

DAFTAR ISI

LEMBAR PENGESAHAN	iii
LEMBAR PERNYATAAN	iv
KATA PENGANTAR	vi
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
II KAJIAN PUSTAKA	5
2.1 Pengertian Klasifikasi Objek	5
2.2 <i>Viola Jones Object Detection Framework</i>	5
2.2.1 <i>Features</i>	6
2.2.2 <i>Adaboost</i>	6
2.2.3 <i>Weaklearn</i>	7
2.2.4 <i>Attentional Cascade</i>	8
III METODOLOGI PENELITIAN	10
3.1 Tahapan Penelitian	10
3.2 Desain Sistem	10
3.3 <i>Training Strong Classifier</i>	11
3.3.1 <i>Input Dataset</i> Pelatihan	11
3.3.2 Pembuatan <i>Haar like Features</i>	13
3.3.3 Pembuatan <i>Decision Tree</i>	14
3.3.4 <i>Boosting</i>	15
3.3.5 Pembuatan <i>Attentional Cascade</i>	16
3.4 Skenario Eksperimen dan Validasi	17
3.4.1 <i>Pre-processing</i>	17

3.4.2	Klasifikasi	18
3.4.3	Anotasi	19
IV	HASIL DAN PEMBAHASAN	20
4.1	<i>Training Strong Classifier</i>	20
4.1.1	<i>Input Gambar dan labeling</i>	20
4.1.2	<i>Generate Haar-like Features</i>	21
4.1.3	<i>Calculating all features of all images</i>	22
4.1.4	<i>Create Decision Tree for each Feature</i>	26
4.1.5	<i>Boosting</i>	33
4.1.6	<i>Training Cascade</i>	38
4.2	Validasi	40
4.2.1	Validasi Tes Lapangan	43
4.3	Analisa Hasil	44
V	KESIMPULAN DAN SARAN	45
5.1	Kesimpulan	45
5.2	Saran	45
	DAFTAR PUSTAKA	47
A	<i>Source Code</i>	48
B	Tabel data <i>training</i>	49
C	Tabel pengolahan data gambar validasi	52
D	Gambar Pelatihan Tes Lapangan	60
E	Tabel pengolahan data gambar validasi tes lapangan	62
	DAFTAR RIWAYAT HIDUP	63

DAFTAR GAMBAR

Gambar 2.1	Beberapa <i>Haar-like features</i> yang digunakan framework Viola-Jones.	6
Gambar 2.2	<i>Workflow</i> dari <i>Attentional Cascade</i>	8
Gambar 3.1	Diagram alir untuk algoritma pelatihan klasifikasi objek . . .	10
Gambar 3.2	Diagram alir untuk algoritma klasifikasi objek	10
Gambar 3.3	Contoh gambar Abudehduf, Amphiprion, Chaetodon dan contoh gambar-gambar negatif	11
Gambar 3.4	<i>sub-window</i> setiap kelas ikan yang akan dipelajari oleh <i>classifier</i>	13
Gambar 3.5	Sebuah fitur dua persegi panjang menghadap ke kiri, lokasi x = 12 piksel, lokasi y = 10 piksel, dengan ukuran 12 x 12 piksel.	13
Gambar 3.6	Gambaran fitur-fitur yang akan digunakan	14
Gambar 3.7	Contoh sebuah <i>decision tree</i> dengan kelas 0, 1, 2 dan 3	14
Gambar 3.8	titik awal <i>sliding window</i> (kotak hijau) dan titik akhir (kotak kuning)	19
Gambar 3.9	Anotasi gambar ikan yang sudah diklasifikasi	19
Gambar 4.1	Gambar-gambar yang akan dipakai untuk training	20
Gambar 4.2	<i>source code: read</i> gambar, labelisasi, pengubahan ke <i>greyscale</i> , dan memastikan ukuran gambar 350 x 200 piksel .	20
Gambar 4.3	<i>source code: labelisasi</i> gambar sesuai dengan foldernya . . .	21
Gambar 4.4	<i>source code: load</i> gambar-gambar dari folder yang bersangkutan dan menggabungkannya	21
Gambar 4.5	<i>source code: sebuah data dalam features</i> memiliki semua informasi yang diperlukan untuk melakukan perhitungan nilai sebuah fitur. Tipe-tipe fitur akan menentukan rumus perhitungan fitur tersebut	22
Gambar 4.6	<i>source code: write_csv()</i> mengambil semua gambar, label, fitur dan mengkalkulasi semua fitur untuk ketiga <i>sub-window</i>	23
Gambar 4.7	<i>source code: class Dataset</i>	24
Gambar 4.8	<i>source code: offset</i> ini di-inisialisasi untuk setiap kelas Dataset sehingga bisa diakses langsung oleh fungsi <i>Find_Feature_Value</i>	25
Gambar 4.9	<i>source code: offset</i> ini di-inisialisasi untuk setiap kelas Dataset sehingga bisa diakses langsung oleh fungsi <i>Find_Feature_Value()</i>	26

Gambar 4.10	<i>source code: data training, data testing dan data validation disimpan kedalam array X_train, Y_train, X_test, Y_test, X_valid, Y_valid. Yang lalu disimpan kedalam splits</i>	27
Gambar 4.11	<i>source code: get data dan readcsv yang digunakan oleh split data</i>	27
Gambar 4.12	<i>source code: build_all_tree untuk membuat semua decision tree untuk setiap fitur</i>	28
Gambar 4.13	<i>source code: class node digunakan untuk menyimpan informasi cabang dan threshold pada node decision tree</i> . . .	28
Gambar 4.14	<i>source code: class digunakan untuk menyimpan tinggi maksimal dan minimal split pada decision tree. Semua data lainnya disimpan pada node</i>	29
Gambar 4.15	<i>source code: fungsi utama dari class DecisionTreeClassifier</i> .	29
Gambar 4.16	<i>source code: fungsi get_best_split</i>	30
Gambar 4.17	<i>source code: fungsi split hanya bertugas membagi node berdasarkan threshold yang sudah ditemukan</i>	30
Gambar 4.18	<i>source code: information_gain() mencari data dengan menghitung gini atau entropy</i>	31
Gambar 4.19	<i>source code: perhitungan entropy</i>	31
Gambar 4.20	<i>source code: perhitungan gini</i>	31
Gambar 4.21	<i>source code: fungsi untuk mencari mayoritas kelas pada leaf node</i>	31
Gambar 4.22	<i>source code: fungsi fit() adalah fungsi yang dipanggil untuk mulai membangun decision tree setelah dibuat</i>	32
Gambar 4.23	<i>source code: Predict() digunakan untuk melakukan prediksi dengan decision tree yang sudah dibuat</i>	32
Gambar 4.24	<i>source code: penyimpanan decision tree kedalam pickle</i> . . .	33
Gambar 4.25	<i>source code: training_strong_classifier</i>	34
Gambar 4.26	<i>source code: inisialisasi bobot gambar sebelum /emphBoosting</i>	35
Gambar 4.27	<i>source code: pengurutan weak classifier berdasarkan akurasi</i>	35
Gambar 4.28	<i>source code: pencarian nilai bobot boosting menggunakan fungsi start_boosting()</i>	36
Gambar 4.29	<i>source code: klasifikasi yang dilakukan setelah setiap iterasi boosting</i>	37
Gambar 4.30	<i>source code: bentuk class PickleTreeFinal</i>	37
Gambar 4.31	<i>source code: bentuk class Cascade</i>	38
Gambar 4.32	<i>source code: fungsi untuk mengisi stages pada Cascade</i> . .	39
Gambar 4.33	<i>source code: class CascadeStage</i>	39
Gambar 4.34	<i>source code: pelatihan stage dalam Cascade</i>	39
Gambar 4.35	<i>source code: fungsi penyimpanan Cascade ke pickle dengan menggunakan fungsi dump_to_pickle lagi</i>	40
Gambar 4.36	Gambaran cascade window 1 (kiri) dan cara kerjanya	40

Gambar 4.37	<i>source code:</i> predict.py untuk melakukan klasifikasi sebenarnya (bagian 1)	41
Gambar 4.38	<i>source code:</i> predict.py untuk melakukan klasifikasi sebenarnya (bagian 2)	42
Gambar 4.39	Ketiga kelas ikan	42
Gambar 4.40	Ketiga ikan yang akan diklasifikasi	43
Gambar 4.41	Gambar tes Amphiprion23. Kotak merah menunjukkan ekspektasi klasifikasi yang benar untuk kelas Amphiprion, karena <i>offset</i> diletakan disitu. Kotak hijau menunjukkan lokasi klasifikasi yang dilakukan <i>sliding window</i> ketika mengklasifikasi gambar Amphiprion23	44

DAFTAR TABEL

Tabel 4.1	Hasil boosting pada ketiga <i>window</i>	38
Tabel 2.1	Gambar-gambar yang digunakan untuk training	49
Tabel 3.1	Hasil prediksi menggunakan <i>cascade</i> dan <i>sliding window</i> . .	52
Tabel 4.1	Gambar-gambar yang digunakan untuk <i>training</i> tes lapangan	60
Tabel 5.1	Hasil prediksi menggunakan <i>cascade</i> dan <i>sliding window</i> . .	62