

# IDENTIFIKASI JENIS IKAN MENGGUNAKAN MODEL HYBRID DEEP LEARNING DAN ALGORITMA KLASIFIKASI

Anifuddin Azis

Departemen Ilmu Komputer dan Elektronika, Fakultas MIPA, Universitas Gadjah Mada  
Sekip Utara, Bulaksumur, Yogyakarta  
E-mail : anifudin@ugm.ac.id

## ABSTRAK

Indonesia merupakan negara dengan keanekaragaman hayati terbesar kedua di dunia setelah Brazil. Indonesia memiliki sekitar 25.000 spesies tumbuhan dan 400.000 jenis hewan dan ikan. Diperkirakan 8.500 spesies ikan hidup di perairan Indonesia atau merupakan 45% dari jumlah spesies yang ada di dunia, dengan sekitar 7.000an adalah spesies ikan laut. Untuk menentukan berapa jumlah spesies tersebut dibutuhkan suatu keahlian di bidang taksonomi. Dalam pelaksanaannya mengidentifikasi suatu jenis ikan bukanlah hal yang mudah karena memerlukan suatu metode dan peralatan tertentu, juga pustaka mengenai taksonomi. Pemrosesan video atau citra pada data ekosistem perairan yang dilakukan secara otomatis mulai dikembangkan. Dalam pengembangannya, proses deteksi dan identifikasi spesies ikan menjadi suatu tantangan dibandingkan dengan deteksi dan identifikasi pada objek yang lain. Metode *deep learning* yang berhasil dalam melakukan klasifikasi objek pada citra mampu untuk menganalisis data secara langsung tanpa adanya ekstraksi fitur pada data secara khusus. Sistem tersebut memiliki parameter atau bobot yang berfungsi sebagai ekstraksi fitur maupun sebagai pengklasifikasi. Data yang diproses menghasilkan *output* yang diharapkan semirip mungkin dengan data *output* yang sesungguhnya. CNN merupakan arsitektur *deep learning* yang mampu mereduksi dimensi pada data tanpa menghilangkan ciri atau fitur pada data tersebut. Pada penelitian ini akan dikembangkan model *hybrid CNN (Convolutional Neural Networks)* untuk mengekstraksi fitur dan beberapa algoritma klasifikasi untuk mengidentifikasi spesies ikan. Algoritma klasifikasi yang digunakan pada penelitian ini adalah : *Logistic Regression (LR)*, *Support Vector Machine (SVM)*, *Decision Tree*, *K-Nearest Neighbor (KNN)*, *Random Forest*, *Backpropagation*.

**Kata Kunci:** Identifikasi Ikan, Deep Learning, CNN, Algoritma Klasifikasi, Model Hybrid

## 1. PENDAHULUAN

Indonesia merupakan negara dengan keanekaragaman hayati terbesar kedua di dunia setelah Brazil. Indonesia memiliki sekitar 25.000 spesies tumbuhan dan 400.000 jenis hewan dan ikan. Diperkirakan 8.500 spesies ikan hidup di perairan Indonesia atau merupakan 45% dari jumlah spesies yang ada di dunia, dengan sekitar 7.000an adalah spesies ikan laut. (Keputusan Kepala Badan Karantina Ikan Pengendalian Mutu dan Keamanan Hasil Perikanan No 67/Kep-BKIPM/2015).

Untuk menentukan berapa jumlah spesies tersebut dibutuhkan suatu keahlian di bidang taksonomi. Salah satu bagian penting dari taksonomi adalah teknik identifikasi. Dalam pelaksanaannya mengidentifikasi suatu jenis ikan bukanlah hal yang mudah karena memerlukan suatu metode dan peralatan tertentu, juga pustaka mengenai taksonomi (Haryono, 2017).

Pemrosesan video atau citra pada data ekosistem perairan yang dilakukan secara otomatis mulai dikembangkan. Dalam pengembangannya, proses deteksi dan identifikasi spesies ikan menjadi suatu tantangan dibandingkan dengan deteksi dan identifikasi pada objek yang lain (Zheng dkk., 2018). Citra dengan objek ikan menjadi sulit untuk dideteksi karena pada umumnya kontras cahaya atau kualitas citra yang

kurang baik pada kondisi perairan tertentu, objek ikan yang hanya mencakup sebagian kecil dari keseluruhan luas citra, serta keberadaan objek lain yang secara alami menyatu dengan objek ikan (Socher dkk., 2012). Sehingga berkembang metode pengolahan data sebelum dilakukannya deteksi dan identifikasi spesies ikan untuk mengambil ciri atau fitur dari objek ikan pada citra tersebut seperti tekstur, bentuk, dan warna. Proses tersebut bergantung pada metode bagaimana melakukan ekstraksi fitur sehingga perlu wawasan yang dalam terhadap setiap kemungkinan citra pada data yang telah diambil.

Pengenalan ikan secara otomatis dapat dibagi ke dalam dua komponen: (1) deteksi ikan otomatis pada rangkaian video dan (2) klasifikasi ikan otomatis pada suatu frame video. (Salman dkk, 2016).

Penggunaan metode *deep learning* sudah banyak dilakukan untuk mengidentifikasi / klasifikasi jenis ikan. Villon dkk (2018) menggunakan CNN untuk mengidentifikasi 20 spesies ikan dengan tingkat akurasi 94%. Cui, dkk (2020) juga menggunakan *I* untuk mendeteksi ikan menggunakan CNN (*Convolution Neural Networks*) dengan beberapa metode optimasi pada CNN. Sedangkan, Tamou, dkk (2018)

menggunakan CNN untuk klasifikasi ikan dengan 202 membandingkan beberapa fungsi aktivasi, yaitu: ReLU, tanh, dan SoftMax. Dan hasilnya fungsi aktivasi ReLU menghasilkan tingkat akurasi paling baik.

Sementara menggabungkan teknik pengolahan citra dan CNN pada dataset The Fish Recognition Ground-Truth dari Fish4Knowledge (27142 citra, 21 kelas) dengan tingkat akurasi 96,26 % (Rathi,dkk, 2017).

CNN dan algoritma klasifikasi KNN pada dataset The Fish Recognition Ground-Truth dari Fish4Knowledge (27370 citra, 23 kelas) dengan hasil akurasi 98,79% (Deep dan Dash, 2019).

Penelitian ini merupakan pengembangan dari penelitian yang menggunakan CNN dan algoritma klasifikasi KNN, yaitu dengan menggunakan beberapa algoritma klasifikasi yang lain : *Logistic Regression (LR)*, *Support Vector Machine (SVM)*, *Decision Tree*, *K-Nearest Neighbor (KNN)*, *Random Forest*, *Backpropagation*. Penggunaan beberapa algoritma klasifikasi ini untuk mengetahui algoritma klasifikasi yang memiliki akurasi yang paling baik dalam mengidentifikasi jenis ikan (Deep dan Dash, 2019).

## 2. RUANG LINGKUP

Dalam penelitian ini permasalahan mencakup:

1. Bagaimana model arsitektur CNN yang terbaik dalam mengekstraksi fitur citra ikan.
2. Algoritma klasifikasi yang memiliki akurasi terbaik.

## 3. BAHAN DAN METODE

Data yang digunakan pada penelitian ini adalah data citra yang bersumber dari : <http://groups.inf.ed.ac.uk/f4k/GROUNDTRUTH/RECOG/> Nama spesies ikan yang digunakan pada eksperimen ini seperti pada Tabel 1.

**Tabel 1. Nama Spesies Ikan**

Kelas	Nama spesies
0	<i>Plectroglyhidodon dickii</i>
1	<i>Chaetodon lunulatus</i>
2	<i>Chaetodon trifascialis</i>
3	<i>Myripristis kuntee</i>
4	<i>Acanthurus nigrofuscus</i>
5	<i>Hemigymnus fasciatus</i>
6	<i>Neoniphon sammara</i>
7	<i>Abudefduf vaigiensis</i>
8	<i>Canthigaster valentini</i>
9	<i>Pomacentrus moluccensis</i>
10	<i>Zebrasoma scopas</i>
11	<i>Hemigymnus melapterus</i>

12	<i>Lutjanus fulvus</i>
13	<i>Scolopsis bilineata</i>
14	<i>Scaridae</i>
15	<i>Pempheris vanicolensis</i>
16	<i>Zanclus cornutus</i>
17	<i>Neoglyphidodon nigroris</i>
18	<i>Balistapus undulatus</i>
19	<i>Siganus fuscescens</i>

Dataset yang telah diperoleh dibagi menjadi data latih (*data train*) dan data uji (*data test*). Banyak data latih adalah : 6.108 dan banyak data uji: 1.518.

Seluruh citra diubah ukurannya terlebih dahulu sebelum proses pelatihan menjadi 96x96 piksel. Alasan dipilihnya ukuran tersebut adalah karena rata-rata dimensi panjang dan lebar dari seluruh citra adalah 96 piksel

### 3.1 Convolutional Layer

*Convolutional Neural Network (CNN)* merupakan arsitektur *deep learning* yang mampu mereduksi dimensi pada data tanpa menghilangkan ciri atau fitur pada data tersebut. Beberapa jenis layer yang digunakan pada arsitektur tersebut adalah *convolutional layer*, *pooling layer*, dan *fully connected layer*. *Convolutional layer* melakukan operasi berupa *convolution* pada data untuk mengekstraksi fitur atau ciri. Pada klasifikasi sebuah citra, data merupakan sebuah matriks yang memiliki 3 dimensi, di mana dimensi ketiga disebut sebagai *depth*. Untuk melakukan ekstraksi pada satu fitur, digunakan sebuah *filter* yang merupakan matriks 3 dimensi dengan *depth* yang sama namun memiliki ukuran yang lebih kecil dibandingkan dengan data yang diekstraksi pada dimensi yang lain. Nilai yang terdapat pada *filter* merupakan parameter yang dapat dilatih (Dertat, 2017).

Proses *convolution* pada matriks  $\mathbf{M}$  dengan ukuran  $(W_M, h_M, d)$  menggunakan *filter* dengan  $\mathbf{K}$  dengan ukuran  $(W_K, h_K, d)$  dilakukan dengan mengoperasikan setiap sampling submatriks  $\mathbf{V}_{ij}$  yang berukuran sama dengan  $\mathbf{K}$  pada  $\mathbf{M}$  dengan  $\mathbf{K}$  menggunakan dot *product* pada (1).

$$x_{i,j} = V_{i,j} \cdot K = \sum_{a=0}^{WK} \sum_{b=0}^{hK} \sum_{c=0}^d K_{a,b,c} M_{i+a,j+b,c} \quad (1)$$

Setiap nilai  $x_{i,j}$  merupakan bilangan real yang memiliki rentang nilai  $(-\infty, \infty)$  sehingga dilakukan proses aktivasi menggunakan fungsi aktivasi ReLU pada (2).

$$f(x) = \max(0, x) \quad (2)$$

Kemudian setiap nilai  $\mathbf{X}_{ij}$  digabungkan untuk membentuk matriks baru yang disebut sebagai *feature map* dengan ukuran  $(W_M - W_K + 1, h_M - h_K + 1)$ . Pada umumnya banyaknya *filter* yang digunakan ada lebih dari satu sesuai dengan banyak fitur yang ingin

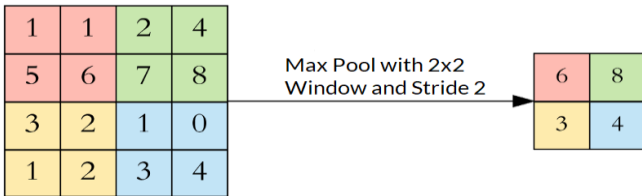
diekstraksi. Setiap *feature map* yang dihasilkan pada setiap *filter* ditumpuk atau digabungkan sehingga membentuk matriks 3 dimensi yang baru. Apabila terdapat sebanyak  $d'$  *filter*, maka *feature map* berukuran  $(W_M - W_K + 1, h_M - h_K + 1, d')$ .

**3.2 Pooling Layer**

*Pooling layer* melakukan operasi berupa *pooling* pada data untuk mereduksi dimensi serta mengurangi banyaknya parameter yang digunakan. Proses *pooling* pada *feature map*  $M$  yang merupakan matriks 3 dimensi berukuran  $(w_M, h_M, d)$  menggunakan ukuran *pooling* sebesar  $(w_p, h_p)$  dilakukan dengan membagi matriks pada setiap fitur menjadi  $\frac{W_M}{W_p} \times \frac{h_M}{h_p}$  sub matriks dengan masing-masing submatriks berukuran sama dengan ukuran *pooling* pada (3).

$$x_{i,j,k} = pool_{\substack{iw_p \leq a \leq (i+1)w_p \\ jh_p \leq b \leq (j+1)h_p}}(M_{a,b,k}) \quad (3)$$

Di mana fungsi  $pool(\cdot)$  yang digunakan pada umumnya adalah fungsi maksimum (*maximum pooling*) dan fungsi rata-rata (*average pooling*). Kemudian untuk  $k \in [1, d]$  setiap nilai  $x_{i,j,k}$  digabungkan sehingga akan diperoleh matriks *feature map* yang baru berukuran  $(\frac{w_M}{w_p}, \frac{h_M}{h_p}, d)$ . Proses *pooling* menggunakan *max pooling* digambarkan pada Gambar 1.



**Gambar 1. Proses Pooling Menggunakan Max Pooling (Dertat, 2017)**

**3.3 Fully Connected Layer**

merupakan *layer* yang digunakan untuk klasifikasi setelah proses ekstraksi fitur. *Feature map* yang memiliki 3 dimensi berukuran  $(w, h, d)$  diubah ke 1 dimensi menjadi sebuah vector dengan panjang  $w \times h \times d$  untuk digunakan sebagai *input* pada model jaringan syaraf untuk proses klasifikasi.

**3.4 Evaluasi**

Untuk mengetahui arsitektur CNN dan model klasifikasi yang terbaik, diperlukan suatu ukuran evaluasi sebagai pembanding.

1. Arsitektur CNN

Arsitektur CNN yang dipilih adalah arsitektur yang paling stabil konvergensi nilai *loss*-nya dan yang memberikan akurasi validasi tertinggi. Konvergensi nilai *loss* dikatakan stabil jika untuk sejumlah epoch terakhir, tren nilai *loss*-nya tidak menaik. Bisa saja nilai *loss* tersebut fluktuatif/naik-turun. Jika hal ini terjadi pada lebih dari satu arsitektur, arsitektur yang dipilih adalah yang lebih rendah naik-turunnya.

2. Model Klasifikasi

Untuk menentukan model klasifikasi mana yang memberikan hasil terbaik, model tersebut diukur dengan menggunakan ukuran evaluasi yang diukur pada data uji. Salah satu ukuran evaluasi yang dapat digunakan yaitu akurasi, yang dihitung dengan (4).

$$akurasi = \frac{\text{banyak data yang diprediksi dengan benar}}{\text{banyak data seluruhnya}} \quad (4)$$

Namun, data yang dimiliki bersifat *imbalance* atau tidak seimbang pada distribusi kelasnya. Ketidak seimbangan ini dapat menyebabkan nilai akurasi yang tinggi karena dengan memprediksi suatu data sebagai kelas yang dominan, akan meningkatkan nilai akurasi dengan mudah, sehingga tidak merepresentasikan kemampuan model untuk melakukan klasifikasi untuk seluruh kelas. Untuk itu diperlukan ukuran evaluasi yang sesuai untuk data yang bersifat *imbalance*, antara lain *precision*, *recall*, dan F1.

A. Precision

Ukuran ini dihitung untuk setiap kelas. *Precision* dari model  $M$  untuk kelas  $c$  menggambarkan dari seluruh data dengan kelas  $c$ , berapa banyak yang berhasil diprediksi oleh model  $M$ . *Precision* dari model  $M$  untuk kelas  $c$  dihitung dengan (5).

$$precision = \frac{TP}{TP + FP} \quad (5)$$

Dengan :

- a. TP menyatakan banyak data kelas  $c$  yang diprediksi sebagai kelas  $c$  oleh model  $M$ .
- b. FP menyatakan banyak data kelas  $c$  yang diprediksi sebagai **bukan** kelas  $c$  oleh model  $M$ .
- c. Nilai *precision* berkisar dari 0 sampai 1. Semakin tinggi nilai *precision* dari suatu model, semakin baik model tersebut.

B. Recall

Ukuran ini juga dihitung untuk setiap kelas. *Recall* dari model  $M$  untuk kelas  $c$  menggambarkan dari seluruh data yang diprediksi sebagai kelas  $c$  oleh model  $M$ , berapa banyak yang benar-benar merupakan kelas  $c$ . *Recall* dari model  $M$  untuk kelas  $c$  dihitung dengan (6).

$$recall = \frac{TP}{TP + FN} \quad (6)$$

dengan FN menyatakan banyak data yang bukan kelas  $c$ , yang diprediksi sebagai kelas  $c$  oleh model  $M$ . Nilai  $recall$  berkisar dari 0 sampai 1. Semakin tinggi nilai  $recall$  dari suatu model, semakin baik model tersebut.

#### C. F1

Ukuran ini juga dihitung untuk setiap kelas. Ukuran ini menghitung nilai rata-rata harmonik dari  $precision$  dan  $recall$ . Nilai F1 dari model  $M$  untuk kelas  $c$  dihitung dengan (7)

$$F1_c = \frac{precision_c \times recall_c}{precision_c + recall_c} \quad (7)$$

Nilai F1 berkisar dari 0 sampai 1. Semakin tinggi nilai F1 dari suatu model, semakin baik model tersebut. Ukuran F1 dapat dianggap seperti "akurasi" yang lebih representatif untuk data bersifat *imbalanced*.

### 3.5 Algoritma Klasifikasi

Terdapat enam algoritma berbeda yang akan dibandingkan pada penelitian ini, yaitu :

1. Regresi Logistik (LR)  
LR bekerja mirip dengan regresi linear yang menghasilkan sejumlah bobot sebanyak fitur yang ada pada data. Pencarian bobot ini dilakukan secara iteratif. Pada eksperimen ini dibatasi banyak iterasi yang digunakan, yaitu 1.000 iterasi.
2. *Support Vector Machine* (SVM)  
SVM bekerja dengan cara memetakan fitur yang ada pada data ke dimensi lain, kemudian mencari pembatas/*hyperplane* sehingga setiap kelas terbedakan dengan baik. Ukuran baik atau tidaknya pembedaan tersebut diukur dengan *hyperplane* yang memberikan margin yang optimal. Proses pencarian *hyperplane* ini dilakukan dengan metode optimasi secara iteratif. Namun banyak iterasi untuk SVM tidak dibatasi.
3. *K-Nearest Neighbors* (KNN)  
KNN bekerja dengan cara mencari  $K$  data terdekat dari suatu data. Kelas dari suatu data adalah kelas yang terbanyak dari  $K$  data terdekat dengannya. Jarak dari dua data dihitung dengan *euclidean distance*. Setiap  $K$  data terdekat tersebut juga dapat diberi bobot sesuai dengan inversi dari jaraknya, sehingga data yang lebih dekat lebih mempengaruhi untuk penentuan kelas dari data tersebut.
4. *Decision Tree* (DT)  
DT bekerja dengan mencari aturan yang dapat diperoleh dari data. Hasil dari algoritma DT berupa pohon aturan yang akan menentukan kelas dari suatu sampel data. Penentuan pohon aturan ini diperoleh dengan mengolah *entropy* yang ada pada data.
5. *Random Forest* (RF)

204

RF bekerja dengan teknik *ensemble*, yaitu dengan menggabungkan beberapa *decision tree*.

#### 6. *Fully Connected Network* (NN)

Arsitektur NN yang digunakan adalah arsitektur NN yang diperoleh saat pencarian arsitektur CNN untuk ekstraksi ciri sebelumnya, yaitu menggunakan algoritma *Backpropagation*.

### 3.6 Tahapan Penelitian

Tahapan penelitian yang dilakukan adalah :

1. Pemilihan arsitektur dilakukan dengan membandingkan sejumlah model berbeda. Antara model satu dan lainnya yang membedakan bisa dari parameter arsitektur atau parameter pelatihan yang digunakan. Parameter pelatihan yang dimaksud antara lain adalah metode optimasi yang digunakan (Adam atau *Stochastic Gradient Descent* -- SGD), *learning rate* dan *momentum*, dan *batch size*. Implementasi dilakukan dengan menggunakan bahasa pemrograman Python dan pustaka TensorFlow.
2. Setelah diperoleh arsitektur CNN terbaik, selanjutnya dilakukan penentuan model klasifikasi yang memberikan ukuran evaluasi terbaik. Arsitektur CNN mengubah citra menjadi vektor ciri satu dimensi yang kemudian akan diklasifikasi oleh model klasifikasi. Implementasi dilakukan dengan bahasa pemrograman Python dan pustaka Scikit Learn.

## 4. PEMBAHASAN

Langkah pertama pada penelitian ini adalah menentukan arsitektur dan parameter pelatihan yang terbaik pada CNN dalam mengekstraksi fitur. Dari hasil pengujian, diperoleh arsitektur CNN yang terbaik adalah terdiri dari 4 blok, yaitu :

1. Blok 1 terdiri dari *Convolutional layer* 8 filter ukuran  $7 \times 7$ , *pooling layer*, menggunakan *batch normalization* dan fungsi ReLU.
2. Blok 2 dan blok 3 terdiri dari *Convolutional layer* 16 filter ukuran  $5 \times 5$ , *pooling layer*, menggunakan *batch normalization* dan fungsi ReLU.
3. Blok 4 terdiri dari *Convolutional layer* 32 filter ukuran  $3 \times 3$ , *pooling layer*, menggunakan *batch normalization* dan fungsi ReLU.

Dan parameter pelatihan terbaiknya adalah : metode optimasi : Adam, *learning rate* : 0,001, momentum beta 1 : 0,9, momentum beta 2 : 0,999, *batch size* : 64, dan *epoch* : 40.

Dengan arsitektur dan parameter pelatihan ini, CNN mencapai akurasi data latih sebesar 100% dan akurasi validasi sebesar 94,2%. Dan dengan arsitektur ini, citra berukuran  $96 \times 96$  piksel diekstraksi cirinya menjadi vektor satu dimensi dengan ukuran 288.

Setelah didapatkan ekstraksi ciri, kemudian diujikan menggunakan beberapa algoritma klasifikasi, yaitu : *Logistic Regression* (LR), *Support Vector Machine* (SVM), *Decision Tree* (DT), *K-Nearest Neighbor* (KNN), *Random Forest*, *Backpropagation*.

Dari pengujian tingkat akurasi, diperoleh hasil tingkat akurasi tertinggi menggunakan algoritma Backpropagation dan cukup berimbang dengan LR, SVM, dan KNN dan terendah *Decision Tree*, seperti pada Tabel 2.

**Tabel 2. Tingkat Akurasi Pada Data Uji**

Nama Algoritma	akurasi (%)
LR	93,35
SVM	93,48
DT	74,77
Backpropagation	94,2
KNN	92,09
RF	89,86

Dari pengujian *precision*, diperoleh hasil tingkat presisi tertinggi menggunakan SVM dan berimbang *Backpropagation* dan terendah *Decision Tree*, seperti pada Tabel 3.

**Tabel 3. Tingkat Precision Pada Data Uji**

Nama Algoritma	presisi (%)
LR	83,03
SVM	90,81
DT	39,88
Backpropagation	88,31
KNN	83,04
RF	77,41

Dari pengujian tingkat *recall*, diperoleh hasil tingkat *recall* tertinggi menggunakan algoritma *Backpropagation* dan cukup berimbang dengan LR, SVM, dan KNN dan terendah *Decision Tree*, seperti pada Tabel 4.

**Tabel 4. Tingkat Recall Pada Data Uji**

Nama Algoritma	recall (%)
LR	93,35
SVM	93,48
DT	74,77
Backpropagation	94,2
KNN	92,09
RF	89,86

Dan dari pengujian F1, diperoleh hasil tingkat presisi tertinggi algoritma *Backpropagation* dan cukup berimbang dengan LR, SVM, dan KNN dan terendah *Decision Tree*, seperti pada Tabel 5.

**Tabel 5. Tingkat Akurasi Pada Data Uji**

Nama Algoritma	F1 (%)
LR	76,98
SVM	76,79
DT	39,22
Backpropagation	77,84
KNN	76,04
RF	62,5

## 5. KESIMPULAN

Dari pengujian yang dilakukan pada penelitian ini dapat disimpulkan bahwa arsitektur CNN yang digunakan pada penelitian ini adalah terdiri dari 4 blok, diantaranya Blok 1 terdiri dari Convolutional layer 8 filter ukuran 7x7, pooling layer, menggunakan batch normalization dan fungsi ReLU, blok 2 dan blok 3 terdiri dari Convolutional layer 16 filter ukuran 5x5, pooling layer, menggunakan batch normalization dan fungsi ReLU, dan blok 4 terdiri dari Convolutional layer 32 filter ukuran 3x3, pooling layer, menggunakan batch normalization dan fungsi ReLU.

1. Dan parameter pelatihan terbaiknya adalah : metode optimasi : Adam, learning rate : 0,001, momentum beta 1 : 0,9, momentum beta 2 : 0,999, batch size : 64, dan epoch : 40.

Berdasarkan pengujian pada beberapa algoritma klasifikasi, dapat disimpulkan algoritma yang mendapatkan hasil yang baik pada penelitian ini adalah : Backpropagation, SVM, Logistic Regression, dan KNN

## 6. SARAN

Untuk penelitian berikutnya akan digunakan algoritma klasifikasi yang lain, juga algoritma *deep learning* lain seperti RNN. Untuk implementasi bisa digunakan menggunakan data citra ikan yang diambil langsung dari perairan Indonesia.

## 7. DAFTAR PUSTAKA

- Cui, S., Zhou, Y., Wang, Y., & Zhai, L. 2020 Article 373810
- Deep, B. V. and Dash, R. 2019. Underwater Fish Species Recognition Using Deep Learning Techniques. *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, pp. 665-669, doi: 10.1109/SPIN.2019.8711657.
- Dertat, A. 2017. Applied Deep Learning - Part 4: Convolutional Neural Networks. <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>.
- Haryono, H. 2017. Fauna Ikan Air Tawar Di Perairan Kawasan Gunung Sawal, Jawa Barat, Indonesia. *Berita Biologi*, 16(2), 147-156.
- Keputusan Kepala Badan Karantina Ikan Pengendalian Mutu dan Keamanan Hasil Perikanan No 67/Kep-BKIPM. 2015
- Rathi, D., Jain, S., Indu, S. 2017. Underwater Fish Species Classification using Convolutional Neural Network and Deep Learning. Ninth International Conference on Advances in Pattern Recognition
- Salman, A., Jalal, A., Shafait, F., Mian, A., Shortis, M., Seager, J., Harvey, E. 2016., *Limnology and Oceanography: Methods*, 14 570-585
- Socher, R., Huval, B., Bath, B. P., Manning, C. D., and Ng A. Y. 2012. Convolutional-recursive *deep*

- learning* for 3d object classification. *NIPS*, vol. 3, no.2067, p. 8.
- Tamou, A. B., Benzinou, A., Nasreddine, K. and Ballihi, L. 2018. Transfer Learning with deep Convolutional Neural Network for Underwater Live Fish Recognition. *2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS)*, Sophia Antipolis, France, pp. 204-209, doi: 10.1109/IPAS.2018.8708871.
- Villon, S., Mouillot, D., Chaumont, M., Darling, E. 2018. A Deep learning method for accurate and fast identification of coral fishes in underwater images, *Ecological Informatics, Elsevier*, pp238-244)
- Zheng, Z., Guo, C., Zheng, X., Yu, Z., Wang, W., Zheng, H., Fu, M., and Zheng, B. 2018. Fish Recognition from a Vessel Camera Using Deep Convolutional Neural Network and Data Augmentation. *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, Kobe, pp. 1-5, doi: 10.1109/OCEANSKOB.2018.8559314.

#### **UCAPAN TERIMA KASIH**

Terima kasih saya ucapkan kepada Departemen Ilmu Komputer dan Elektronika Fakultas MIPA UGM atas bantuan dana pada penelitian ini.