

PENGENALAN SPESIES IKAN BERDASARKAN KONTUR *OTOLITH* MENGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*

Heri Darmanto

AMIK Taruna Probolinggo

heridarmanto@amik-taruna.ac.id

Diterima: April 2019. Disetujui: Mei 2019. Dipublikasikan: Juni 2019

ABSTRAK

Hasil sensus kehidupan laut pada tahun 2013 di seluruh dunia terdapat lebih dari 23.000 spesies dan masih banyak sekali spesies ikan yang belum diidentifikasi. Otolith merupakan organ yang sangat penting di belakang telinga ikan, karena melalui otolith ini dapat diketahui jenis ikan, pertumbuhan dan lingkungan, serta sejarah kehidupannya, misalnya, umur, reproduksi, dan migrasi. Dengan semakin canggihnya komputer dan pengolahan di bidang citra, diharapkan kemampuan mengidentifikasi jenis ikan yang dimiliki oleh manusia bisa diadopsi dan diterapkan pada perangkat komputer. *Deep Learning* saat ini semakin berkembang memanfaatkan sumber daya perangkat keras yang semakin canggih termasuk penggunaan GPU (*Graphical Processing Unit*) untuk perhitungan proses komputasi dengan akurasi yang lebih baik dan proses yang lebih cepat. Pada penelitian ini metode yang diusulkan, untuk keperluan klasifikasi ikan menggunakan metode *Convolutional Neural Network* dengan teknik *Transfer Learning* dari model *Alexnet* dan optimasi *Momentum Stochastic Gradient Descent*. Hasil eksperimen diperoleh akurasi sebesar 95.4% lebih tinggi dibanding metode *Discriminant Analysis* yang memiliki akurasi sebesar 92%.

Kata Kunci: Ikan, *Otolith*, Klasifikasi, CNN, *Deep Learning*, *Transfer Learning*, *Gradient Descent*

ABSTRACT

Census results of marine life in 2013 around the world there are more than 23,000 species and still many species of fish that have not been identified. Otolith is a very important organ behind the ears of fish, because through this otolith can be known species of fish, growth and environment, and history of life, for example, age, reproduction, and migration. With the increasingly sophisticated computer and processing in the field of image, it is hoped that the ability to identify fish species owned by humans can be adopted and applied to computer devices. Deep Learning is increasingly expanding utilizing increasingly sophisticated hardware resources including the use of GPU (Graphical Processing Unit) for computing process calculations with better accuracy and faster processing. In this study the proposed method, for the purpose of classification of fish using the method of Convolutional Neural Network with Transfer Learning technique from Alexnet model and the optimization of Stochastic Gradient Descent Momentum. The experimental results obtained an accuracy of 95.4% higher than the Discriminant Analysis method that has an accuracy of 92%.

Keyword : Fish, *Otolith*, CNN, Classification, *Deep Learnig*, *Transfer Learning*, *Gradient Descent*

PENDAHULUAN

Menurut sensus kehidupan laut tahun 2013, di seluruh dunia terdapat lebih dari 23.000 spesies, yang mana 12% diantaranya merupakan jenis spesies ikan (2.760 – 3.000 spesies). Dari jumlah tersebut masih 1.200 spesies yang telah diberi

diskripsi secara formal (Alexander, Miloslavich, & Yarincik, 2011). Dibutuhkan ahli di bidang biota laut untuk bisa melakukan identifikasi, disertai suatu alat bantu untuk mengenali jenis-jenis spesies ikan tersebut.

Pengenalan ikan adalah cara mengidentifikasikan ikan berdasarkan ciri-ciri khusus, bisa melalui gambaran bentuk, pola

tubuh ikan, warna ataupun ciri-ciri lainnya. Manusia mempunyai kemampuan yang handal dalam melakukan pengenalan tersebut, tetapi sayangnya manusia memiliki keterbatasan seperti kelelahan, daya tahan untuk bekerja dalam waktu yang lama. Dengan semakin berkembang dan meluasnya kemampuan komputer, kemampuan mengidentifikasi ikan oleh manusia bisa ditingkatkan dengan bantuan perangkat komputer. Terjadi peningkatan yang signifikan dalam studi sistem identifikasi di bidang biologi berdasarkan morfologi (penampakan bentuk) maupun taksonomi (persamaan dan perbedaan sifatnya) secara otomatis, terutama disebabkan oleh peningkatan kemampuan pemrosesan dan pengolahan menggunakan perangkat komputer (Reichenbacher, Sienknecht, Kuchenhoff, & Fenske, 2007).



Gambar 0.1 Ilustrasi Letak Otolith Ikan (“The Secret Lives of Fish: Cohen Lab,” n.d.)

Lombarte dan Castellon (1991), mengamati bahwa setiap spesies ikan memiliki otolith dengan karakteristik tertentu (bentuk dan ukuran), dimana morfologi otolith dan morfometri telah dipelajari untuk mengidentifikasi spesies dan fauna fosil (Lombarte & Castellón, 1991), serta menentukan jenis makanan spesies dari kandungan isi perutnya (Fitch & Brownell, 1968) (Parisi-Baradad et al., 2010). Lebih penting lagi adalah kenyataan bahwa otolith bertindak sebagai data logger alami, yang berisi rekaman informasi dari ikan hidup terkait dengan pertumbuhan dan lingkungan mereka (Kalish, 1991) (Campana, 1999). Informasi ini, meliputi usia dan pertumbuhan, serta pola pergerakan dan interaksi habitat, yang

dapat digunakan untuk menjelaskan ekologi, demografi atau sejarah kehidupan, dan telah menjadi kepentingan mendasar dalam manajemen ikan dan perlindungan spesies (Soria & Baradad, 2012).



Gambar 0.2 Foto Otolith Ikan (Thomas, Ganio, Roberts, & Swearer, 2017)



Gambar 0.3 Berbagai Macam Ukuran Otolith Ikan (VanderKooy, 2009)

Otolith adalah sekumpulan *calcium carbonate* yang terletak di tulang dalam dari telinga ikan. Otolith ini menyimpan semua histori kehidupan dan lingkungan yang merekonstruksi parameter lingkungan seperti temperatur dan salinitas (kadar garam). Otolith merupakan organ yang sangat penting, karena melalui otolith ini dapat diketahui jenis ikan, pertumbuhan dan lingkungan, serta sejarah kehidupannya, misalnya umur, reproduksi, dan bahkan juga proses migrasinya (Roumillat, 2004). Penelitian tentang otolith merupakan topik yang menarik, terbukti sudah dilakukan penelitian sejak tahun 1968 oleh John E. Fitch dan Robert L. Brownell dari USA (Fitch & Brownell, 1968), hingga tahun 2016 oleh Nima Salimi dari Malaysia (Salimi, Loh, Kaur Dhillon, & Chong, 2016) masih melakukan penelitian tentang otolith ini.

Penelitian identifikasi ikan berdasar kontur otolith dilakukan oleh Nima Salimi (2016)

dengan menggunakan *Short Time Fourier Transform* dan *Discriminant Analysis*. Pembedaan antar spesies ikan berdasarkan garis tepi luar otolith. Kontur dari otolith diekstrak dan dihitung jarak antara pusat masa dengan garis tepinya. Citra grayscale dari otolith dikonversi ke citra biner dengan nilai threshold 0,1. Proses ekstraksi fitur menggunakan metode *Short Time Fourier Transform* pada pengambilan sample sinyal yang ditentukan dengan fungsi *Gaussian Window*. Hasil uji coba terbaik diperoleh dari 100 titik dan 40 sample overlap diperoleh masing-masing 16 segmen dari nilai jarak dan 16 segmen nilai sudut sehingga diperoleh total 32 segmen (attribut). Ke-32 segmen diolah menggunakan klasifikasi *Discriminant Analysis* untuk identifikasi kontur otolith ikan. Dengan metode klasifikasi *Discriminant Analysis* diperoleh hasil akurasi sebesar 92%(Salimi et al., 2016).

Pada akhir tahun 2017 Shiddiqui et al, melakukan penelitian tentang klasifikasi spesies ikan dari citra ikan underwater menggunakan *Deep Learning* metode *Convolutional Neural Network* dengan dataset primer 2209 citra ikan dalam 16 kelas spesies ikan. Citra ikan dibagi menjadi 1309 dataset untuk pelatihan dan sisanya untuk pengujian. Akurasi klasifikasi yang dihasilkan dalam penelitian sebesar 94,43%(Siddiqui et al., 2018).

Deep learning adalah cabang ilmu dari machine learning yang berbasis jaringan saraf tiruan (JST) atau bisa dikatakan perkembangan dari JST. Perbedaan dengan JST sendiri adalah banyaknya hidden layer pada *deep learning* yang dimodelkan sedemikian rupa sehingga mampu memberikan output yang lebih akurat. *Deep learning* mengajari komputer melakukan sesuatu yang natural seperti manusia dan memiliki beberapa algoritma, misalnya untuk tugas mendeteksi dan mengklasifikasi. Salah satu algoritma *deep learning* yang digunakan dalam penelitian ini adalah *convolutional neural network* dimana dapat memproses data 2 dimensi, misalnya gambar. *Convolutional Neural Network* ini diklaim sebagai algoritma terbaik dan paling banyak digunakan untuk

klasifikasi dan mendeteksi objek dari data citra digital(Beale, Hagan, & Demuth, 2017).

Penggunaan *Deep Learning* dengan metode *Convolutional Neural Network* pertama kali berhasil diaplikasikan oleh Yann LeCun pada tahun 1998. Pada penelitian ini Yann LeCun mengemukakan metode CNN untuk mengenal tulisan tangan untuk keperluan pembacaan dokumen. Hasil yang didapat dari penelitian tersebut menunjukkan akurasi yang cukup tinggi hingga mencapai *test error* hanya sebesar 1,7%(Y. Lecun, Bottou, & Bengio, 1998).

Penerapan metode *Convolutional Neural Network* dapat dikembangkan dari sisi arsitektur dan banyaknya lapisan yang digunakan pada jaringan. Penggunaan arsitektur yang benar akan sangat baik untuk klasifikasi citra dalam berbagai macam kategori. Contohnya seperti pada dataset ImageNet yang memiliki 1000 kategori. Pada tahun 2012 teknik *Deep Learning* dengan metode CNN dipopulerkan dengan arsitektur AlexNet yang diuji dengan dataset ImageNet tersebut. Arsitektur yang dibuat oleh Alex Krizhevsky menunjukkan hasil yang sangat signifikan pada testing set dengan test error sebesar 17%. Hasil tersebut dinilai sudah sangat luar biasa karena citra pada dataset yang digunakan saat itu sangatlah kompleks dan banyak(Krizhevsky, Sutskever, & Hinton, 2012).

Jika mempunyai sumber daya hardware yang kuat dan dataset penelitian banyak, bisa melakukan training sendiri jaringan CNN dengan dataset dalam jumlah besar dan banyak kelas tersebut seperti yang dilakukan oleh Alex tersebut di atas. Namun jika yang diperlukan melakukan proses training dan testing dataset dalam jumlah kecil (kurang dari 1000 citra), kurang efektif jika melakukan training sendiri dari awal tersebut. Untuk dataset kecil, cukup dilakukan *Transfer Learning* dengan memanfaatkan model jaringan CNN yang sudah ada(Le et al., 2018).

Banyak penelitian tentang *Transfer Learning* dari CNN untuk klasifikasi citra seperti yang telah dilakukan Jongwan Chang et al (2017) dalam penelitian sehubungan dengan Kanker Payudara. Chang melakukan *Transfer Learning*

dari CNN Model Google Inception v3 yang telah ditraining(Chang & Park, 2017).

Gradient Descent (GD) merupakan algoritma optimasi orde pertama untuk mencari nilai minimum lokal dari suatu fungsi(Lemaréchal, 2012). GD terus berkembang sepanjang waktu, sehingga banyak peneliti mengembangkan optimasi pada GD untuk meningkatkan kinerjanya. *Stochastic Gradient Descent* (SGD) hanya memakai satu data yang dipilih secara acak untuk diproses(Leon Bottou, 2010). Dalam hal meningkatkan kecepatan learning, *Momentum Gradient Descent* (MGD) memberi kecepatan tambahan pada proses learning(Qian, 1999). Gabungan SGD dan MGD *Momentum Stochastic Gradient Descent* (MSGD), menghasilkan kinerja optimasi yang lebih baik lagi(Loizou & Richtárik, 2017).

Berdasarkan hasil penelitian sebelumnya tersebut maka peneliti tertarik untuk melakukan penelitian klasifikasi citra otolith ikan menggunakan *Convolutional Neural Network* dengan *transfer learning* dari model CNN AlexNet berdasarkan optimasi metode *Momentum Stochastic Gradient Descent* (MSGD).

TINJAUAN PUSTAKA

Citra Digital

Citra merupakan gambar analog dalam 2 dimensi. Dari sudut pandang matematika, citra adalah fungsi menerus dari intensitas cahaya pada bidang dua dimensi. Di dalam sebuah citra mengandung banyak informasi yang sering mengalami derau (noise), mengakibatkan informasi yang diperoleh dari citra tersebut menjadi kurang akurat. Derau (noise) yang terjadi misalnya warna yang kelebihan kontras, ketajamannya kurang, kabur (blurring) dan lain sebagainya.

Citra digital adalah citra yang diproses dengan komputer. Citra digital bisa dijelaskan dengan matriks berukuran $N \times M$

Interval $(0, G)$ adalah skala keabuan (grayscale). Besar G tergantung dari proses digitalisasinya. Biasanya derajat keabuan 0

menyatakan intensitas hitam, G menyatakan intensitas putih. Untuk citra 8-bit nilai $G=2^8=256$ warna (derajat keabuan). Teknik dasar menampilkan warna pada citra digital didasarkan pada penelitian bahwa sebuah warna adalah kombinasi dari 3 warna dasar, yaitu merah(Red), hijau(Green), dan biru(Blue) - RGB(Rafael C. Gonzalez & Woods, 2002).

Citra Berwarna

Citra berwarna 8-bit pada setiap piksel hanya diwakili oleh 8-bit dengan jumlah warna maksimum yang dapat dipakai sebanyak 256 warna. Ada dua jenis warna 8-bit. Pertama, citra warna 8-bit dengan menggunakan palet 256 warna yang mana tiap paletnya mempunyai pemetaan nilai (colormap) RGB tertentu.

Citra warna 16-bit (biasanya disebut sebagai citra high color), setiap pikselnya diwakili dengan 2 byte memory (16-bit). Citra berwarna 16-bit memiliki sebanyak 65.536 warna. Dalam formasi bitnya, nilai merah dan biru berada di 5-bit pada kanan dan kiri. Komponen hijau memiliki 5-bit dengan 1-bit tambahan. Pemilihan warna hijau ini, dengan deret 6-bit disebabkan penglihatan manusia lebih sensitif akan warna hijau.

Citra warna 24-bit pada setiap piksel dari citra warna 24-bit diwakili dengan 24-bit sehingga total mengandung 16.777.216 variasi warna. Variasi ini sudah lebih dari cukup untuk memvisualisasikan seluruh warna yang bisa dilihat oleh mata manusia. Penglihatan manusia dipercaya hanya mampu membedakan sampai 10 juta warna saja. Setiap nilai informasi piksel (RGB) disimpan ke dalam 1 byte data. 8-bit pertama adalah nilai biru, kemudian berikutnya nilai hijau pada 8-bit kedua dan pada 8-bit terakhir adalah merah(Rafael C. Gonzalez & Woods, 2002).

Machine Learning

Machine Learning Sejak komputer ditemukan para peneliti telah berpikir adakah kemungkinan agar komputer dapat belajar. Jika kita mengerti bagaimana cara memprogram komputer agar mereka dapat belajar, dan

berkembang dari pengalaman secara otomatis, hasilnya akan luar biasa dramatis. Bayangkan komputer belajar dan data-data medical untuk menemukan cara baru menangani suatu penyakit belajar dari pengalaman untuk mengoptimalkan energi yang dibutuhkan untuk melakukan pekerjaan rumah tangga, dan lain-lain. Sejak tahun 1980-an bidang ilmu soft computing mulai muncul dan berkembang berdampingan dengan bidang ilmu konvensional hard computing. Adapun yang membedakan antara kedua ilmu ini adalah setelah diprogram hard computing akan memberikan hasil yang sama untuk input yang sama, sementara soft computing akan belajar dari input-input yang diberikan sebelumnya untuk memberikan hasil yang lebih kuat di masa depan (Fajar, 2013).

Deep Learning

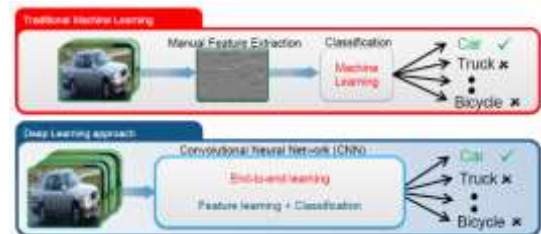
Deep learning merupakan metode baru dalam dunia *machine learning*, dimana dalam beberapa tahun terakhir menjadi topik hangat karena berhasil mencapai hasil terbaik dalam berbagai permasalahan dalam bidang AI. Teknologi *Deep Learning* ini sendiri pada dasarnya memberikan kemampuan pada komputer atau media atau gadget seperti smartphone dan tablet untuk bisa mempelajari sesuatu yang berasal dari berbagai sumber. Data-data yang ada kemudian bisa diinterpretasi dan dimanfaatkan oleh para pengguna.

Ada berbagai contoh penggunaan teknologi *Deep Learning* yang bisa dimanfaatkan dalam kehidupan sehari-hari. Teknologi ini bisa dimanfaatkan untuk mempercepat proses pencarian gambar, mampu menemukan berbagai informasi yang bersifat personal misalnya saja data pembayaran atau foto dari individu yang bersangkutan, mampu mengenali anggota keluarga berdasarkan foto yang ada yang kemudian bisa dimanfaatkan untuk melakukan tagging secara otomatis dan berbagai contoh lainnya seperti fitur di media sosial Facebook (Yann Lecun, Bengio, & Hinton, 2015).

Perbedaan Deep Learning dan Machine Learning

Deep Learning adalah cabang dari *Machine Learning*. Dengan *Machine Learning*, proses secara manual mengekstrak fitur gambar yang

relevan. Dengan *Deep Learning*, proses memberi umpan langsung gambar ke dalam jaringan *Deep Learning* yang mempelajari fitur secara otomatis. *Deep Learning* seringkali membutuhkan ratusan ribu atau jutaan gambar untuk hasil terbaik. Ini juga merupakan komputasi intensif dan membutuhkan GPU berperforma tinggi (Beale et al., 2017).



Gambar 0.1 Ilustrasi Perbedaan *Machine Learning* dan *Deep Learning*

Klasifikasi

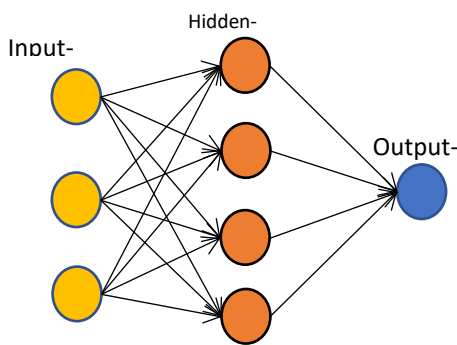
Klasifikasi adalah suatu pengelompokan data dimana data yang digunakan tersebut mempunyai kelas label atau target. Sehingga algoritma-algoritma untuk menyelesaikan masalah klasifikasi dikategorisasikan ke dalam supervised learning atau pembelajaran yang diawasi. Maksud dari pembelajaran yang diawasi adalah data label atau target ikut berperan sebagai ‘supervisor’ atau ‘guru’ yang mengawasi proses pembelajaran dalam mencapai tingkat akurasi atau presisi tertentu.

Klasifikasi merupakan salah satu teknik dasar dalam bidang data mining. Teknik klasifikasi diperlukan untuk dapat melakukan pengenalan pola, prediksi nilai, hingga pengambilan keputusan. Dalam hal pengambilan keputusan, diperlukan sistem rekomendasi yang berperan dalam memberikan rekomendasi sejumlah item kepada user sehingga dapat membantu user dalam mengambil keputusan (A. S. Abdul Kadir, 2013).

Artificial Neural Network (Jaringan Saraf Tiruan)

Artificial Neural Network (ANN) adalah suatu konsep rekayasa pengetahuan dari cabang ilmu kecerdasan buatan, yang dirancang meniru sistem saraf manusia, yang mana pengendali utama sistem saraf manusia terletak di otak. Otak manusia berisi berjuta-juta sel saraf

(neuron) yang bertugas untuk memproses informasi. Sebagaimana yang dijelaskan bahwa kemampuan ANN dalam melakukan pembelajaran dirancang sedemikian rupa seperti kinerja otak manusia, dimana manusia mempunyai kemampuan dapat memproses informasi, mengingat, dan melakukan perhitungan. Beberapa permasalahan yang dapat diselesaikan dengan ANN antara lain prediksi, klasifikasi, optimasi dan pengenalan pola. Berdasarkan kemampuan yang dimiliki, hasil dari pembelajaran *Artificial Neural Network* dapat digunakan untuk menemukan solusi dari suatu permasalahan dalam kehidupan sehari-hari. Dari waktu ke waktu ANN ini berkembang terus dan semakin maju dan lebih baik daripada sebelumnya. Perusahaan raksasa IT seperti Google, Microsoft, Facebook, Amazon, dll. melakukan riset penelitian terus menerus untuk mengembangkan kecanggihan kecerdasan buatan berbasis *Neural Network*. Hingga sampai saat ini telah berkembang maju pesat teknologi *Deep Learning* dengan *neural network* yang lebih canggih dengan banyakk *hidden layer* seperti CNN, RNN, dan sejenisnya. Di bawah ini adalah struktur ANN.



Gambar 0.2 Struktur ANN/JST(“Neural Network architectures,” n.d.)

Gambar diatas adalah struktur jaringan saraf tiruan terdiri dari:

- *Input Layer*

Input layer merupakan *node-node* yang masing-masing menyimpan sebuah nilai *input* yang tidak berubah saat fase latih dan hanya dapat berubah apabila diberikan nilai *input* baru. *Node* pada lapisan ini berdasarkan pada jumlah *input* dari suatu pola.

- *Hidden Layer*

Lapisan ini tidak pernah kelihatan sehingga dinamakan *hidden layer* (lapisan tersembunyi). Namun semua proses pada tahap pelatihan dan tahap pengenalan dilakukan pada lapisan ini. Jumlah lapisan ini tergantung dari arsitektur yang dirancang, umumnya hanya terdiri dari satu lapisan *hidden layer*.

- *Output Layer*

Output layer akan memunculkan hasil perhitungan sistem oleh fungsi aktivasi di lapisan *hidden layer* berdasarkan input yang sudah diterima (Muhammad Arhami, S.Si., 2006).

Pembelajaran Pada ANN/JST

Pembelajaran pada jaringan saraf tiruan merupakan proses pencarian konfigurasi bobot-bobot dalam tiap layer. Perubahan bobot akan terjadi pada saat proses pembelajaran. Berdasarkan pembelajarannya Jaringan Saraf Tiruan dibagi menjadi:

- *Supervised Learning* (pembelajaran terawasi)

Sistem pembelajaran pada *supervised* adalah jumlah masukan dan jumlah keluaran (target) pada jaringan ditentukan. Target yang ditentukan akan dibandingkan dengan hasil *output* pada jaringan. Perbedaan antara target dan output ini disebut dengan *error*. Apabila hasil output dan target pada jaringan menghasilkan *error* yang cukup besar, maka perlu dilakukan pembelajaran ulang sampai diperoleh *error* yang cukup kecil. Ada beberapa metode pembelajaran *supervised* antara lain *single layer perceptron*, *multi layer perceptron* dan *backpropagation*.

- *Unsupervised Learning* (pembelajaran tak terawasi)

Sistem pembelajaran pada *unsupervised* tidak memerlukan target jaringan keluaran sehingga jaringan akan mengatur sendiri keluarannya. Metode pembelajaran *unsupervised* antara lain metode *kohonen*, *hopfield*, *radial basic function* dan lain sebagainya.

- *Hybrid Learning* (pembelajaran campuran)

Merupakan gabungan dari metode pembelajaran *Supervised Learning* dengan *Unsupervised Learning*, sebagian dari bobot-bobotnya ditentukan dengan pembelajaran terawasi dan sebagian yang lainnya ditentukan dengan pembelajaran tak terawasi. Contoh dari metode *hybrid learning* misalnya seperti algoritma RBF (*Radial Basis Function*) (Kubat, 2017).

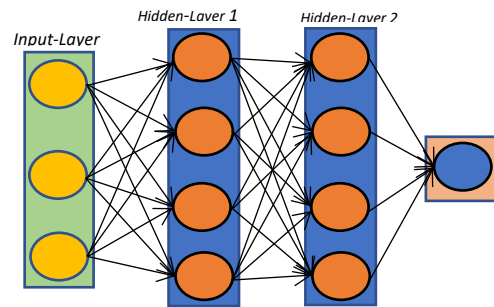
Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang dirancang untuk mengolah data 2 dimensi. CNN ini ada dalam jenis *Deep Neural Network* karena kedalaman jaringannya dengan banyak *hidden layer* dan merupakan *Neural Network* yang banyak digunakan pada penelitian untuk permasalahan data citra. Pada masalah klasifikasi citra, MLP kurang sesuai untuk diaplikasikan karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang mandiri sehingga hasilnya kurang baik.

CNN pertama kali dikembangkan berupa *Neo Cognitron* oleh Kunihiko Fukushima, seorang ahli yang bekerja di *NHK Broadcasting Science Research Laboratories*, Tokyo-Jepang. Metode tersebut berikutnya dimatangkan oleh Yann LeChun, seorang ahli peneliti dari *AT&T Bell Laboratories* New Jersey-USA. Model CNN bernama LeNet sukses diterapkan dengan baik oleh LeChun pada penelitiannya masalah pengenalan angka juga tulisan tangan. Tahun 2012, Alex Krizhevsky menerapkan model CNN miliknya dengan nama *AlexaNet* sukses juara pada kompetisi *ImageNet Large Scale Visual Recognition Challenge 2012*. Prestasi tersebut merupakan tonggak pembuktian metode *Deep Learning*, khususnya CNN yang mampu mengungguli akurasi metode *Machine Learning* seperti SVM pada penelitian klasifikasi obyek pada citra.

CNN merupakan variasi dari *Multilayer Perceptron* (MLP) yang terinspirasi dari cara kerja jaringan saraf manusia. Penelitian awal yang mendasari penemuan ini pertama kali dilakukan oleh Hubel dan Wiesel yang melakukan penelitian *visual cortex* pada indera

penglihatan kucing. *Visual cortex* pada hewan sangat kuat dalam sistem pemrosesan visual yang pernah ada. Hingga banyak penelitian yang terinspirasi dari cara kerjanya dan menghasilkan model-model baru diantaranya seperti *Neo cognitron*, HMAX, dan lain-lainnya. Cara kerja CNN mempunyai kemiripan dengan MLP, tapi pada CNN setiap *neuron* dipresentasikan dalam bentuk 2 dimensi, tidak seperti MLP yang mana setiap *neuron* hanya berukuran 1 dimensi saja. Dengan kelebihan ini, CNN lebih banyak bisa memecahkan masalah dibandingkan MLP



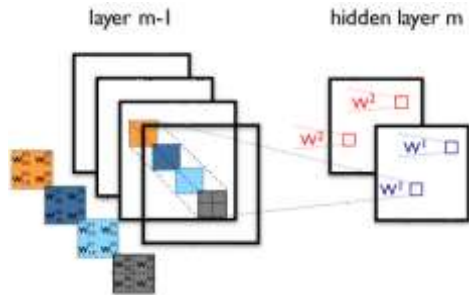
Gambar 0.3 Diagram MLP (“Deep Learning 101 - Part 2: Multilayer Perceptrons,” n.d.)

ar 2.3. mempunyai 1 layer (kotak hijau, biru dan orange) dan masing-masing layer terdiri dari *j* neuron (gambar lingkaran). Jaringan MLP menerima input data 1 dimensi kemudian mempropagasikan data dalam jaringan untuk menghasilkan output. Setiap hubungan antar neuron pada 2 layer yang berdampingan mempunyai parameter bobot 1 dimensi. Pada setiap data input dalam layer melakukan operasi linear dengan nilai bobot yang ditentukan, kemudian hasil perhitungan akan diubah bentuknya menggunakan operasi non linear yang disebut dengan fungsi aktivasi.

Pada jaringan CNN data yang dipropagasikan adalah data 2 dimensi, sehingga yang terjadi operasi linear dan parameter bobot dalam CNN berlainan. Pada CNN operasi linear memakai proses operasi konvolusi, sedangkan bobot tidak lagi 1 dimensi, tapi berbentuk 4 dimensi yang merupakan kumpulan dari kernel konvolusi, ilustrasinya tampak seperti pada Gambar 2.4. Dimensi bobot pada CNN sebagai berikut:

Neuron Input x Neuron Output x Tinggi x Lebar...

Karena sifat dari proses konvolusi inilah, maka CNN hanya bisa dipakai pada data yang mempunyai struktur 2 dimensi semacam citra atau suara.



Gambar 0.4 CNN Proses Konvolusi (“Convolutional Neural Network - Documentation,” n.d.)

Arsitektur Jaringan CNN

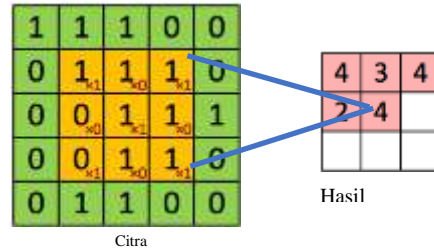
Jaringan Saraf Tiruan terdiri dari bermacam layer dan banyak neuron pada tiap-tiap layer. Kedua hal tersebut tidak bisa ditetapkan menggunakan aturan yang pasti dan berlaku berbeda-beda pada data yang berlainan.

Dalam MLP sebuah jaringan tanpa hidden layer bisa memetakan persamaan linear apa pun, sedangkan jaringan dengan 1 atau 2 *hidden layer* mampu memetakan kebanyakan persamaan dengan data yang sederhana. Tapi pada data yang lebih rumit, MLP mempunyai keterbatasan. Pada kasus jumlah *hidden layer* kurang dari 3 layer, ada pendekatan untuk menetapkan jumlah neuron pada tiap-tiap layer agar mendekati hasil yang optimal. Penggunaan di atas 2 layer pada umumnya tidak disarankan karena akan menyebabkan terjadinya *overfitting* serta akurasi dari backpropagation berkurang signifikan.

Dengan berkembangnya metode *deep learning*, ditemukan bahwa untuk menyelesaikan kelemahan MLP saat menangani data kompleks, diperlukan fungsi untuk mengubah bentuk data input menjadi lebih mudah dimengerti oleh MLP. Hal itu membuat berkembangnya metode *deep learning* yang mana dalam 1 model digunakan beberapa layer untuk mengerjakan transformasi data, sebelum

diolah memakai metode klasifikasi. Hal tersebut menimbulkan berkembangnya model neural network dengan lebih dari 3 hidden layer. Namun disebabkan karena fungsi layer awal untuk metode fitur ekstraksi, maka jumlah layer pada sebuah jaringan CNN tidak memiliki aturan baku tergantung dataset yang dipakai dalam penelitian.

Dalam pengolahan citra, konvolusi menerapkan sebuah kernel pada citra ke seluruh area citra seperti yang ditunjukkan pada Gambar 2.5. Kotak warna hijau seluruhnya merupakan area citra yang akan dikonvolusi. Kernel diproses mulai dari sudut kiri atas menuju ke kanan bawah. Hasil konvolusi pada citra tersebut tampak pada gambar di sebelah kanan. Tujuan dilakukan proses konvolusi pada citra adalah bertujuan mengekstraksi fitur yang terdapat pada citra input. Konvolusi akan menghasilkan transformasi linear dari citra input. Bobot pada layer mengkhhususkan kernel konvolusi yang akan digunakan, sehingga kernel konvolusi bisa dilatih menurut citra input pada CNN.

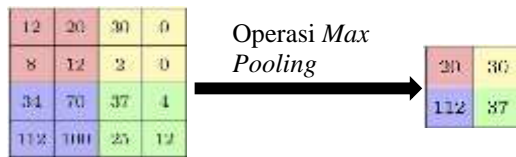


Gambar 0.5 Operasi Konvolusi (“Convolutional Neural Network (CNN),” n.d.)

Subsampling merupakan proses memperkecil ukuran suatu data citra. Kebanyakan jaringan CNN, metode *sub sampling* yang dipakai adalah *max pooling*. *Max pooling* memecah output dari *convolution layer* menjadi beberapa grid kecil kemudian mencari nilai maksimal dari setiap grid untuk disusun matriks citra yang telah diperkecil ukurannya seperti tampak pada Gambar 2.6. Grid berwarna merah, kuning, biru dan hijau adalah kelompok grid yang akan dicari nilai angka terbesarnya.

Hasil proses *subsampling* dimaksud bisa dilihat pada kotak grid yang lebih kecil ukurannya yang terletak pada sebelah kanannya.

Proses ini memastikan fitur yang diperoleh masih tetap sama walaupun obyek citra mengalami pergeseran.



Gambar 0.6 Operasi Max Pooling (“CS231n Convolutional Neural Networks for Visual Recognition,” n.d.)

Menurut Springenberg et al., penggunaan *max pooling* layer pada CNN hanya bertujuan untuk memperkecil ukuran citra sehingga bisa mudah diganti dengan suatu *convolution layer* beserta *stride* yang sama dengan *pooling layer* yang dimaksud.

Convolutional Neural Networks merupakan suatu jaringan yang mana mempunyai susunan neuron 3D (lebar, tinggi, kedalaman). Lebar dan tinggi adalah ukuran dari layer, sedangkan kedalaman merupakan jumlah layer. Secara umum pada jaringan CNN ada 2 jenis layer utama yaitu:

- a. Layer ekstraksi fitur citra, terletak di awal arsitektur jaringan yang tersusun dari beberapa layer dan tiap layernya tersusun atas *neuron* yang terhubung di daerah lokal (*local region*) pada layer sebelumnya. Layer pertama yaitu layer konvolusi dan layer kedua disebut *layer pooling*. Setiap layer diterapkan fungsi aktivasi. Posisinya berselang-seling antara layer jenis pertama dengan layer jenis kedua. Layer ini memproses input citra secara langsung hingga menghasilkan *output* berupa vektor untuk diproses pada layer selanjutnya.
- b. Layer untuk klasifikasi, tersusun dari beberapa layer dan tiap layernya tersusun dari *neuron* yang terhubung secara penuh (*fully connected*) dengan layer lainnya. Layer ini menerima input berdasarkan hasil *output* layer ekstraksi fitur citra berupa vektor yang selanjutnya ditransformasikan dengan tambahan beberapa *hidden layer*. Hasil *output* berupa nilai angka kelas untuk klasifikasi.

Dengan demikian CNN merupakan metode untuk mentransformasikan gambar original

layer per layer dari nilai piksel gambar ke dalam nilai skoring kelas untuk klasifikasi. Dan setiap layer ada yang memiliki *hyperparameter* dan ada yang tidak memiliki parameter (bobot dan bias pada neuron).

- a. *Convolutional Layer*. Layer yang pertama kali menerima input gambar langsung pada arsitektur. Operasi pada layer ini sama dengan operasi konvolusi yaitu melakukan operasi kombinasi linier filter terhadap daerah lokal. Filter merupakan representasi bidang *reseptif* dari *neuron* yang terhubung kedalam kedalam daerah lokal (*local connectivity*) pada input gambar. Bentuk layer direpresentasikan sebagai volume $B \times K \times L$ atau layer ukuran $B \times K$ dengan jumlah sebanyak L . *Convolutional layer* memiliki *hyperparameter* dan parameter.
- b. *Pooling Layer C1*. Pooling layer akan mereduksi ukuran spasial dan jumlah parameter dalam jaringan serta mempercepat komputasi dan mengontrol terjadinya *overfitting*. *Pooling layer* bekerja dengan blok spasial yang bergerak sepanjang ukuran *feature pattern*. Ukuran pergeseran blok pada umumnya adalah ukuran pada dimensi blok ($H \times H$) itu sendiri sehingga tidak ada *overlapping* seperti pada *Convolutional Layer*. Pergerakan blok diikuti dengan perhitungan pooling pada masukan pola fitur. Pada layer ini tidak memiliki parameter karena parameter sudah ditentukan sebelumnya (*fixed*). Pooling layer memiliki beberapa macam tipe antara lain *average pooling*, *max pooling*, dan *Lp Pooling*.
- c. Fungsi Aktivasi (*Neurons*). Fungsi aktivasi atau fungsi transfer merupakan fungsi non-linear yang memungkinkan sebuah jaringan untuk dapat menyelesaikan permasalahan permasalahan *non trivial*. Setiap fungsi aktivasi mengambil sebuah nilai dan melakukan operasi matematika. Pada arsitektur CNN, fungsi aktivasi terletak pada perhitungan akhir keluaran *feature map* atau sesudah proses perhitungan konvolusi atau pooling untuk menghasilkan suatu pola fitur. Beberapa macam fungsi aktivasi yang sering

digunakan dalam penelitian antara lain fungsi *sigmoid*, *tanh*, *Rectified Liniear Unit* (ReLU), *Leaky ReLU* (LReLU) dan *Parametric ReLU* (“CS231n Convolutional Neural Networks for Visual Recognition,” n.d.).

Momentum Stochastic Gradient Descent (MSGD)

Metode optimasi *Stochastic Gradient Descent* (SGD) sering digunakan untuk pelatihan *Artificial Neural Network*. Metode ini berperan dalam menemukan nilai bobot yang memberikan nilai keluaran terbaik. Prinsip kerja metode *Gradient Descent* adalah memperkecil nilai fungsi dengan mengubah nilai parameter selangkah demi selangkah. Semakin baik kinerja metode optimasi maka akurasi dan waktu proses yang diberikan oleh jaringan akan semakin baik pula. Formulanya adalah sebagai berikut:

$$\theta_{\ell + 1} = \theta_{\ell} - \alpha \nabla E(\theta_{\ell}) \dots \dots \dots (0.1)$$

di mana ℓ mewakili nomor iterasi, $\alpha > 0$ adalah tingkat pembelajaran, θ adalah vektor parameter, dan $E(\theta)$ adalah fungsi kerugian. Gradien fungsi kerugian, $\nabla E(\theta)$, dievaluasi menggunakan seluruh set pelatihan, dan algoritma *gradient descent* standar menggunakan seluruh kumpulan data sekaligus.

Algoritma *stochastic gradient descent* mengevaluasi gradien dan memperbarui parameter menggunakan subset dari set pelatihan. Bagian ini disebut *mini-batch*. Setiap evaluasi gradien menggunakan *mini-batch* adalah iterasi. Pada setiap iterasi, algoritma mengambil satu langkah untuk meminimalkan fungsi kerugian.

Algoritma *stochastic gradient descent* berhasil di sepanjang jalur penurunan curam menuju optimum. Menambahkan momentum ke pembaruan parameter adalah salah satu cara untuk mengurangi osilasi ini. Formula Penurunan *stochastic gradient descent* dengan pembaruan momentum adalah sebagai berikut:

$$\theta_{\ell + 1} = \theta_{\ell} - \alpha \nabla E(\theta_{\ell}) + \gamma(\theta_{\ell} - \theta_{\ell - 1}) \dots \dots \dots$$

di mana γ menentukan kontribusi dari langkah gradien sebelumnya ke iterasi saat ini. Metode penambahan momentum pada proses tersebut inilah yang dimaksud dengan metode

Momentum Stochastic Gradient Descent (MSGD)(Beale et al., 2017).

Alexnet

Alexnet adalah salah satu model arsitektur CNN yang pertama kali diperkenalkan oleh Alex Krizhevsky, Ilya Sutskever dan Geoffrey E. Hinton dari Universitas Toronto Kanada dalam penelitiannya yang berjudul “*ImageNet Classification with Deep Convolutional Neural Networks*” tahun 2012 dengan dataset 1.5 juta citra dengan 1000 Class dari server ImageNet. Model Alexnet ini meraih top-5 test akurasi di ImageNet, pemenang pada *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) 2012(Krizhevsky et al., 2012). Alexnet adalah tonggak awal kebangkitan *Deep Learning* yang mana sebelumnya penelitian Yan LeeCun mengalami stagnasi sejak tahun 1998. Setelah sukses penelitian Alex, semakin banyak berkembang penelitian tentang *Deep Learning*, termasuk *Convolutional Neural Network* (CNN) ini.



Gambar 0.7 Ilustrasi Jaringan CNN Model Alexnet(Krizhevsky et al., 2012)

Model jaringan ALEXNET ini memiliki 8 lapisan inti dengan bobot yang dapat dipelajari: 5 lapisan konvolusi untuk tugas fitur ekstraksi, dan di akhir jaringan terdapat 3 lapisan yang terhubung sepenuhnya (*Fully Connected Layer*), lapisan *dropout*, dan lapisan *softmax* untuk fungsi klasifikasi. Setiap lapisan konvolusi terdapat fungsi aktivasi dan selang-seling beberapa lapisan terdapat lapisan *pooling*(Krizhevsky et al., 2012).

Transfer Learning

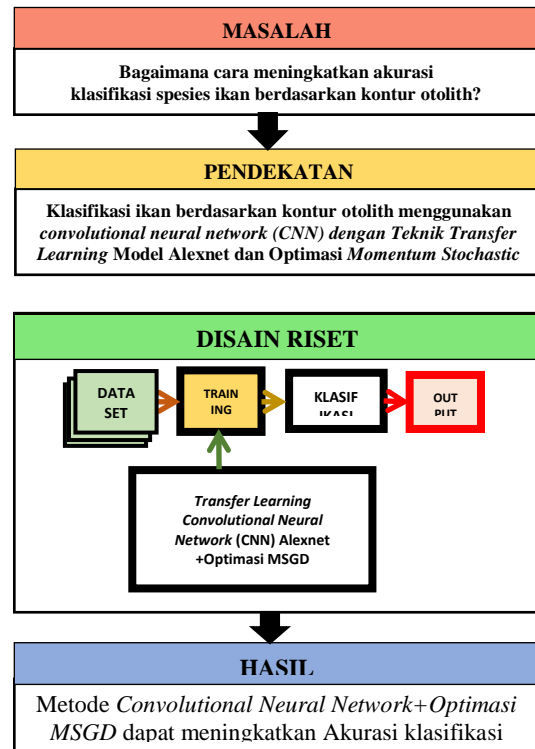
Transfer Learning biasanya digunakan dalam aplikasi *Deep Learning*. Penyesuaian jaringan dengan *transfer learning* jauh lebih cepat dan mudah daripada membangun dan melatih jaringan baru dari awal. Metode ini akan dapat dengan cepat mentransfer pembelajaran ke

tugas baru dengan menggunakan sejumlah kecil gambar pelatihan. Keuntungan dari *transfer learning* adalah bahwa jaringan pra-operasi telah mempelajari serangkaian fitur yang kaya. Fitur-fitur ini dapat diterapkan pada berbagai macam tugas sejenis lainnya. Misalnya, mengambil jaringan yang terlatih pada jutaan gambar dan melatihnya untuk klasifikasi objek baru yang spesifik hanya dengan menggunakan ratusan gambar saja.

Transfer Learning bisa cocok digunakan dalam CNN untuk klasifikasi citra dengan dataset kurang dari 1000 Citra. Dalam *Transfer Learning*, digunakan model pra-latih lalu disesuaikan dimodifikasi arsitekturnya termasuk opsi-opsi training sesuai yang dibutuhkan. *Convolutional Neural Network (CNN)* membutuhkan dataset besar dan banyak waktu untuk melatih. Beberapa jaringan bisa memakan waktu hingga 2-3 minggu di beberapa GPU untuk dilatih. *Transfer learning* adalah teknik yang sangat berguna yang mencoba mengatasi kedua masalah tersebut. Alih-alih melatih jaringan dari awal, *transfer learning* menggunakan model yang terlatih pada dataset yang berbeda, dan menyesuaikannya dengan masalah yang akan diselesaikan (Beale et al., 2017).

METODOLOGI PENELITIAN

Pengolahan Data



Data citra diperoleh dari *University of Malaya-Institute Animal Care and Use Committee (UM-IACUC)* yang didownload dari internet di alamat <https://peerj.com/preprints/1517/>. Data berupa citra dari otolith ikan ini berjumlah 403 citra terdiri dari 3 (tiga) *family*, yaitu *Engraulidae*, *Sciaenidea*, dan *Ariidae*. *Family Sciaenidea* terdiri dari 5 (lima) spesies yaitu *Dendrophysa Russeli*, *Johnius Belangerii*, *Johnius carouna*, *Otolithes Ruber*, dan *Panna Microdon*. *Family Ariidae* terdiri dari 6 (enam) spesies yaitu *Nemapteryx caelata*, *Arius maculatus*, *Cryptarius Truncatus*, *Hexanemathictys sagor*, *Osteogeneiosus militaris*, dan *Plicofollis Argyropleuron*. *Family Engraulidae* terdiri dari 3 (tiga) spesies yaitu *Coilia Dussumieri*, *Setipinna Taty*, dan *Thryssa Hamiltonii*.

Pengolahan Data Awal (Preprocessing)

Data citra yang diperoleh dilakukan pengolahan data awal (*preprocessing*) data citra

terdahulu. Proses *preprocessing* dimaksudkan agar diperoleh citra yang lebih baik sehingga hasil proses selanjutnya menjadi lebih baik pula. Adapun tahapannya sebagai berikut:

1. Citra RGB dilakukan pemusatan obyek dengan cara di-cropping
 Agar mendapatkan hasil maksimal dalam training nantinya, dataset otolith ikan dilakukan cropping agar citra otolith bisa lebih fokus dan seragam. Cropping dilakukan secara manual dengan bantuan software tool pengolah citra.
2. Citra RGB dilakukan resize sebesar 227x227 piksel
 Agar bisa diproses dalam training nantinya, dataset otolith ikan harus dilakukan resize sesuai dengan input yang ditentukan oleh Jaringan CNN Alexnet yaitu 227x227x3. Angka 3 menunjukkan channel dari RGB yaitu R(Red) Green(G) dan Blue(B). Resizing citra dilakukan secara otomatis dengan bantuan software tool pengolah citra.
3. Pembersihan Citra dari noise atau derau.
 Derau (noise) adalah pixel yang mengganggu kualitas gambar sehingga menyebabkan degradasi dalam citra. Hal ini bisa disebabkan oleh gangguan fisis dari kamera saat pengambilan citra atau akibat kesalahan pada proses pengolahan citra. Tujuan membersihkan derau adalah untuk memuluskan citra sehingga dapat mengurangi derau, yang akhirnya dapat meningkatkan kualitas citra. Pembersihan noise dilakukan dengan cara manual dengan tool pengolah citra.

Memisahkan Dataset Training dan untuk Testing

Dataset berjumlah 403 citra otolith ikan, yang terdiri dari 14 kelas ditentukan 90% untuk data training dan 10% data testing. Data testing diambil secara acak dari data masing-masing kelas. Total jumlah data training sebanyak 361 citra. Dan data testing sisanya sejumlah 42 citra dari semua kelas.

Menyiapkan Jaringan Convolutional Neural Network (CNN)

Metode yang dipake untuk klasifikasi dalam penelitian ini adalah *Convolutional Neural Network* dengan model jaringan Alexnet. Model Alexnet didownload dari tool Matlab 2017b dari menu Add-Ons. Model Alexnet mempunyai 5 layer konvolusi untuk tugas fitur ekstraksi dan 3 layer untuk klasifikasi. Model bisa didownload secara otomatis dari Menu Add-Ons ataupun secara manual langsung dari internet.

Transfer Learning dari Pretrained Model Alexnet

Karena model hasil training Alexnet mengklasifikasi 1000 kelas, sedangkan penelitian ini hanya akan mengklasifikasi 14 kelas, maka perlu dilakukan modifikasi penyesuaian layer untuk 3 (tiga) layer terakhir yang dipakai untuk proses klasifikasi. Proses training dengan teknik ini jauh lebih cepat dibandingkan dengan mentraining seperti dataset asli yang jumlahnya sangat besar dengan kelas yang sangat banyak.

Perancangan Struktur Dan Parameter Untuk Training

Setelah itu perlu menentukan parameter-parameter *Convolutional Neural Network*. Dengan penentuan parameter yang tepat, dihasilkan akurasi yang diharapkan. MiniBatch ditetapkan 10, epoch 4, InitialLearnRate=0.0001. Optimasi parameter menggunakan MSGD (*Momentum Stochastic Gradient Descent*). MSGD memperbarui bobot dan bias (parameter) dengan mengambil langkah kecil ke arah gradien negatif dari fungsi kehilangan, sedemikian rupa untuk meminimalkan kerugian. Ini memperbarui parameter menggunakan sub kumpulan data setiap saat.

Proses Training CNN

Setelah dataset sudah siap dipre-processing, model CNN Alexnet sudah didownload dan parameter-parameter telah ditetapkan,

selanjutnya dilakukan proses utama training CNN dengan teknik transfer learning. Proses Training menggunakan Tool Matlab versi terbaru 2017b yang sudah mendukung fitur *Deep Learning*. Model Alexnet yang sudah ada diakses dan setelah disesuaikan dengan parameter yang telah ditetapkan sebelumnya, dilakukan proses training dataset otolith ikan sampai selesai.

Proses Training terdiri dari 2 bagian, yaitu bagian fitur ekstraksi berupa 5 (lima) proses konvolusi dan bagian klasifikasi berupa 3 (tiga) proses fully connected layer. Diakhiri dengan fungsi aktivasi Softmax untuk mengklasifikasi sejumlah kelas yang telah ditentukan. Masing-masing proses ada beberapa lapisan pendukung seperti Fungsi Aktivasi ReLU, *Local Respons Normalization*, Max-Pooling. Pada proses klasifikasi ada proses Dropout dalam *Fully Connected Layer*. Seluruhnya ada 25 proses mulai dari input sampai output.

Proses Operasi Konvolusi

Operasi konvolusi adalah operasi pada dua fungsi argumen bernilai nyata. 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:

$$S(t)=(x*w)(t).....(3.1)$$

Fungsi S(t) memberikan output tunggal berupa Feature Map, argumen pertama adalah input yang merupakan x dan argumen 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. Oleh karena itu, operasi untuk konvolusi ke input dengan lebih dari satu dimensi dapat ditulis sebagai berikut:

$$S(i,j)=(K*I)(i,j)=\sum_{m,n} [I(i-m,j-n)K(m,n)](3.2)$$

Persamaan di atas adalah perhitungan dasar dalam operasi konvolusi dimana I dan j adalah piksel dari citra. Perhitungannya bersifat komutatif dan muncul saat K sebagai kernel, I sebagai input dan kernel yang dapat dibalik

relatif terhadap input. Sebagai alternatif, operasi konvolusi dapat dilihat sebagai perkalian matriks antara citra masukan dan kernel dimana keluarannya dapat dihitung dengan *dot product*.

Kita juga dapat menentukan volume output dari masing-masing lapisan dengan *hyperparameters*. *Hyperparameter* yang digunakan pada persamaan dibawah ini digunakan untuk menghitung berapa banyak neuron aktivasi dalam sekali output.

$$H=((W-F+2P))/(S+1)(3.3)$$

Dari persamaan diatas kita dapat menghitung ukuran spasial dari volume output dimana hyperparameter yang dipakai adalah ukuran volume (W), filter (F), Stride yang diterapkan (S) dan jumlah padding nol yang digunakan (P). *Stride* adalah nilai yang digunakan untuk menggeser filter melalui input citra dan *Zero Padding* adalah nilai untuk menempatkan angka nol di sekitar border citra.

Proses Fungsi Aktivasi ReLU

Banyak fungsi aktivasi yang digunakan pada algoritma Neural Network dan CNN. Dalam penelitian CNN ini menggunakan fungsi aktivasi ReLU. Fungsi Aktivasi ReLU (Rectified Linear Unit) merupakan proses aktivasi pada model CNN yang mengaplikasikan fungsi dengan rumus:

$$f(x)=\max(0,x)(3.4)$$

Yang berarti fungsi ini melakukan Thresholding dengan nilai nol terhadap nilai piksel pada input citra. Aktivasi ini membuat seluruh nilai piksel yang bernilai kurang dari nol atau bernilai negatif pada suatu piksel akan dijadikan 0.

Proses Local Respons Normalization

Local Response Normalization adalah proses yang berguna untuk menormalkan output neuron fungsi aktivasi ReLU. Neuron ReLU memiliki aktivasi yang tidak terbatas dan perlu untuk dinormalkan. Jika kita normalisasi di sekitar lingkungan lokal dari neuron yang tereksitasi, itu menjadi lebih sensitif dibandingkan dengan tetangganya. Pada saat yang sama, itu akan mengurangi respons besar seragam di lingkungan lokal tertentu. Jika semua nilai besar,

maka normalisasi nilai-nilai itu akan mengurangi semuanya. Pada dasarnya ini akan mendorong beberapa jenis penghambatan dan meningkatkan neuron dengan aktivasi yang relatif lebih besar. Proses ini pada jaringan AlexNet dilakukan setelah fungsi aktivasi ReLU pada konvolusi-1 dan konvolusi-2.

Untuk setiap elemen x dalam input, proses menghitung nilai normalisasi x' menggunakan rumus:

$$x' = x / [(K + \alpha * ss) / w]^\beta \dots \dots \dots (3.5)$$

di mana K , α , dan β adalah *hyperparameter*, dan ss adalah jumlah kuadrat dari elemen-elemen di w jendela normalisasi.

Proses operasi Max-Pooling

Pooling Layer adalah lapisan yang menggunakan fungsi dengan *Feature Map* sebagai masukan dan mengolahnya dengan berbagai macam operasi statistik berdasarkan nilai piksel terdekat. Pada model CNN, lapisan Pooling biasanya disisipkan secara teratur setelah beberapa lapisan konvolusi. Lapisan Pooling yang dimasukkan diantara lapisan konvolusi secara berturut-turut dalam arsitektur model CNN dapat secara progresif mengurangi ukuran volume output pada *Feature Map*, sehingga mengurangi jumlah parameter dan perhitungan di jaringan, dan untuk mengendalikan Overfitting. Hal terpenting dalam pembuatan model CNN adalah dengan memilih banyak jenis lapisan Pooling dan hal ini bisa menguntungkan kinerja model. Lapisan Pooling bekerja di setiap tumpukan *Feature Map* dan mengurangi ukurannya. Bentuk lapisan Pooling yang paling umum adalah dengan menggunakan filter berukuran 2×2 yang diaplikasikan dengan langkah sebanyak 2 dan kemudian beroperasi pada setiap irisan dari input. Bentuk seperti ini akan mengurangi *Feature Map* hingga 75% dari ukuran aslinya. Ada beberapa macam metode pooling seperti *Max-Pooling*, *Average-Pooling*, dan *Min-Pooling*. Dalam penelitian ini menggunakan metode *Max-Pooling*.

Lapisan Pooling akan beroperasi pada setiap

irisian kedalaman volume input secara bergantian. Pada gambar di atas, lapisan pooling menggunakan salah satu operasi maksimal yang merupakan operasi yang paling umum. Gambar 3.4. menunjukkan operasi *Max-Pooling* dengan langkah 2 dan ukuran filter 2×2 . Dari ukuran input 4×4 , pada masing-masing 4 angka pada input operasi mengambil nilai maksimalnya dan membuat ukuran output baru menjadi 2×2 . Fungsi *Max-Pooling*: $Y = \max(x)$, yang mana Y adalah neuron piksel hasil pooling dan x adalah angka-angka dalam range sesuai opsi yang telah ditentukan.

Max-Pooling menghasilkan nilai maksimum dari range inputnya. Lapisan tersebut akan memindai melalui masukan secara horizontal dan vertikal dalam ukuran langkah dengan ukuran pergeseran sesuai opsi stride yang telah ditentukan. Ukuran output lapisan penggabungan dengan ukuran input I , ukuran pool Z , padding P dan stride S memiliki dimensi M sesuai rumus:

$$M = ((I - Z + 2P)) / (S + 1) \dots \dots \dots (3.6)$$

Nilai ini harus berupa bilangan bulat agar seluruh gambar sepenuhnya tertutup. Jika kombinasi parameter ini tidak menyebabkan gambar tertutup sepenuhnya, perangkat lunak, secara default, mengabaikan bagian gambar yang tersisa di sepanjang tepi kanan dan bawah.

Proses Fully Connected Layer

Fully Connected Layer adalah lapisan di mana semua neuron aktivasi dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya seperti halnya jaringan saraf tiruan biasa. Setiap aktivasi konvolusi dari lapisan sebelumnya berbentuk output berupa array multi dimensi perlu dilakukan *flatten* untuk menjadi satu dimensi berbentuk vektor sebelum dapat dihubungkan ke semua neuron di *Fully Connected Layer*. Proses ini seperti digunakan pada metode *Multi Layer Perceptron* (MLP) dan bertujuan untuk mengolah data sehingga bisa diklasifikasikan.

Proses Dropout

Dropout adalah teknik regularisasi dalam

neural network dimana beberapa neuron akan dipilih secara acak dan tidak dipakai selama pelatihan. Neuron-neuron ini dapat dibuang secara acak. Hal ini berarti bahwa kontribusi neuron yang dibuang akan dihentikan sementara jaringan dan bobot baru juga tidak diterapkan pada neuron pada saat melakukan *backpropagation*.

Pada gambar 3.7. jaringan syaraf (a) merupakan jaringan syaraf biasa dengan 2 lapisan tersembunyi. Sedangkan pada bagian (b) jaringan syaraf sudah diaplikasikan teknik regularisasi dropout dimana ada beberapa neuron aktivasi yang tidak dipakai lagi. Teknik ini sangat mudah diimplementasikan pada model CNN dan akan berdampak pada performa model dalam melatih serta mengurangi terjadinya *overfitting*. Proses CNN pun juga akan semakin cepat karena jumlah neuron lebih sedikit.

Fungsi Aktivasi Softmax

Softmax Classifier merupakan bentuk lain dari algoritma *Logistic Regression* yang dapat kita gunakan untuk mengklasifikasi lebih dari dua kelas. Standar klasifikasi yang biasa dilakukan oleh algoritma *Logistic Regression* adalah tugas untuk klasifikasi kelas biner. Pada *Softmax* bentuk persamaan yang muncul adalah sebagai berikut:

$$S(Y_i) = \frac{e^{(Y_i)}}{(\sum_{j=1}^J e^{(Y_j)})} \dots\dots\dots(3.7)$$

Notasi S menunjukkan hasil fungsi untuk setiap elemen ke-i pada vektor keluaran kelas. Argumen Y adalah hipotesis yang diberikan oleh model pelatihan agar dapat diklasifikasi oleh fungsi *Softmax*. *Softmax* juga memberikan hasil yang lebih intuitif dan juga memiliki interpretasi probabilistik yang lebih baik dibanding algoritma klasifikasi lainnya. *Softmax* memungkinkan kita untuk menghitung probabilitas untuk semua label. Dari label yang ada akan diambil sebuah vektor nilai bernilai riil dan merubahnya menjadi vektor dengan nilai antara nol dan satu yang bila semua dijumlah akan bernilai satu.

Crossentropy Loss Function

Dalam *Supervised Learning*, training data terdiri dari input dan output/target. Pada saat *forward pass*, input akan dipropagasi menuju output layer dan hasil prediksi output akan dibandingkan dengan target dengan menggunakan sebuah fungsi yang biasa disebut dengan *Loss Function*.

Loss function digunakan untuk mengukur seberapa bagus performa dari *neural network* dalam melakukan prediksi terhadap target. Ada berbagai macam *loss function*, namun yang paling sering digunakan adalah *Squared Error* (L2 Loss) untuk regresi. Untuk klasifikasi yang biasa digunakan adalah *Cross Entropy*. *Loss Function* yang baik adalah fungsi yang menghasilkan error yang diharapkan paling rendah. Ketika suatu model memiliki kelas yang cukup banyak, perlu adanya cara untuk mengukur perbedaan antara probabilitas hasil hipotesis dan probabilitas kebenaran yang asli, dan selama pelatihan banyak algoritma yang dapat menyesuaikan parameter sehingga perbedaan ini diminimalkan. Gambaran umum algoritma ini adalah meminimalkan kemungkinan log negatif dari dataset, yang merupakan ukuran langsung dari performa prediksi model.

Pengukuran Kinerja Klasifikasi Dengan Confusion Matrix

Salah satu tahapan akhir yang penting dari sebuah proses penelitian adalah evaluasi dari model yang dihasilkan. Evaluasi dari model yang dihasilkan bertujuan untuk mengukur performa model tersebut. Untuk melakukan sebuah tes, seorang peneliti harus memiliki sekumpulan dataset yang terpisah dan tidak berhubungan dengan dataset yang dipakai untuk membentuk model tersebut.

Salah satu cara untuk melakukan evaluasi sebuah model adalah dengan menggunakan *confusion matrix*. *Confusion matrix* yang sering juga disebut *classification matrix* merupakan sebuah matrik yang memberikan gambaran penuh mengenai tingkat kesalahan (*error rate*) serta kualitas prediksi sebuah

model. Tabel 3.3 dibawah ini merupakan contoh sebuah *confusion matrix* dengan memakai dua buah kelas (“true”, “false”).

		Nilai Sebenarnya	
		Positif (P)	Negatif (N)
Hasil Prediksi	Positif (P)	True Positif (TP) ^d	False Negatif (FN) ^b
	Negatif (N)	False Positif (FP) ^c	True Negatif (TN) ^a

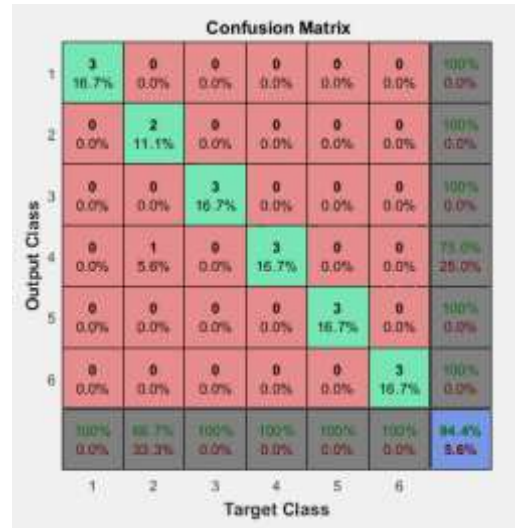
Definisi terkait dengan evaluasi yang bisa dipakai melalui *classification matrix*, diantaranya adalah:

- *Recall* atau *True Positive (TP)*, adalah proporsi dari sample bernilai “true” yang diprediksi secara benar. TP dihitung dengan menggunakan persamaan: $TP/(FP+TP)$.
- *False Positive (FP)*, yaitu proporsi antara sampel bernilai “false” yang salah diprediksi sebagai sample bernilai “true”. Persamaan yang digunakan adalah: $FN/(TN+FN)$.
- *True Negative (TN)*, didefinisikan sebagai perbandingan antara sampel bernilai “false” yang diprediksi secara benar. Persamaan yang digunakan adalah: $TN/(TN+FN)$.
- *False Negative (FN)*, didefinisikan sebagai proporsi sampel bernilai “true” yang salah diprediksi sebagai sampel bernilai “true”. Persamaan yang digunakan adalah : $FP/(FP+TF)$.
- Akurasi (AC), didefinisikan sebagai proporsi jumlah sampel yang diprediksi secara tepat, terhadap jumlah seluruh sampel. Persamaan yang digunakan adalah: $(TP+TN)/(TN+FN+FP+TP)$.
- Presisi (PR), didefinisikann sebagai proporsi jumlah sampel bernilai “true” yang berhasil diprediksi secara tepat. Persamaan yang digunakan adalah: $TP/(FN+TP)$.
- Error Rate (ER) atau tingkat kesalahan

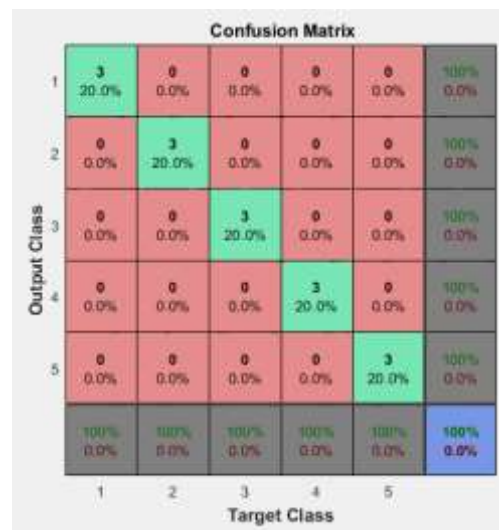
dihitung dengan persamaan $ER = 1 - AC$.

HASIL DAN PEMBAHASAN

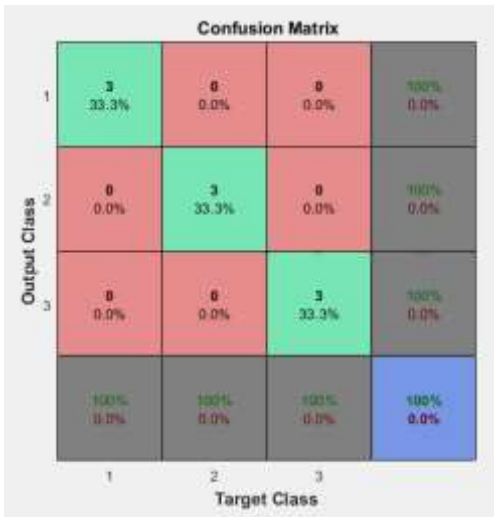
Confusion Matrix untuk Kelas Ariidae



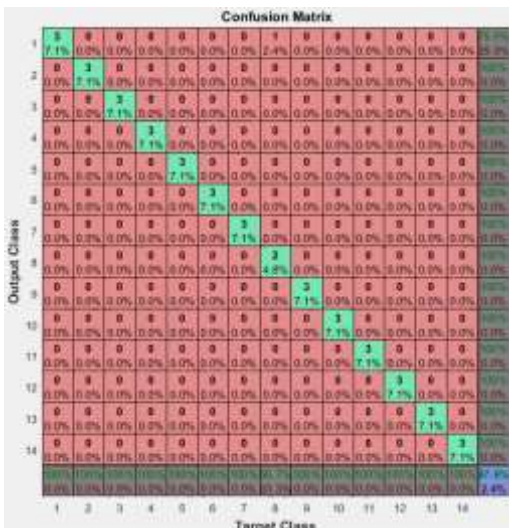
Confusion Matrix untuk Kelas Scianidea



Confusion Matrix untuk Kelas Engraulidae



Confusion Matrix untuk Semua Kelas



Gambar 0.1 Penilaian Klasifikasi Dengan Confusion Matrix (semua kelas)

PENUTUP

Kesimpulan

Hasil eksperimen yang sudah dilakukan pada penelitian ini dapat disimpulkan sebagai berikut:
 1. Preprocessing sangat dibutuhkan pada pengolahan citra untuk berbagai tujuan. Dengan pemusatan obyek berupa cropping dan menghilangkan noise di citra dataset otolith ikan berguna untuk meningkatkan akurasi klasifikasi.
 2. Pengenalan spesies ikan berdasarkan kontur otolith ikan menggunakan metode

Convolutional Neural Network memberikan akurasi sebesar 100% pada family Sciaenidae, 94.45% pada family Ariidae, 100% pada family Engraulidae, dan 95,4% pada semua family. Dibandingkan dengan menggunakan metode *Discriminant Analysis* diperoleh peningkatan akurasi peningkatan akurasi sebesar 3,2%.

Saran

Penelitian tentang pengenalan jenis ikan berdasarkan otolith memiliki peluang untuk terus dikembangkan, baik pada area preprocessing maupun pada metode klasifikasinya antara lain;

1. Klasifikasi Citra otolith ikan perlu diujicobakan menggunakan *transfer learning* model pra-latih yang lain seperti VGG, GoogleNet, ResNet, Inception, dan yang lain jika ada arsitektur model CNN yang lebih canggih di kemudian hari. Hal tersebut dimaksudkan untuk meningkatkan akurasi klasifikasi citra.
2. Perlu dikembangkan CNN menggunakan metode optimasi lainnya untuk klasifikasi citra otolith ikan, seperti PSO, GA, atau metode optimasi lain yang mungkin memberikan akurasi lebih baik.

DAFTAR PUSTAKA

A. S. Abdul Kadir. (2013). *Pengolahan Citra, Teori dan Aplikasi*. Andi Publisher.
 Alexander, V., Miloslavich, P., & Yarincik, K. (2011). The Census of Marine Life-evolution of worldwide marine biodiversity research. *Marine Biodiversity*, 41(4), 545–554.
<https://doi.org/10.1007/s12526-011-0084-1>
 Beale, M. H., Hagan, M. T., & Demuth, H. B. (2017). *Neural Network Toolbox™ MATLAB User &™s Guide*.
 Campana, S. (1999). Chemistry and composition of fish otoliths: pathways, mechanisms and applications. *Marine Ecology Progress Series*, 188, 263–297.
<https://doi.org/10.3354/meps188263>

- Chang, J., & Park, E. (2017). A Method for Classifying Medical Images using Transfer Learning : A Pilot Study on Histopathology of Breast Cancer, 2015–2018.
- Convolutional Neural Network - Documentation. (n.d.). Retrieved March 3, 2018, from <http://deeplearning.net/tutorial/lenet.html>
- Convolutional Neural Network (CNN). (n.d.). Retrieved March 1, 2018, from http://i-systems.github.io/HSE545/machine-learning/all/Workshop/KSME/04_KSME_CNN.html
- CS231n Convolutional Neural Networks for Visual Recognition. (n.d.). Retrieved March 1, 2018, from <http://cs231n.github.io>
- Deep Learning 101 - Part 2: Multilayer Perceptrons. (n.d.). Retrieved March 1, 2018, from https://beamandrew.github.io/deeplearning/2017/02/23/deep_learning_101_part2.html
- Fajar, A. (2013). *Data Mining*. Yogyakarta: Andi Offset Yogyakarta.
- Fitch, B. J. E., & Brownell, R. L. (1968). Fish Otoliths Importance Cetacean Stomachs and Their Feeding Habits Interpreting, 25(12), 2561–2574.
- Kalish, J. M. (1991). Oxygen and carbon stable isotopes in the otoliths of wild and laboratory-reared Australian salmon (*Arripis trutta*). *Marine Biology*, 110, 37–47.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, 1–9. <https://doi.org/http://dx.doi.org/10.1016/j.procy.2014.09.007>
- Kubat, M. (2017). *An Introduction to Machine Learning*. <https://doi.org/10.1007/978-3-319-63913-0>
- Le, H. T., Urruty, T., Beurton-aimar, M., Nghiem, T. P., Tran, H. T., Verset, R., ... Visani, M. (2018). Study of CNN Based Classification for Small Specific Datasets. <https://doi.org/10.1007/978-3-319-76081-0>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Lecun, Y., Bottou, L., & Bengio, Y. (1998). Gradient-based learning applied to document recognition.
- Lemaréchal, C. (2012). Cauchy and the Gradient Method. *Documenta Mathematica, ISMP*, 251–254. Retrieved from https://www.math.uni-bielefeld.de/documenta/vol-ism/40_lemarechal-claude.pdf
- Leon Bottou. (2010). Large-Scale Machine Learning with Stochastic Gradient Descent. <https://doi.org/10.1007/978-3-7908-2604-3>
- Loizou, N., & Richtárik, P. (2017). Momentum and Stochastic Momentum for Stochastic Gradient, Newton, Proximal Point and Subspace Descent Methods. Retrieved from <http://arxiv.org/abs/1712.09677>
- Lombarte, A., & Castellón, A. (1991). Interspecific and intraspecific otolith variability in the genus *Merluccius* as determined by image analysis. *Can. J. Zool.*, 69, 2442–2449. <https://doi.org/10.1139/z91-343>
- Mathworks. (2017). Introducing Deep Learning with Matlab.
- Muhammad Arhami, S.Si., M. K. (2006). *Konsep Kecerdasan Buatan*. Yogyakarta: Andi Publisher.
- Neural Network architectures. (n.d.). Retrieved February 27, 2018, from <http://cs231n.github.io/neural-networks-1/>
- Parisi-Baradad, V., Manjabacas, A., Lombarte, A., Olivella, R., Chic, Ò., Piera, J., & García-Ladona, E. (2010). Automated Taxon Identification of Teleost fishes using an otolith online database-AFORO. *Fisheries Research*, 105(1), 13–20. <https://doi.org/10.1016/j.fishres.2010.02.005>
- Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1), 145–151.

- [https://doi.org/10.1016/S0893-6080\(98\)00116-6](https://doi.org/10.1016/S0893-6080(98)00116-6)
- Rafael C. Gonzalez, & Woods, R. E. (2002). *Digital Image Processing* (2nd ed.). Prentice Hall.
- Reichenbacher, B., Sienknecht, U., Kuchenhoff, H., & Fenske, N. (2007). Combined Otolith Morphology and Morphometry for Assessing Taxonomy and Diversity in Fossil and Extant Killifish (*Aphanius, yProlebias*). *Journal of Morphology*, 268(February), 254–274. <https://doi.org/10.1002/jmor>
- Roumillat, W. A. (2004). Manual of Fish Sclerochronology. *Copeia*, 2004(1), 195–195. <https://doi.org/10.1643/OT-03-266>
- Salimi, N., Loh, K. H., Kaur Dhillon, S., & Chong, V. C. (2016). Fully-automated identification of fish species based on otolith contour: using short-time Fourier transform and discriminant analysis (STFT-DA). *PeerJ*, 4, e1664. <https://doi.org/10.7717/peerj.1664>
- Siddiqui, S. A., Salman, A., Malik, M. I., Shafait, F., Mian, A., Shortis, M. R., & Harvey, E. S. (2018). Automatic fish species classification in underwater videos: Exploiting pre-trained deep neural network models to compensate for limited labelled data. *ICES Journal of Marine Science*, 75(1), 374–389. <https://doi.org/10.1093/icesjms/fsx109>
- Soria, A., & Baradad, P. (2012). *On the Automatic Detection of Otolith Features for Fish Species Identification and their Age Estimation*.
- The Secret Lives of Fish: Cohen Lab. (n.d.). Retrieved April 1, 2018, from <http://www.whoi.edu/page.do?pid=130656&tid=3622&cid=2466>
- Thomas, O. R. B., Ganio, K., Roberts, B. R., & Swearer, S. E. (2017). Trace element–protein interactions in endolymph from the inner ear of fish: implications for environmental reconstructions using fish otolith chemistry. *Metallomics*, 9(3), 239–249. <https://doi.org/10.1039/C6MT00189K>
- VanderKooy, S. (2009). A Practical Handbook for Determining the Ages of Gulf of Mexico Fishes. *Gulf States Marine Fisheries Commission*, (167), 157. Retrieved from www.gsmfc.org