

PROTOTYPE SYSTEM KLASIFIKASI GENUS IKAN MENGGUNAKAN VIOLA-JONES FEATURE EXTRACTION DAN BOOSTING BERBASIS DECISION TREE

Nehemiah Austen Pison¹, Muhammad Eka Suryana², Med Irzal³

Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Negeri Jakarta, Jakarta Timur, Indonesia

ezraelvio12111926@gmail.com¹, eka-suryana@unj.ac.id², medirzal@unj.ac.id³

Abstrak— Sektor budidaya perikanan di Indonesia adalah sebuah sektor ekonomi yang bernilai besar bagi ekonomi masyarakat Indonesia, namun sektor ini masih memiliki problem yang lumayan besar dalam klasifikasi ikan. Dalam penelitian ini, penulis menggunakan metode *Viola-Jones Feature Extraction* dan *Boosting Berbasis Decision Tree* untuk melakukan klasifikasi ikan. Langkah pertama dalam metode ini adalah dengan melatih *weak classifier* untuk mengklasifikasi beberapa fitur beberapa kelas ikan, diikuti dengan *Boosting*, dan pembuatan sebuah *cascade* untuk mempercepat proses klasifikasi. Hasil dari proses ini adalah sebuah program yang dapat mengklasifikasi dan menganotasi ikan berdasarkan *input* gambar.

Kata kunci: *klasifikasi, Viola-Jones, Boosting, Decision Tree, Ikan, Cascade.*

I. PENDAHULUAN

Budidaya ikan di Indonesia memiliki problem yang lumayan besar yaitu diperlukannya usaha yang besar untuk menghitung dan mengawasi jumlah ikan yang dibudidayakan. Dalam penghitungan bibit pada proses jual beli contohnya, dalam menghitung bibit lele para pedagang masih menghitung ikan dengan metode menghitung satu-persatu yang memakan waktu lama, atau menggunakan bantuan timbangan yang tidak selalu akurat [1].

Untuk menyelesaikan permasalahan diatas, diperlukan alat yang dapat melakukan penghitungan ikan secara otomatis dan akurat. Al-Amri [1] menciptakan sebuah alat yang menghitung ikan menggunakan sensor *proximity*, dengan tingkat akurasi kurang lebih 96% dan kecepatan deteksi kurang lebih 4 ikan per detik. Sementara itu Rusydi [2] menciptakan sebuah alat yang menghitung ikan menggunakan konsep *through beam*, yang menghitung ikan sampai jumlah tertentu. Alat ini memiliki akurasi 100% dan dapat menghitung dengan kecepatan 58 ms per ikan.

Kedua alat yang diciptakan baik oleh Al-Amri [1] maupun Rusydi [2] bukanlah tanpa kekurangan. Keduanya hanya bisa bekerja dengan ikan yang berukuran tidak lebih besar dari alat yang digunakan. Alat milik Al-Amri harus dimodifikasi di bagian lubang keluar ikan untuk dapat mengamodasi ikan yang lebih besar. Sementara itu alat milik Rusydi sangat bergantung kepada ukuran ikan dan kelandaian pipa deteksi. Mengganti ukuran ikan yang akan dihitung mengharuskan pengguna untuk

mengkalibrasi ulang kelandaian dan ukuran pipa. Kedua alat tersebut sangat tidak fleksibel. Sangat tidak sesuai dengan industri budidaya ikan yang tidak hanya membudidayakan satu jenis ikan saja.

Berdasarkan latar yang telah dijelaskan, penulis akan menggunakan menggunakan metode *Viola-Jones Feature Extraction* dan *Boosting Berbasis Decision Tree* untuk mengkonstruksi sebuah *classifier* genus ikan yang nantinya dapat mengklasifikasi ikan dengan input gambar. Hasil yang diharapkan adalah sistem mampu mengklasifikasi ikan dari gambar secara akurat.

II. KAJIAN PUSTAKA

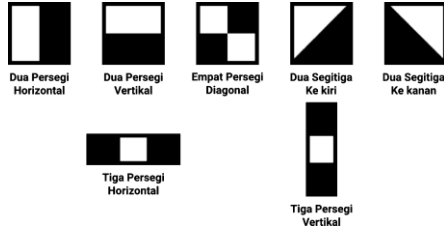
A. Pengertian Klasifikasi Objek

Fungsi dari klasifikasi objek adalah untuk memberikan deskripsi label ke sebuah segmentasi objek. Bila dilihat dari representasi fitur objek, ini dapat dicapai dengan melihat tanda-tanda keberadaan fitur yang dapat mengindikasikan kelas dari objek. Hal ini umumnya dicapai dengan menentukan *threshold* diantara kelas-kelas yang direpresentasikan oleh *training set* yang sudah dilabeli.

B. Viola Jones Object Detection Framework

Paul Viola dan Michael, J, Jones [3] mempublikasikan sebuah makalah ilmiah dengan judul “*Robust Real-Time Face Detection*”. Makalah tersebut mendeskripsikan sebuah *framework* pendeteksian wajah yang dapat memproses gambar secara cepat dengan tingkat akurasi yang tinggi. Ada tiga kontribusi penting dari makalah tersebut: Pertama adalah pengenalan sebuah representasi gambar baru yang dipanggil *Integral Image* yang memungkinkan penghitungan fitur yang dipakai oleh detektor dilakukan dengan cepat. Yang kedua adalah sebuah *classifier* yang efisien dan sederhana, yang dibuat menggunakan algoritma pembelajaran *Adaboost* [4] untuk memilih sejumlah fitur-fitur kritis dari fitur-fitur potensial. Yang ketiga adalah sebuah metode untuk menggabungkan fitur-fitur tersebut dalam sebuah bentuk *cascade* yang memungkinkan algoritma untuk memfokuskan deteksi di area-area yang memiliki potensial saja.

C. Features



Gambar 1. Fitur yang digunakan

Ada banyak alasan dimana penggunaan fitur lebih baik daripada nilai piksel secara langsung. Alasan paling umum adalah fitur dapat berfungsi untuk meng-*encode ad-hoc domain knowledge* yang sulit dipelajari dengan jumlah data pelatihan yang terbatas. Untuk sistem *framework Viola-Jones* ada alasan besar lainnya untuk penggunaan fitur yaitu sistem berbasis fitur beroperasi lebih cepat daripada sistem yang berbasis nilai piksel.

Fitur sederhana yang digunakan mirip dengan *Haar Basis Function* yang digunakan Papageorgiou dkk [5]. Lebih tepatnya ada tiga jenis fitur: dua persegi panjang, tiga persegi dan empat persegi diagonal. Nilai dari sebuah fitur adalah perbedaan antara jumlah nilai piksel didalam dua atau lebih area. Selain itu ada juga ditambahkan fitur dua segitiga. Fitur-fitur dapat dilihat pada Gambar 1.

D. Adaboost

Adaboost [4] digunakan untuk memperkuat performa klasifikasi dari *weak learner*. Adaboost melakukan ini dengan menggabungkan sekumpulan classifier lemah untuk membuat sebuah classifier kuat. Didalam istilah *Boosting*, *weak classifier* disebut juga dengan *weak learner*. Algoritma pembelajaran disebut lemah karena kita tidak berharap classifier terbaik untuk mengklasifikasi data dengan benar dengan tingkat akurasi lebih dari 51%.

Untuk mem-boost sebuah *weak classifier*, ia akan dipanggil untuk menyelesaikan sederet problem pembelajaran. Setelah satu ronde pembelajaran selesai, setiap contoh pembelajarannya akan dibobot ulang untuk menekankan problem yang salah diklasifikasi oleh *weak learner* tersebut. Problem yang sudah dibobot ulang ini nantinya akan dipakai untuk melatih *weak classifier* berikutnya. Bentuk final strong classifier adalah sebuah kombinasi *weak learner* yang sudah memiliki bobot *voting*.

Untuk klasifikasi multi-kelas perlu dilakukan modifikasi kecil terhadap algoritma boosting. Hal ini dengan menambahkan perhitungan kecil pada rumus penghitungan bobot, dengan memperhitungkan jumlah kelas yang akan diklasifikasi. Dimana e merupakan jumlah bobot salah klasifikasi dan N adalah jumlah kelas yang akan diklasifikasi.

$$\alpha = \frac{1}{2} \log \left(\frac{1 - e}{e} \right) + \log (N - 1) \quad (1)$$

E. Weak Learn

Framework Viola Jones menggunakan sebuah *weak classifier* yang bernama *Decision Stump*, atau sebuah *Decision Tree* yang hanya memiliki dua daun kelas saja. *Decision Tree* sendiri mampu digunakan untuk permasalahan multi-class. *Decision Tree* menghasilkan sebuah classifier didalam bentuk sebuah pohon pilihan, sebuah struktur yang berbentuk:

- Sebuah daun, mengindikasi sebuah kelas, atau
- Sebuah decision node yang menspesifikasi sebagian tes untuk dikerjakan atas sebuah nilai atribut, dengan satu branch dan subtree untuk setiap hasil dari tes.

Sebuah *Decision Tree* dapat digunakan untuk mengkasifikasi sebuah kasus dengan memulai tes dari akar pohon dan bergerak sampai sebuah daun kelas ditemukan. Pada setiap node yang bukan merupakan daun, hasil dari tes kasus dideterminasi dan tes dilanjutkan ke node berikutnya sesuai dengan hasil tes tersebut. Hingga proses pada akhirnya (dan dengan pasti) menuju ke sebuah daun kelas.

F. Attentional Cascade

Attentional Cascade [3] adalah sebuah *cascade* dari banyak *weak classifier* yang dibuat untuk meningkatkan performa deteksi dengan secara radikal mengurangi jumlah komputasi. Intinya *weak classifier* yang telah di-*boost* dapat dibuat lebih kecil dan efisien, yang dapat menolak mayoritas *sub-window* negatif dan mendeteksi sebagian besar dari *sub-window* positif. *Classifier* yang lebih sederhana digunakan untuk menolak mayoritas *sub-window* sebelum *classifier* yang lebih kompleks dipanggil untuk menurunkan tingkat *false positives*.

Cascade dimulai dengan membuat sebuah *stage* awal, dimana *weak classifier* ditambahkan secara perlahan hingga *detection rate* yang diinginkan tercapai. *Detection rate* ini harus disesuaikan oleh pengguna sesuai dengan keperluannya, tujuannya tetap untuk mengurangi waktu komputasi tapi juga mencoba agar tidak terlalu banyak kasus *false positive* yang dapat lewat.

III. METODE PENELITIAN

Metode penelitian dibagi menjadi dua tahap yaitu tahap *training* dan validasi hasil oleh pengguna.

Training dan validasi dilakukan menggunakan data yang diambil dari lapangan. Kelas ikan yang akan diklasifikasi ada tiga, yaitu ikan Emas, Lele dan Nila. Gambar ikan diambil dari hasil pembudidaya ikan lokal.

Training dilakukan di tiga bagian dari ikan: Pada bagian mulut ikan, sirip bagian atas dan ekor. Tujuannya adalah agar hasil dari ketiga *classifier* yang mengklasifikasi ketiga bagian tersebut bisa dibandingkan untuk menentukan kelas dari ikan pada gambar. Kelas yang mendapat hasil klasifikasi mayoritas dari ketiga *classifier* tersebut akan dipilih sebagai hasil klasifikasi akhir dan dianotasikan kedalam gambar hasil.

Training dibagi menjadi beberapa Langkah sebagai berikut:

1. *Pre-processing* data. Data diselaraskan ukruannya menjadi 350x200 piksel dan warnanya diubah menjadi *greyscale*.
2. Pembuatan 7 *features* kedalam sebuah *window* berukuran 50x50 piksel. Semua kemungkinan *feature* yang dapat masuk akan dibuat.
3. Penghitungan semua *features* pada semua gambar
4. Pembuatan *decision tree* untuk semua *features*
5. *Boosting* seluruh *decision tree*. Pada tahap ini juga dilakukan eliminasi *features* yang akurasi dibawah 50%
6. Pembuatan *cascade*

Pada tahap validasi digunakan metode pendeteksian *sliding window* untuk mencari bagian mulut, sirip dan ekor ikan pada gambar. Hal ini dilakukan karena posisi mulut, sirip dan ekor dari ikan pada tiap gambar bisa berada di posisi yang berbeda. Validasi diakhiri dengan menghitung tingkat *error* dari *classifier* yang sudah dibuat pada tahap *training*.

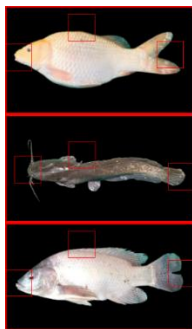
IV. HASIL DAN PEMBAHASAN

Pada tahap ini *training classifier* dimulai menggunakan total 36 gambar dengan 9 gambar merupakan gambar ikan Emas, 9 gambar ikan Lele, 9 gambar ikan Nila dan 9 gambar contoh negatif. Contoh negatif digunakan agar *classifier* dapat mengklasifikasi latar belakang kosong. Hasil dari *training* ini adalah sebuah file *pickle* yang bersisikan *object class cascade classifier* yang bisa digunakan untuk tahap validasi. Optimalnya akurasi yang diharapkan mendekati 100%.

1. Hasil Training

Training dilakukan dengan melatih tiga *classifier* pada posisi spesifik yang ditentukan pada setiap kelas. Hal ini dikarenakan untuk ketiga kelas ikan, lokasi mulut, sirip dan ekor mungkin berbeda satu dengan yang lainnya, atau karena fitur yang lebih kritis bisa didapatkan pada tempat yang berbeda. Gambar 2 menunjukan lokasi *offset* yang digunakan pada pelatihan *classifier* ditandai oleh persegi merah.

Features awal yang berhasil dibuat pada *window* berukuran 50x50 piksel berjumlah 520.000 dan untuk semuanya dibuatkan sebuah *decision tree* untuk melakukan klasifikasi. *Boosting decision tree* dilakukan untuk semua *decision tree* untuk setiap *window* (Kiri, Tengah dan Kanan). *Cascade* dibuat dari urutan *decision tree* yang sudah di-*boosting* pada tahap sebelumnya.



Gambar 2. Lokasi *offset training*

Tabel 1 menunjukan hasil dari *training* ketiga *classifier* untuk mulut, sirip dan ekor. Jumlah *weak classifier* akhir adalah sisa *weak classifier* yang memiliki akurasi lebih tinggi dari 50%. Akurasi didapat dari mengetes klasifikasi menggunakan sebagian dari gambar latihan.

Tabel 1 *classifier* hasil proses *training*

Classifier	Jumlah weak classifier	Akurasi saat training
Mulut	473	100%
Sirip	325	100%
Ekor	157	100%

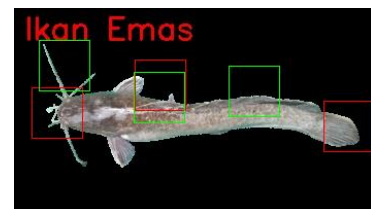
Bisa dilihat pada proses *training* ketiga *classifier* memiliki tingkat akurasi 100%. Yang berarti ketiganya sangat akurat dalam melakukan klasifikasinya masing-masing.

2. Hasil Validasi

Validasi menggunakan *sliding window* yang bergerak dari kiri atas ke kanan bawah untuk mencari lokasi mulut, sirip dan ekor pada gambar ikan.

Validasi langsung menggunakan *classifier* yang dibuat pada tahap *training* menghasilkan akurasi klasifikasi yang sangat rendah. Dari tiga gambar yang digunakan untuk tes klasifikasi, hanya satu yang berhasil diklasifikasikan secara benar, atau hanya 33% dari total gambar. Hal ini jauh dari akurasi semua *classifier* yang dites pada saat pelatihan.

Setelah ditelusuri, ada dua penyebab yang berkontribusi terhadap rendahnya tingkat akurasi dari *classifier*. Yang pertama adalah *sliding window* yang melakukan deteksi di lokasi yang tidak sesuai. Pada gambar 3 dapat dilihat ada dua jenis kotak, hijau dan merah. Hijau merupakan lokasi dimana *sliding window* berhasil mengklasifikasikan kelas ikan, dan merah adalah lokasi klasifikasi yang diharapkan karena merupakan lokasi yang dipakai untuk pelatihan. Pada contoh gambar ini *classifier* kiri dengan salah mengklasifikasikan "Ikan Emas", *classifier* tengah juga salah mengklasifikasikan "Ikan Emas", dan *classifier* kanan juga lagi-lagi salah dan mengklasifikasi "Ikan Nila". Karena dua dari tiga *classifier* mengklasifikasikan "Ikan Emas" maka gambar dianotasi sebagai gambar "Ikan Emas".



Gambar 3. Perbandingan lokasi klasifikasi validasi dan *training*

Salah lokasi klasifikasi ini dapat terjadi karena problem kedua yaitu bias *weak learner* terhadap salah satu kelas, dalam hal ini ke kelas "1" Ikan Emas. Bobot *voting* dari *weak learner* awal yang lebih tinggi dari bobot *voting weak learner*

setelahnya menyebabkan total nilai *voting* untuk kelas yang salah tidak bisa dikalahkan oleh bobot *voting* untuk kelas yang benar. Hal ini menyebabkan *classifier* mengeluarkan hasil positif di lokasi yang jelas-jelas tidak sesuai seperti pada gambar 3.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil implementasi yang telah dilakukan penulis serta pengujian fitur klasifikasi yang telah penulis rancang, maka didapatkan kesimpulan sebagai berikut:

1. Dibuatnya *classifier* genus ikan berbasis Viola-Jones Feature Extraction dan Boosting Berbasis Decision Tree yang menggunakan decision tree classifier sebagai klasifier yang digunakan, boosting sebagai metode pembobotan decision tree, dan attentional cascade yang digunakan untuk mengoptimalkan kerja classifier.
2. Berdasarkan hasil pengetesan akurasi yang dilakukan pada saat validasi, metode ini berhasil mengklasifikasi 1 dari 3 gambar, atau memiliki akurasi 33%. Akurasi ini sangat rendah bila ingin digunakan untuk mengklasifikasi secara konsisten.
3. Klasifikasi memiliki akurasi yang rendah dikarenakan dua faktor: tidak sesuainya lokasi klasifikasi dan juga bias bobot *voting weak classifier* awal pada *cascade*. Bias nampaknya ditunjukan kepada kelas ikan Abudehduf karena dari 50 hasil yang bukan kelas Abudehduf, 32-nya diklasifikasikan sebagai Abudehduf.

B. Saran

Adapun beberapa saran yang dapat penulis sampaikan untuk penelitian selanjutnya adalah:

1. Penelitian ini masih belum bisa secara akurat mengklasifikasi ikan. Oleh karena itu perlu ada penelitian lanjutan yang menyempurnakan tingkat akurasi klasifikasi dengan mengganti atau merombak metode yang ada, terutama metode *sliding window* dan boosting bobot *voting* yang memiliki bias klasifikasi ke kelas tertentu.
2. Pemutakhiran dapat dilakukan agar dapat menerima gambar dengan ukuran yang dinamis.
3. Pengimplementasian *multiprocessing* agar dapat mempercepat proses training yang memungkinkan melakukan training menggunakan data lebih besar.

DAFTAR PUSTAKA

- [1] Al-Amri, C. F. (2020). "Rancangan Bangun Fish Counter Untuk Menghitung Bibit Ikan Lele".
- [2] Rusydi, M. I. (2019). "Perancangan Mesin Penghitung Benih Ikan Otomatis untuk Membantu Kinerja Peternak Ikan".
- [3] Viola, P. and M. J. Jones (2004). "Robust Real-Time Face Detection". Tersedia pada: <https://doi.org/10.1023/B:VISI.0000013087.49260.fb>.
- [4] Freund, Y. and R. E. Schapire (1996). "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". Tersedia pada: <https://doi.org/10.1006/jcss.1997.1504>.
- [5] Papageorgiou, C. P., M. Oren, and T. Poggio (1998). "A General Framework for Object Detection". Tersedia pada: <https://doi.org/10.1109/ICCV.1998.710772>.