN PENGENALAN IKAN MENGGUNAKAN ALGORITMA *CONVOLUTIONAL NEURAL NETWORK*

R. Mehindra Prasmatio¹⁾, Basuki Rahmat²⁾, Intan Yuniar³⁾

E-mail: ¹⁾mehindraaa@gmail.com, ²⁾basukirahmat.if@upnjatim.ac.id,

³⁾intanyuniar.if@upnjatim.ac.id

^{1, 2, 3} Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur

Abstrak

Identifikasi dan pengenalan identitas jenis ikan secara otomatis merupakan suatu persoalan besar yang menarik dan banyak sekali berbagai macam pendekatan untuk menyelesaikan persoalan ini. Apalagi di dalam akuarium suatu akuarium terdapat banyak jenis objek ikan maupun hiasan akuarium. Ikan tampak hampir sama di beberapa jenis. Tingkat kejernihan air juga sangat diperhitungkan untuk memperlancar proses klasifikasi. Maka dari harus disediakan akuarium yang memiliki air jernih dan pencahayaan tepat agar mempermudah proses ini. Hal-hal tersebutlah yang membuat permasalahan klasifikasi menjadi lebih sulit diselesaikan. Dalam penelitian ini digunakan salah satu metode *deep learning* yaitu *Convolutional Neural Networks* (CNN) sebagai pengidentifikasian ikan secara *real-time* yang terbukti efisien dalam klasifikasi ikan. Metode diimplementasikan dengan bantuan *library* OpenCV untuk deteksi objek dan perangkat kamera. Pada penelitian ini dilakukan 6 kali percobaan training untuk ditemukan nilai paling baik, dan mendapatkan nilai *test score* 2.475, *test accuracy* 0.4237 dan *loss* sebesar 2.2002. Data yang digunakan pada penelitian ini berupa tangkapan gambar dari hasil video secara langsung/*realtime* menggunakan *webcam*. Penelitian ini menghasilkan tingkat akurasi sebesar 85,18% dengan pengujian 27kali yang dimana 4kali tidak dapat mengidentifikasi foto dan 23 kali berhasil dalam mengidentifikasi foto ikan.

Kata kunci: *convolutional neural network, pre processing, citra*

1. PENDAHULUAN

Indonesia merupakan salah satu negara dengan keanekaragaman jenis ikan nya yang melimpah. Letak perairan yang berada di wilayah khatulistiwa dan beriklim tropis membuat Indonesia memiliki kekayaan aneka jenis biota air yang lebih banyak dibandingkan dengan daerah dingin maupun subtropis. Diperkirakan total jenis ikan di dunia mencapai 40.000 jenis (Nelson, 2006). Berdasarkan data yang terdapat pada *Wikipedia.org* hingga 22 april 2020, terdapat 27.000 jenis ikan di dunia yang berhasil dideskripsikan. Jumlah jenis ikan yang tersebar di Indonesia yaitu 4857 spesies terdiri dari 1225 spesies air tawar dan 3632 spesies air laut.

Berdasarkan fakta jumlah pakar ikan di Indonesia yang jumlahnya terbatas sehingga membuat sedikitnya pengetahuan literatur dalam menentukan prinsip-prinsip identifikasi ikan di Indonesia. Hal ini ditambah dengan kemiripan-kemiripan antara jenis ikan satu dengan yang lainnya sehingga perlu identifikasi menyeluruh untuk membedakannya. Oleh karena itu diperlukan suatu sistem yang dapat membantu tugas seorang pakar dalam mengidentifikasi jenis ikan secara terotomasi. Sistem identifikasi dan pengenalan ikan yang akan dibangun diharapkan dapat membantu mempermudah pengamat atau peneliti mengidentifikasi ikan tanpa harus bertemu dengan pakar secara langsung.

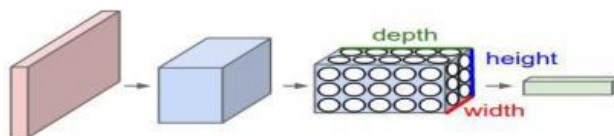
Program akan menggunakan algoritma *Convolutional Neural Network (CNN)* yang merupakan bagian dari *Multi-Layer Perceptron*. *CNN* adalah salah satu jenis perhitungan *neural network* yang sering digunakan pada pengolahan citra untuk mendeteksi dan mengenali objek pada sebuah image. Jadi sistem ini akan menggunakan pengolahan citra sebagai proses identifikasi 9 jenis ikan air tawar. Dari pengolahan citra tersebut maka akan didapatkan 9 parameter yaitu Ikan Arwana, Ikan Comet, Ikan Guppy, Ikan Koki, Kura-Kura, Ikan Molly, Ikan Manfish, Ikan Platys dan Ikan Redfin. Sehingga jumlah klasifikasi pada penelitian ini mencapai 9 kemungkinan klasifikasi.

2. Metodologi

Dalam penelitian yang dilakukan penulis menggunakan algoritma *convolutional neural network* sebagai algoritma pengembangan aplikasi yang berguna dalam melakukan identifikasi dan klasifikasi jenis-jenis ikan.

2.1 Convolutional Neural Network

Merupakan jaringan saraf yang dikhususkan untuk memproses data yang memiliki grid. Sebagai contoh adalah berupa citra dua dimensi. Nama konvolusi sendiri merupakan operasi aljabar linear yang mengkalikan matriks dari filter pada citra yang akan di proses. Proses ini disebut lapisan konvolusi dan merupakan salah satu jenis dari banyak lapisan bisa dimiliki dalam satu jaringan. Lapisan konvolusi merupakan lapisan utama yang paling penting untuk digunakan. Jenis lapisan lain yang biasa digunakan adalah *Pooling Layer*, yakni lapisan yang digunakan untuk mengambil nilai maksimal atau nilai rata-rata dari bagian-bagian piksel pada citra. Berikut adalah contoh jaringan pada *convolutional neural network*.



Gambar 2. 1. Contoh Jaringan *Convolutional Neural Network*

Pada gambar 2.1 menunjukkan setiap lapisan *input* yang dimasukkan memiliki volume yang berbeda dan mewakili dengan kedalaman, tinggi dan lebar. Setiap besaran yang didapatkan tergantung dari hasil filtrasi dari lapisan sebelumnya dan juga banyak filter yang digunakan. Model jaringan seperti ini sudah terbukti sangat ampuh dalam menangani permasalahan klasifikasi citra.

2.2 Operasi Konvolusi

Operasi konvolusi adalah operasi pada dua fungsi argument bernilai nyata (Goodfellow, I., Bengio, Y., & Courville, A.). Operasi ini menerapkan fungsi output sebagai *Feature Map* dari *input* citra. *Input* dan *output* ini dapat dilihat sebagai dua argumen bernilai riil. Secara formal operasi konvolusi dapat ditulis dengan rumus berikut.

$$s(t) = (x * \omega)(t) \quad (2.1)$$

Fungsi $s(t)$ memberikan *output* tunggal berupa *Feature Map*, argument pertama adalah *input* yang merupakan x dan argument kedua w sebagai kernel atau filter. Jika kita melihat *input* sebagai citra dua dimensi, maka kita bisa mengasumsikan t sebagai pixel dan menggantinya dengan i dan j . Operasi untuk konvolusi ke *input* dengan lebih dari satu dimensi dapat ditulis sebagai berikut.

$$S(i, j) = (K * I)(i, j) = \sum \sum l(i - m, j - n)K(m, n) \quad (2.2)$$

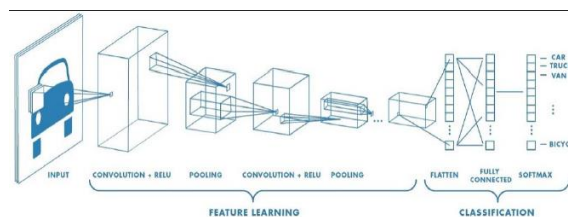
Persamaan diatas adalah perhitungan dasar dalam operasi operasi konvolusi dimana i dan j adalah piksel dari citra. Perhitungannya bersifat komunikatif dan muncul saat K sebagai kernel, I sebagai *input* dan kernel yang dapat dibalik relative terhadap input. Sebagai alternatif, operasi konvolusi dapat dilihat sebagai perkalian matriks antaran citra masukan dan kernel dimana keluarannya dapat dihitung dengan *dot product*.

2.3. Arsitektur Jaringan CNN

Fungsi $s(t)$ memberikan *output* tunggal berupa *Feature Map*, argument pertama adalah *input* yang merupakan x dan argument kedua w sebagai kernel atau filter. Jika kita melihat *input* sebagai citra dua dimensi, maka kita bisa mengasumsikan t sebagai pixel dan menggantinya dengan i dan j . Operasi untuk konvolusi ke *input* dengan lebih dari satu dimensi dapat ditulis sebagai berikut.

$$S(i, j) = (K * I)(i, j) = \sum \sum l(i - m, j - n)K(m, n) \quad (2.3)$$

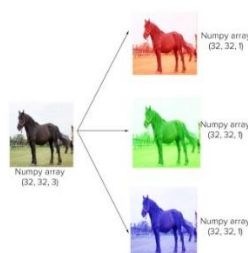
Persamaan diatas adalah perhitungan dasar dalam operasi operasi konvolusi dimana i dan j adalah piksel dari citra. Perhitungannya bersifat komunikatif dan muncul saat K sebagai kernel, I sebagai *input* dan kernel yang dapat dibalik relative terhadap input. Sebagai alternatif, operasi konvolusi dapat dilihat sebagai perkalian matriks antaran citra masukan dan kernel dimana keluarannya dapat dihitung dengan *dot product*. Arsitektur dari CNN juga terbagi menjadi 2 bagian besar yaitu *Feature Extraction Layer* dan *Fully-Connected Layer (MLP)*. Seperti pada gambar 2.2 berikut.



Gambar 2.2. Tampilan Arsitektur Convolutional Neural Network (CNN)

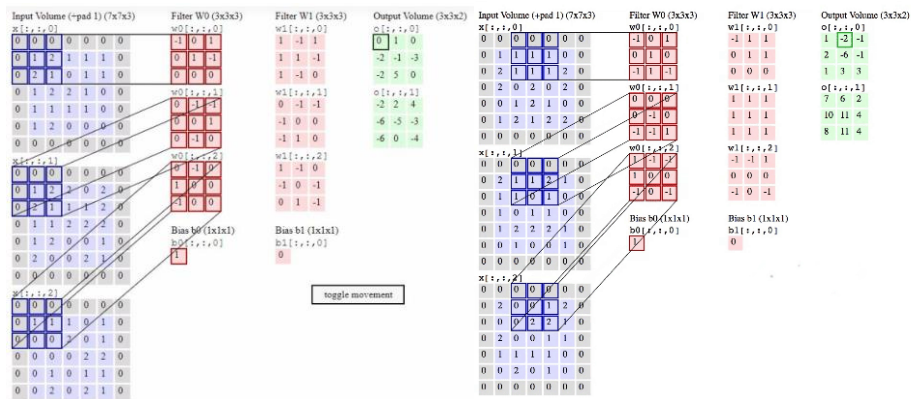
Sebuah CNN terdiri dari beberapa *layer*. Berdasarkan arsitektur LeNet5 [8], terdapat empat macam *layer* utama pada sebuah CNN

A) Convolution Layer



Gambar 2.3 Convolution Layer.

Gambar 2.4 diatas adalah RGB (*Red, Green, Blue*) image berukuran 32x32 pixels yang sebenarnya adalah *multidimensional array* dengan ukuran 32x32x3 (3 adalah jumlah *channel*). Sebagai contoh, layer pertama pada *feature extraction layer* biasanya adalah *conv. layer* dengan ukuran 5x5x3. Panjang 5 pixels, tinggi 5 pixels dan tebal atau jumlah 3 buah sesuai dengan *channel* dari image tersebut. Ketiga filter ini akan digeser keseluruhan bagian dari gambar. Setiap pergeseran akan dilakukan operasi “dot” antara *input* dan nilai dari filter tersebut sehingga menghasilkan sebuah *output* atau biasa disebut sebagai *activation map* atau *feature map*. proses dari *feature map* seperti pada 2.4 berikut.



Gambar 2.4. Feature Map

Pada gambar 2.5 dijelaskan bagaimana pergeseran *stride* tiap pixel nya. Pada *feature map* diatas menggunakan 2 *stride* sehingga akan melompat 2 kotak tiap pergeserannya. Semakin kecil nilai *stride* maka akan semakin bagus hasil nilai yang didapatkan. Tiap pergeseran akan dijumlahkan dengan *filter* sehingga akan menghasilkan *output*. Untuk menghitung dimensi dari *feature map*, bisa gunakan rumus 2.3 dibawah ini:

$$output = \frac{W-N+2P}{s} + 1 \quad (2.3)$$

W = Panjang atau Tinggi *input*

N = Panjang atau Tinggi *filter*

P = Zero *Padding*

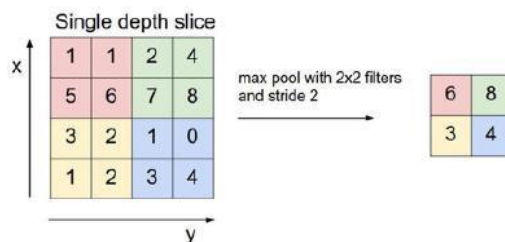
S = *Stride*

B) Pooling Layer

Pooling layer dilakukan untuk mengurangi varians, mengurangi kerumitan perhitungan dan ekstrak fitur tingkat rendah dari lingkungan sekitar. Pooling layer terdapat dua jenis yaitu Max pooling dan Average pooling. Max pooling mengekstrak fitur yang paling penting seperti tepi sedangkan average pooling lebih halus dibandingkan dengan max pooling (Rahman, 2020).

Meskipun keduanya digunakan untuk alasan yang sama, max pooling lebih baik untuk mengekstrak fitur ekstrem. Average pooling kadang-kadang tidak dapat mengekstrak fitur yang baik karena dibutuhkan semua untuk menghitung dan menghasilkan nilai rata-rata yang mungkin atau tidak penting untuk deteksi objek. Average pooling membawa semua ke dalam hitungan dan mengalirkannya ke lapisan berikutnya yang berarti semua nilai benar-benar digunakan untuk pemetaan fitur dan pembuatan keluaran yang merupakan perhitungan yang sangat umum. Jika tidak diperlukan semua masukan dari lapisan konvolusi maka akan mendapatkan akurasi yang buruk untuk average pooling. Pada

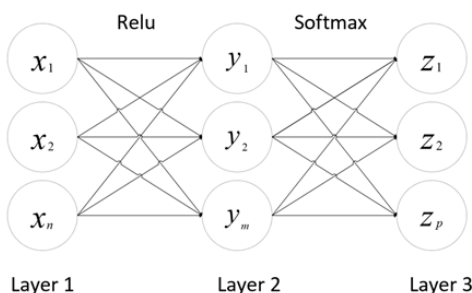
klasifikasi objek pooling layer yang banyak digunakan adalah average pooling. Pemilihan pooling layer tergantung pada jenis dataset yang digunakan (Rahman, 2020).



Gambar 2.5. *Pooling Layer*

C) *Fully Connected Layer*

Proses pada *Fully Connected Layer* berfungsi untuk melakukan proses klasifikasi yaitu dengan menggunakan *softmax* yang sesuai dengan proses yang digambarkan pada gambar 2.6.



Gambar 2.6. *Fully Connected Layer*

Dari gambar 2.6 diatas, dimana *layer 1* akan dilakukan *feedforwarding* menuju *layer 2* dengan menggunakan fungsi aktivasi RelU. Pada *layer 2* akan dilakukan klasifikasi dengan menggunakan *softmax*.

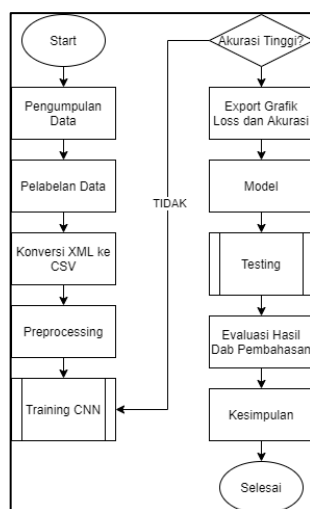
Lapisan *Fully-Connected* biasanya digunakan pada metode *Multi* lapisan *Perceptron* dan bertujuan untuk mengolah data sehingga bisa diklasifikasikan. Perbedaan antara lapisan *Fully-Connected* dan lapisan konvolusi biasa adalah *neuron* di lapisan konvolusi terhubung hanya ke daerah tertentu pada input. Sementara lapisan *Fully-Connected* memiliki *neuron* yang secara keseluruhan terhubung. Namun, kedua lapisan tersebut masih mengoprasikan produk *dot*, sehingga fungsinya tidak begitu berbeda. (Kefin Pudi D, 2013).

2.4. *Jenis dan Sumber Data*

Pada penelitian ini akan membagi tiap jenis ikan hias yang sudah ditentukan. Terdapat 9 jenis ikan pada penelitian ini dan jenis-jenis ikan hias tersebut adalah; ikan arwana, ikan *comet*, ikan guppy, ikan koki, kura-kura brazil, ikan *molly*, ikan *manfish*, ikan *platys* dan ikan *redfin*. Pada tiap jenis ikan akan digolongkan pada *folder* sesuai dengan nama-nama ikan tersebut. Pada tiap *folder* diberi masing-masing 100 gambar yang telah diunduh dari *google* sebagai *dataset*. Sehingga akan menghasilkan total 900 gambar *dataset*.

Kemudian, tiap-tiap *folder dataset* tersebut akan dibagi untuk data *train*, data *test*, dan data *validation* masing-masing dengan perbandingan 80:20 atau 720 gambar untuk data *training* dan 180 gambar untuk data *test*. Setelah itu gambar-gambar tersebut akan diatur dimensinya menjadi 50x50.

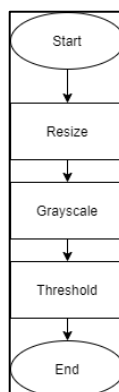
2.5. Alur Penelitian



Gambar 3. 1. Alur penelitian

Pada gambar 3.1 menjelaskan bahwa pada penelitian ini akan dimulai dengan tahap pengumpulan data, yaitu dataset jenis-jenis ikan. Setelah dataset telah terkumpul, kemudian dilakukan tahap pelabelan data. Dalam tahap pelabelan data ini menggunakan labelimg citra satu persatu. Pelabelan ini akan menghasilkan file XML yang tiap citra nya memiliki isi yang berbeda-beda. Kemudian dikonversi menjadi CSV agar kumpulan XML ini dapat dijadikan 1 file saja. Setelah dilakukannya convert, kemudian dilakukan tahap *preprocessing* yang akan memodifikasi citra sesuai dengan yang dibutuhkan. Setelah seluruh citra sudah sesuai dengan yang dibutuhkan, lalu dilakukan tahap pelatihan menggunakan metode *convolutional neural network*. Jika hasil pelatihan menghasilkan akurasi yang tinggi maka lanjut, jika belum sesuai sasaran maka lakukan pelatihan kembali. Setelah akurasi tinggi, maka akan menampilkan grafik akurasi dan *loss* saat *training*. Kemudian jika sudah, maka akan secara otomatis model tersimpan. Kemudian *model* tersebut dilakukan *testing*. Dalam *testing* juga terdapat *setup* kamera untuk memprediksi. Setelah itu dilakukan prediksi menggunakan foto atau objek secara nyata untuk dihasilkan ke akurasian hasil dari *training* untuk mengukur apakah telah dapat mendeteksi jenis ikan atau belum. Kemudian diakhir dilakukan kesimpulan apakah metode ini dapat dengan baik dan akurat dalam melakukan pendeteksian suatu objek khususnya jenis-jenis ikan.

2.6. Preprocessing

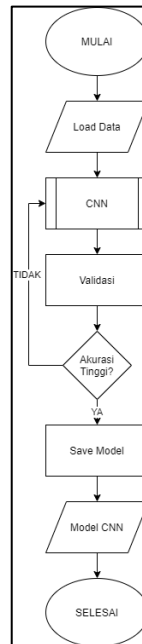


Gambar 3.2. Preprocessing

Preprocessing berfungsi untuk menyiapkan citra agar dapat digunakan pada proses selanjutnya merupakan proses *resize* yang gunanya adalah menyesuaikan semua ukuran citra agar sama. Kemudian dilakukan proses *grayscale* agar citra menjadi berwarna abu-abu. Setelah itu dilakukan proses *thresholding* yang merupakan proses untuk menjadikan nilai citra menjadi 0-1 dari 0-255 yang gunanya mempermudah dalam proses input training

kemudian.

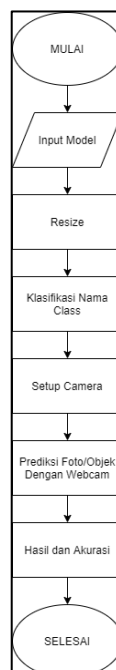
2.7. Training



Gambar 3. 3. Proses Training

Pada gambar 3.3 diatas *training* dimulai dengan *load data* hasil *preprocess* yang telah dilakukan. Kemudian dilakukan tahap *Convolutional Neural Network* dengan berbagai komponen didalamnya. Setelah itu dilakukan proses validasi dengan citra yang telah disiapkan, yaitu 20% sisa dari total gambar untuk *training*. Jika akurasi belum mencapai tujuan, maka proses akan diulangi lagi ke tahap metode CNN hingga akurasi yang didapatkan sesuai. Jika sudah sesuai, maka *model* akan disimpan untuk nantinya proses *testing*.

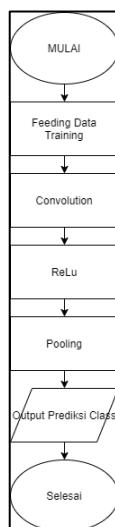
2.8. Testing



Gambar 3. 4. Proses Training

Pada gambar 3.4 dijelaskan proses *testing* yang fungsinya menguji apakah data yang telah di *training* telah berhasil atau belum. Tahap pertama adalah *input model*, tahap ini merupakan *import model* yang telah disimpan sebelumnya setelah tahap *training*. Setelah itu tahap *resize* yang gunanya agar citra dapat disesuaikan dengan ukuran *window webcam*. Kemudian dilakukan klasifikasi nama *class* yang awalnya dataset *training* diberi nama 0-8. Klasifikasi ini mempermudah tahap klasifikasi untuk mempermudah dalam mengenal jenis-jenis ikan. Setelah 0-8 telah diberi nama sesuai dengan keinginan lalu dilakukan *setup* kamera yang bertujuan mengatur ukuran Panjang lebar *window* tampilan *webcam*. Setelah di jalankan, maka *webcam* akan terbuka dan langsung dapat *memprediksi gambar atau objek*.

2.9. Convolutional Neural Network



Gambar 3.5. CNN

Pada gambar 3.5 merupakan tahap dalam metode *convolutional neural network*. Tahap *feeding data training* disini adalah tahap *import* dataset citra training dan CSV *file*. Kemudian tahap *convolution* yang menentukan banyaknya jumlah *layer* dan semacamnya. Lalu tahap ReLu yang fungsinya hampir sama dengan fungsi *threshold* yang dimana jika 0 adalah negative dan 1 adalah positif. Kemudian tahap *pooling* yang tujuan dari penggunaan *pooling layer* adalah mengurangi dimensi dari *feature map* (*downsampling*), sehingga mempercepat komputasi karena parameter yang harus di *update* semakin sedikit dan mengatasi *overfitting*.

2.10. Skenario Ujicoba

Dalam Skenario Pengujian dibuat untuk mengimplementasikan hasil dari penelitian ini, yang nantinya akan digunakan dan diimplementasikan berdasarkan apa yang telah ditulis oleh peneliti. Perancangan skenario ini cukup penting karena dengan adanya skenario ini sebagai penunjang sebuah penelitian dan juga dengan adanya skenario ini maka percobaan akan lebih akurat dan efisien.

- Skenario pertama
Untuk percobaan pertama, peneliti akan mengumpulkan data pada situs google sebanyak 100 gambar dengan gambar ikan acak tanpa diberi nama *class*.
- Skenario kedua
Percobaan kedua, mengumpulkan data dengan mengklasifikasikan jenis-jenis ikan menjadi *class* lalu dilakukan pelatihan.
- Skenario ketiga
Percobaan ketiga, penambahan jumlah dataset menjadi 900 gambar (100 gambar tiap *class*) dan menambah CSV *file* untuk menambah keakuratan pengenalan ikan.
- Skenario keempat
Percobaan keempat, dilakukan uji coba sistem untuk mendeteksi ikan menggunakan

webcam dengan mengarahkan gambar ikan pada kamera.

3. HASIL DAN PEMBAHASAN

Dalam bagian anda, penulis harus menjelaskan/memaparkan/menunjukkan hasil dari penelitian atau hasil dari yang anda amati atau temukan ketika melakukan penelitian.

Data uji yang digunakan dalam program ini merupakan gambar ikan hias yang terdiri dari 9 bentuk. Data berupa hasil unduhan dari pencarian pada situs *Google*. Banyaknya data citra yang digunakan adalah 720 buah untuk proses pelatihan dan 180 buah untuk proses pengujian. Jadi total keseluruhan data yang digunakan sebanyak 900 buah citra.

3.1 Hasil Klasifikasi

Dari hasil yang sudah dilakukan, hanya ada kemungkinan 4 kasus yang terjadi:

1. True Positive (TP) : kasus dimana ikan dapat diprediksi dengan benar (positif), memang benar(True) terprediksi.
2. True Negative (TN) : kasus dimana ikan diprediksi tidak(Negatif) atau salah dan sebenarnya ikan tersebut memang (True) benar.
3. False Positve (FP) : kasus dimana ikan yang diprediksi positif atau benar, ternyata tidak atau salah. Prediksinya salah (False).
4. False Negatif (FN): kasus dimana ikan yang diprediksi salah(Negatif), tetapi ternyata sebenarnya benar (true).

Tabel4. 1. Tabel *Confussion Matrix*

		TRUE/ACTUAL								
		A	B	C	D	E	F	G	H	I
P										
R	A	3	0	0	0	0	0	0	0	0
E	B	0	0	1	0	1	0	0	1	0
D	C	0	0	3	0	0	0	0	0	0
I	D	0	0	0	3	0	0	0	0	0
C	E	0	0	0	0	3	0	0	0	0
T	F	0	0	0	0	0	2	0	0	1
	G	0	0	0	0	0	0	3	0	0
	H	0	0	0	0	0	0	0	3	0
	I	0	0	0	0	0	0	0	0	3

Keterangan:

A = Arwana
 B = Comet
 C = Guppy
 D = Koki
 E = Layar/Manfish
 F = Kura-Kura

G = Molly
H = Platys
I = Redfin

Maka dari kemungkinan-kemungkinan tersebut dapat dilakukan perhitungan untuk mendapatkan nilai akurasi, presisi, dan *recall*.

Tabel 4. 1. Tabel Perhitungan Confession Matrix

Akurasi	$= (TP + TN) / (TP + FP + FN + TN)$ $= (23 + 0) / (23 + 4 + 0 + 0)$ $= 23/27 = 85,14\%$
Presisi	$= (TP) / (TP + FP)$ $P(A) = 3/3 + 0 = 1 \quad P(B) = 0/0 + 0 = 0 \quad P(C) = 3/3 + 1 = 0,75$ $P(D) = 3/3 + 0 = 1 \quad P(E) = 3/3 + 1 = 0,75 \quad P(F) = 2/2 + 0 = 1$ $P(G) = 3/3 + 0 \quad P(H) = 3/3 + 1 = 0,75 \quad P(I) = 3/3 + 1 = 0,75$ <p>Seluruh Presisi</p> $= \frac{P(A) + P(B) + P(C) + P(D) + P(E) + P(F) + P(G) + P(H) + P(I)}{\text{Jumlah class}}$ <p>Presisi</p> $= \frac{1 + 0 + 0,75 + 1 + 0,75 + 1 + 1 + 0,75 + 0,75}{9} = 0,778 = 77,8\%$
Recall	$= (TP) / (TP + FN)$ $R(A) = 3/3 + 0 = 1 \quad R(B) = 0/0 + 3 = 0 \quad R(C) = 3/3 + 0 = 1$ $R(D) = 3/3 + 0 = 1 \quad R(E) = 3/3 + 0 = 1 \quad R(F) = 2/2 + 1 = 0,667$ $R(G) = 3/3 + 0 = 1 \quad R(H) = 3/3 + 0 = 1 \quad R(I) = 3/3 + 0 = 1$ $\text{Recall} = \frac{0 + 0 + 1 + 1 + 1 + 0,667 + 1 + 1 + 1}{9} = 0,852 = 85,2\%$

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan pengujian dan analisa hasil pengujian dari keseluruhan sistem yang telah dilakukan didapatkan kesimpulan sebagai berikut:

- Pengujian algoritma menghasilkan rata-rata pengujian sebesar 79,52% dari total 27 citra uji coba dengan menggunakan foto ikan.
- Proses *training* telah dilakukan 6 kali percobaan, *training* pertama mendapatkan nilai paling sempurna dengan jumlah *epoch* sebanyak 20, *steps per epoch* sebanyak 2000, jumlah *filter* sebanyak 60 dan ukuran *filter* sebesar 5x5 didapatkan *test score* sebesar 2.475 dan *test accuracy* sebesar 0.4237 dengan *loss* hanya sebesar 2.2002. Sedangkan percobaan yang mendapatkan nilai paling buruk ada pada percobaan *training* keempat dengan jumlah *epoch* hanya 1, *steps per epoch* sebanyak 1000 dan jumlah *filter* sebanyak 50 didapatkan *test score* sebesar 2.205 dan *test accuracy* sebesar 0.3389 dengan *loss* sebesar 2.0864.
- Akurasi dalam penelitian ini mendapatkan nilai sebesar 85,14% dari hasil 23 dapat memprediksi dengan benar dan 4 tidak dapat diprediksi dengan benar. Selanjutnya mendapatkan nilai presisi sebesar 77,8% dan nilai *recall* sebesar 85,2%.

4.2 Saran

Banyaknya kekurangan dalam penulisan laporan maupun sistem yang jauh dari kata sempurna. Tetapi telah dibuat secara maksimal dapat mencakup segala aspek dan hasil yang diharapkan. Berikut adalah saran dari peneliti dalam penelitian ini:

- Meskipun tingkat akurasi dari metode *convolutional neural network* ini rata-rata hanya mencapai 79,52% akan tetapi lebih baik pada penelitian selanjutnya untuk dirubah tahap *pre processing* dan jumlah data training agar mendapatkan hasil akurasi yang lebih tinggi.
- Untuk penelitian selanjutnya sebaiknya memberikan gambar citra yang lebih jelas dan lebih unik sehingga lebih mudah dikenali saat proses prediksi.
- Menggunakan kamera atau *webcam* external yang kualitasnya jauh lebih bagus dibandingkan penelitian ini yang menggunakan kamera atay *webcam* bawaan (*internal*).

5. DAFTAR RUJUKAN

- [1] Fadhilla, M., Adin S, M. R., & Sahid, D. S. (2017). Pengenalan Kepribadian Seseorang Berdasarkan Pola Tulisan Tangan Menggunakan Jaringan Saraf Tiruan. JNTETI ISSN 2301 - 4156, 6, 365-373.
- [2] Freeman, J. (1991). Neural Networks: Algorithms, Applications, and Programming Techniques (Computation and Neural Systems Series).
- [3] Goodfellow, I., Bengio, Y., & Courville, A. (t.thn.). Deep Learning (Adaptive Computation And Machine Learning Series). The MIT Press, 2016.
- [4] Hermawan, A. (2006). Jaringan Saraf Tiruan dan Aplikasinya. Yogyakarta: Andi.
- [5] I Wayan Suartika, Yudhi Wijaya dan Rully Soelaiman. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network Pada Caltech 101.
- [6] Muhammad Zufar & Budi Setiyono. (2016). Convolutional Neural Networks untuk Pengenalan Wajah Secara Real Time. 1.
- [7] Mukhlisatun Nada. (2019, Januari 11). PENERAPAN DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) DI R DAN TENSORFLOW UNTUK KLASIFIKASI GAMBAR KITAB DAN AL-QUR'AN. Indonesia.
- [8] Nelson, J. (2006). Fishes of the world. 4th Edition. John Wiley & Sons. New Jersey, USA.
- [9] Rahman, N. (2020, January 13). What is the benefit of using average pooling rather than max pooling. Diambil kembali dari quora: <https://www.quora.com/What-is-the-benefit-of-using-average-pooling-rather-than-max-pooling>.
- [10] Schuerer, Katja dan Corinne Maufrais. (2010). Introduction to Programming using. Boston: Pearson, 1-242.
- [11] Srivastava, Hinton, G, & Krizhevsky, A. (2014). Dropout: A Simple Way to Prevent Neural Networks from.
- [12] Sutojo, Mulyanto, & Suhartono. (2011). Kecerdasan Buatan. Yogyakarta: Penerbit Andi.
- [13] Suyanto. (2014). Artificial Intelligence. Bandung: Penerbit Informatika.
- [14] Syafi'i, S. (2011, Desember 15). Open Computer Vision(OpenCV). Diambil kembali dari <http://slametux.blogdetik.com/2011/10/19/open-computer-vision-opencv>
- [15] TRNOVSZKY, T. (2017). Animal Recognition System Based. 522.
- [16] Vivian, F. (2012). Taranis: Neural networks and intelligent agents in the early warning against floods, Expert Systems with Applications.
- [17] Woods, Richard E., Gonzales, Rafael E. (2008). Digital Image Processing Third Edition. New Jersey, USA: Pearson Education.
- [18] Danukusumo, K. 2017. Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Citra Candi Berbasis GPU. Tugas Akhir.

- [19] Hamida, U. 2014. PENGGUNAAN ARTIFICIAL NEURAL NETWORK (ANN) UNTUK MEMODELKAN KEBUTUHAN ENERGI UNTUK TRANSPORTASI. Jurnal Teknologi Manajemen, Vol. 12 , No.2.
- [20] Zufar, M. & Setiyono B. 2016. Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time. JURNAL SAINS DAN SENI ITS Vol. 5 No. 2 . A-72.
- [21] Puri, F.T. 2011. Analisis algoritma eigenface (Pengenalan wajah) pada aplikasi kehadiran pengajar dosen.
- [22] Sena, Samuel. 2018. Pengenalan Deep Learning Part 7: Convolutional Neural Network(CNN). <https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94> . Diakses pada 13 Januari 2020
- [23] Sofia, Nadhifa. 2018. Convolutional Neural Network. <https://medium.com/@nadhifasofia/1-convolutional-neural-network-convolutional-neural-network-merupakan-salah-satu-metode-machine-28189e17335b>. Diakses pada 13 Januari 2020.
- [24] Novyantika, Rizky Dwi. 2018. Deteksi Tanda Nomor Kendaraan Bermotor Pada Media Streaming Dengan Algoritma Convolutional Neural Network Menggunakan Tensorflow. Tugas Akhir.
- [25] Sharma, Sagar. 2017. Activation Functions : Neural Networks. <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>. Diakses pada 8 Februari 2020
- [26] Pudi, Kefin. 2017. Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Citra Candi Berbasis GPU. Tugas Akhir. Program Studi Teknologi Industri Universitas Atma Jaya Yogyakarta.