

KLASIFIKASI MULTILABEL MOTIF CITRA BATIK MENGUNAKAN BOOSTED RANDOM FERNS

M. Nur Fuad¹⁾ dan Nanik Suciati²⁾

^{1, 2)} Departemen Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember,
Kampus ITS, Sukolilo, Surabaya
e-mail: nurfuadmail@gmail.com¹⁾, nanik@if.its.ac.id²⁾

ABSTRAK

Penelitian terkait pengklasifikasian motif batik kebanyakan lebih fokus dalam mengenali satu jenis motif batik pada satu citra. Saat ini banyak terdapat citra batik yang memiliki lebih dari satu motif di dalamnya. Penelitian ini bertujuan untuk mengenali lebih dari satu motif batik dalam satu citra menggunakan metode ekstraksi fitur bentuk Histogram of Oriented Gradient (HOG) dikombinasikan dengan metode klasifikasi Boosted Random Ferns (BRF). Pada penelitian sebelumnya, kombinasi metode tersebut pada satu citra dapat melakukan identifikasi lebih dari satu pejalan kaki. Kehandalan kombinasi metode tersebut untuk melakukan identifikasi banyak objek pada satu label (pejalan kaki) akan dimodifikasi menjadi identifikasi banyak objek pada multilabel (berbagai motif batik). Untuk kasus pengenalan motif batik, sistem yang dikembangkan akan mengekstrak fitur HOG dari data citra training kemudian menyusunnya menjadi fitur ferns untuk membuat model-model klasifikasi motif batik. Model-model klasifikasi motif digunakan untuk identifikasi motif masing-masing pada citra testing. Uji coba dilakukan pada 64 data citra testing dengan 6 jenis motif batik. Performa metode diuji menggunakan sejumlah skenario pengujian dari variasi jumlah weak classifier, iterasi bootstrapping dan jumlah subset random fern. Digunakan empat variasi jumlah weak classifier yakni 100, 200, 300, dan 400, enam variasi iterasi bootstrapping yakni 0, 1, 2, 3, 4, dan 5 iterasi serta empat variasi jumlah subset random ferns yakni 5, 10, 15 dan 20 subset. Label-label hasil klasifikasi kemudian dikalkulasi dengan tanimoto distance. Diperoleh nilai 0.0130 sebagai nilai tanimoto distance terbaik, berdasarkan 62 citra yang dideteksi dengan benar dari 64 citra testing.

Kata Kunci: Boosted random ferns, histogram of oriented gradient, multilabel, klasifikasi, motif batik

ABSTRACT

The research related batik motif classification mostly more focus at recognize one batik motif in one image. Currently there are many images of batik that has more than one motive in it. This study aims to recognize many batik motifs in one image using Histogram of Oriented Gradient (HOG) form feature extraction method combined with the Boosted Random Ferns (BRF) classification method. In previous study, combination of these methods in one image was able to identify many pedestrians. These methods combination ability to identify multiobjects in a single label (pedestrian) was developed to identify multiobjects in multilabel (batik motifs). For batik motifs identifications case, developed system will extract the HOG feature from training image data and compiled it into ferns feature to create batik motif classification models. Each motif classification model is used to identify each motif in the testing image. Experiment conducted on 64 image testing data with 6 types of batik motifs. Proposed method's performance tested on some test scenarios based on variety weak classifier numbers, bootstrapping iterations and random ferns subsets number. There are four variety random subset ferns numbers ie 5, 10, 15 and 20, four variety weak classifier numbers ie 100, 200, 300 and 400, and six variety bootstrapping iterations ie 0, 1, 2, 3, 4, and 5. Then, labels from classification result be calculate using tanimoto distance. 0.0130 is the best tanimoto distance value, based on 62 image that correctly identified from 64 testing image.

Keywords: Batik motif, boosted random ferns, classification, histogram of oriented gradient, multilabel

I. PENDAHULUAN

BATIK adalah salah satu warisan budaya turun temurun dengan nilai estestika tinggi dan filosofi yang mendalam. Mulai dari batik khusus untuk selimut bayi dengan hiasan motif-motif dengan harapan memberi keberuntungan terhadap bayi hingga kafen batik khusus untuk jenazah. Batik juga umum digunakan menjadi baju keseharian, dalam kondisi santai atau resmi seperti ketika sekolah maupun kerja. Pakaian batik bahkan juga menjadi salah satu pakaian favorit yang digunakan dalam menghadiri acara pernikahan.

Kepopuleran penggunaan kain batik oleh masyarakat Indonesia adalah salah satu alasan yang membuat batik menjadi objek penelitian pada bidang pengolahan citra digital. Dalam pengolahan citra digital, analisa ciri motif citra batik diperoleh dari sisi bentuk, warna, atau tekstur. Berdasarkan ciri yang dianalisa diperoleh klasifikasi yang digunakan untuk meingidentifikasi jenis motif batik. Penelitian [1] melakukan klasifikasi batik dengan klasifikasi k-NN berdasarkan struktur dan warna motif batik dengan ekstraksi fitur statistik dari ciri orde satu dan dua. Kemudian [2] mengekstraksi ciri tekstur dari mean, energi dan standard deviasi menggunakan metode wavelet transform jenis daubechies 2 level 2 sebagai input motif batik yang diklasifikasi dengan metode fuzzy neural network. Selanjutnya [3] mengekstraksi fitur tekstur motif batik yang invarian terhadap rotasi

menggunakan metode Rotation Complete Robust Local Binary Pattern Magnitude (rotCRLBP_M) untuk dengan tetap mempertahankan ciri magnitude, kemudian fitur tersebut diklasifikasi menggunakan metode probabilistic neural network. Kemudian [4] mengidentifikasi motif batik berdasarkan fitur frekuensi tepi hasil deteksi tepi canny menggunakan metode learning vector quantization pada platform android.

Penelitian [1][2][3][4] mampu melakukan identifikasi terhadap satu jenis motif batik dalam satu citra, namun pada satu citra tidak dapat melakukan identifikasi lebih dari satu macam motif batik. Padahal pada beberapa kasus, pada satu citra batik terdapat lebih dari satu macam motif batik. Berdasarkan hal tersebut, dalam satu citra diperlukan sistem yang dapat melakukan identifikasi lebih dari satu macam motif batik. Identifikasi bentuk dapat digunakan sebagai salah satu solusi sebab meski berada dalam satu citra setiap bentuk motif batik unik antara satu sama lain, meski ada yang mirip. Sehingga dalam satu citra identifikasi lebih dari satu motif dapat dilakukan dengan memanfaatkan identifikasi bentuk. Histogram of Oriented Gradient (HOG) diajukan untuk mengekstraksi fitur bentuk. HOG telah dimanfaatkan pada identifikasi bentuk dalam penelitian identifikasi pejalan kaki [5] serta dalam penelitian [6] dan [7] pada bidang *image retrieval*.

Dalam perkembangan penggunaan fitur HOG, hasil perbandingan nilai bin dari HOG digunakan sebagai fitur *ferns* yang digunakan dalam klasifikasi Boosted Random Ferns (BRF) [8]. Klasifikasi BRF mampu melakukan identifikasi terhadap beberapa objek sekaligus dalam satu citra. Namun identifikasi yang dilakukan oleh BRF berada pada satu label yang sama. Misalnya jika terdapat beberapa label objek seperti objek pejalan kaki, mobil dan motor pada suatu citra. Pada kasus tersebut hanya satu jenis objek saja yang akan teridentifikasi, seperti pejalan kaki saja, mobil saja atau motor saja. Oleh karena itu agar BRF mampu melakukan identifikasi multiobjek terhadap beberapa jenis label (multilabel) dalam satu citra, pembagian dataset dan cara identifikasi klasifikasi BRF perlu dimodifikasi.

Selanjutnya dalam bidang multilabel telah dilakukan penelitian oleh [9] yang menggunakan klasifikasi kansei multilabel terhadap citra batik. Penelitian [9] menggunakan bentuk, tekstur dan jenis warna motif batik sebagai variasi multilabel kansei. Pada penelitian [9] dilakukan ekstraksi fitur bentuk, tekstur dan warna dari setiap motif batik untuk menentukan pelabelan kansei.

Sedangkan dalam penelitian ini menggunakan klasifikasi BRF untuk melakukan identifikasi berbagai jenis motif batik dalam satu citra. Kemudian dilakukan pelabelan jenis motif batik pada citra berdasarkan hasil identifikasi. Selanjutnya label-label yang telah didapatkan, diuji akurasi menggunakan *tanimoto distance*.

Penelitian ini bertujuan untuk mengenali berbagai macam motif pada citra batik dengan klasifikasi multilabel motif citra batik. Metode yang diajukan adalah BRF untuk menyusun klasifikasi multilabel motif batik, dan metode HOG digunakan untuk mendapatkan ekstraksi bentuk. Penelitian ini berkontribusi untuk menghasilkan sebuah sistem dengan kemampuan mengklasifikasi macam-macam motif batik dalam citra batik menggunakan klasifikasi multilabel dengan metode BRF. Penelitian ini tersusun sebagai berikut, bab 1 berisi latar belakang dan penelitian terkait, bab 2 berisi penjelasan mengenai metode penelitian, bab 3 berisi skenario pengujian, bab 4 berisi hasil dari pengujian dan bab 5 berisi kesimpulan penelitian yang dilakukan.

II. METODE PENELITIAN

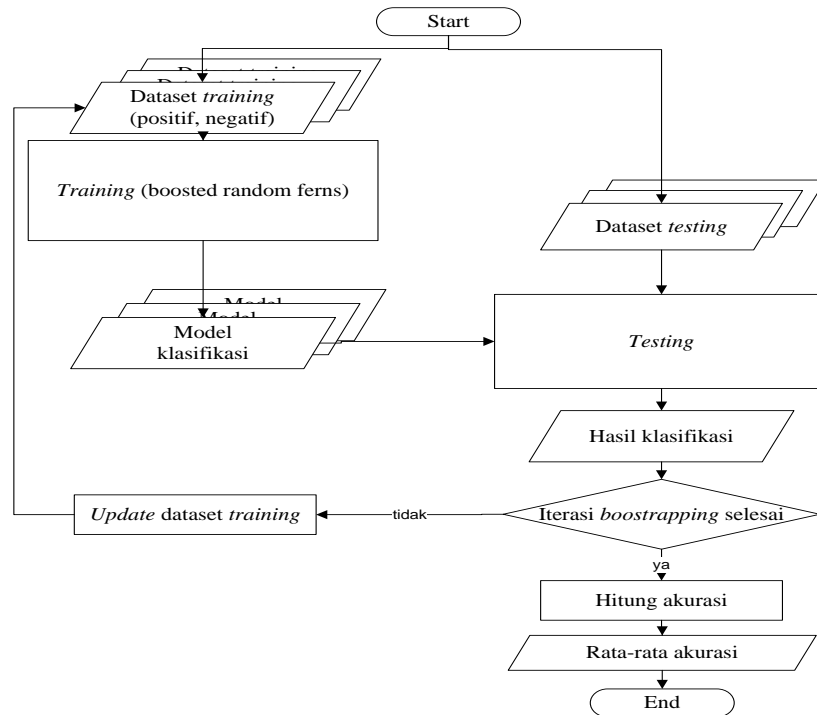
Pada bab ini dijelaskan mengenai desain sistem yang dibangun beserta proses-prosesnya. Proses *training* dan *testing* menjadi dua proses inti dalam desain sistem. Penyusunan model klasifikasi dilakukan dalam proses *training*. Kemudian performa model klasifikasi dievaluasi melalui proses *testing*. Pada proses *training* dan proses *testing*, dataset *training* dan *testing* yang digunakan mencakup 6 jenis citra motif batik klasik, yaitu: motif parang, kawung, lunglungan, megamendung, nitik dan semen.

Dalam desain sistem, secara umum *input* yang digunakan terbagi menjadi dataset *training* dan dataset *testing*. Dataset *training* digunakan pada proses *training* untuk menyusun model klasifikasi motif batik. Dataset *training* terbagi dalam dataset positif dan negatif. Enam model klasifikasi motif batik yang masing-masing mewakili satu dari enam jenis motif batik menjadi *output* proses *training*. Selanjutnya *input* dari proses *testing* adalah enam model klasifikasi motif batik hasil dari *output* proses *training* dan dataset *testing*. Hasil *output* proses *testing* adalah label-label motif batik dan *bounding box* hasil klasifikasi dari semua model klasifikasi motif batik.

Selanjutnya ditambahkan proses *boostapping* untuk memperbaiki dataset *training* yang digunakan. Jumlah proses *boostapping* yang digunakan adalah 5 iterasi. Dalam setiap iterasi proses *boostapping*, dilakukan penambahan dataset positif maupun negatif dalam dataset *training*. Proses *training* dan proses *testing* juga dilakukan dalam setiap iterasi *boostapping*. Akurasi hasil output proses *testing* dalam masing-masing iterasi disimpan dan dihitung akurasi. Rata-rata akurasi kemudian ditampilkan sebagai *output* akhir. Alur desain sistem dari penelitian yang dilakukan ditampilkan dalam *flowchart* pada Gambar 1.

A. Training

Penyusunan model-model klasifikasi motif batik menjadi tujuan utama dalam proses *training*. Model-model



Gambar 1. Flowchart desain sistem klasifikasi multilabel boosted random ferns

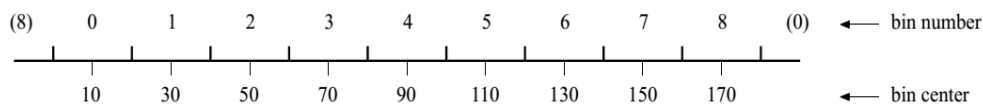
tersebut berfungsi untuk mengidentifikasi motif batik pada proses *testing*. Input proses *training* adalah data *training* berukuran 50 x 50 piksel. Dataset *training* berjumlah 340 citra yang terbagi ke dalam 120 citra background, 30 citra dengan motif semen, 40 citra dengan motif parang, 30 citra dengan motif lunglungan, 40 citra dengan motif kawung, 30 citra motif dengan megamendung, dan 50 citra dengan motif nitik. Perbedaan jumlah variasi setiap motif mengakibatkan perbedaan jumlah citra dalam setiap dataset motif terjadi karena (3 variasi motif semen, 4 variasi motif parang, 3 variasi motif lunglungan, 4 variasi motif kawung, 3 variasi motif megamendung dan 5 variasi motif nitik).

Semua citra pada dataset motif semen, parang, lunglungan, kawung, megamendung, dan nitik kemudian dipilah menjadi dataset positif dan negatif. Contoh hasil pemilahan dataset *training* ditampilkan dalam Tabel I. Citra objek yang mengandung motif batik sesuai label motif dimasukkan ke dalam dataset positif. Sebaliknya citra motif batik dengan label motif yang tidak sesuai dan citra background dimasukkan ke dalam dataset negatif.

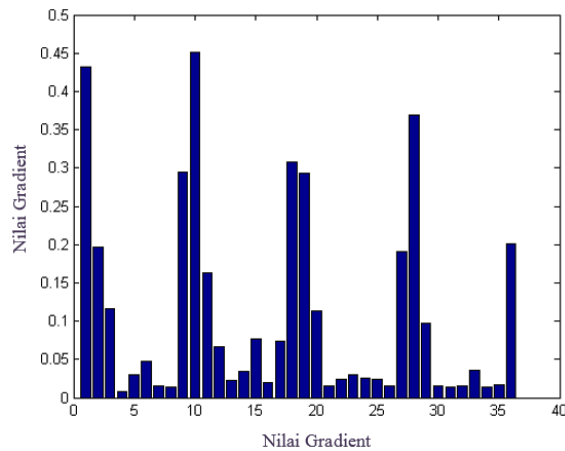
Dataset *training* untuk masing-masing model klasifikasi motif batik diperoleh dari pemilahan dataset. Dataset *training* positif dan negatif setiap model klasifikasi motif batik harus disusun secara berbeda. Klasifikasi yang berbeda-beda dari model-model klasifikasi motif menyebabkan perbedaan susunan dataset. Identifikasi motif

TABEL I.
TABEL CONTOH HASIL PEMILAHAN DATASET TRAINING

No	Motif	Positif	Negatif
1	Parang		
2	Lunglungan		
3	Kawung		
4	Megamendung		
5	Semen		
6	Nitik		



Gambar 2. Rentang bin HOG [11]



Gambar 3. Histogram of Oriented Gradient

lunglungan hanya akan dilakukan oleh model klasifikasi motif lunglungan. Sebaliknya identifikasi motif lainnya juga hanya dilakukan oleh model klasifikasi motif yang lain sesuai motif masing-masing. Model klasifikasi motif lunglungan memiliki dataset positif yang hanya terdiri dari dataset motif lunglungan, sebaliknya dataset background, dataset motif semen, parang, kawung, megamendung, dan nitik, dimasukkan ke dalam dataset negatif model klasifikasi lunglungan.

Selanjutnya setelah dataset *training* dipilih digunakan dalam penyusunan model klasifikasi motif. Terdapat tiga tahap pembuatan model klasifikasi motif, yaitu penyusunan *weak classifier*, perubahan bobot *weak classifier* serta penyusunan strong classifier. Penyusunan *weak classifier* dilakukan dengan membuat model klasifikasi simpel. Kemudian apabila pada identifikasi data *training* terdapat error, dilakukan perubahan bobot terhadap data *training* yang dilakukan oleh *Weak classifier*. Bobotnya akan ditambah jika terdapat data yang salah dalam identifikasi. Sebaliknya bobotnya dikurangi jika data benar dalam identifikasi. Agar data yang salah dalam identifikasi memperoleh prioritas lebih sehingga dididentifikasi secara benar dalam identifikasi selanjutnya, perubahan bobot perlu dilakukan.

Dasar dari pembuatan model klasifikasi adalah penyusunan *weak classifiers*. *Histogram of oriented gradient* (HOG) memiliki ruang bin yang digunakan sebagai ruang perhitungan klasifikasi *random ferns* oleh *weak classifier* [10]. Setiap *weak classifier* diproses dalam iterasi t untuk menyusun *strong classifier*. Jumlah iterasi t adalah jumlah *weak classifier* yang digunakan, misal terdapat 100 *weak classifier*, maka jumlah iterasi t juga 100. Pada setiap iterasi t , dilakukan pemilihan beberapa posisi piksel dalam citra secara *random*. Selanjutnya dari citra yang berpusat pada beberapa posisi yang telah dipilih dilakukan operasi *local HOG* pada region untuk memperoleh fitur HOG.

Fitur HOG diperoleh dengan menghitung nilai gradien dan orientasi gradien pada *region* citra berukuran 16x16 piksel. Kemudian region tersebut dibagi menjadi 4 region kecil yang disebut *cell*, masing-masing *cell* memiliki ukuran yang sama (8x8 piksel). Dalam setiap *cell*, nilai gradien dimasukkan pada bin HOG sesuai nilai orientasi gradiennya. Dalam setiap *cell* terdapat 8 bin HOG, masing-masing bin memiliki rentang 30°. Fitur HOG adalah vektor berukuran 1x36 bin yang terdiri dari gabungan 4 fitur *cell* bin yang dinormalisasi. Setiap *cell* memiliki vektor berukuran 1x8 bin. Tampilan rentang bin ditampilkan pada Gambar 2, sedangkan contoh nilai gradien dalam HOG ditampilkan dalam Gambar 3.

Dari vektor HOG berukuran 1x36, dipilih 14 bin HOG secara *random* untuk dibandingkan. Dari 14 bin tersebut, 7 bin awal dibandingkan dengan 7 bin akhir, jika lebih kecil akan memiliki nilai 0 dan jika lebih besar akan memiliki nilai 1. Hasil perbandingan tersebut digunakan sebagai nilai fitur *ferns*. Selanjutnya nilai fitur *ferns* diklasifikasikan dengan menggunakan *weak classifier*.

Weak classifier (h_t) pada penelitian yang digunakan adalah klasifikasi *random ferns* yang diimplementasikan dalam *adaboost* [12]. Disebut h_t karena *weak classifier* ini didapatkan dalam iterasi t . *Weak classifier* kemudian melakukan perkalian $\frac{1}{2}$ log terhadap nilai hasil klasifikasi *ferns*. Sedangkan klasifikasi *ferns* sendiri menghitung probabilitas jumlah kemunculan fitur *ferns* pada kelas motif dibagi dengan probabilitas jumlah kemunculan fitur *ferns* pada kelas background. Kemudian untuk menghindari pembagian dengan nilai 0, maka ditambahkan 1 kemunculan pada masing-masing probabilitas (pada kelas motif dan pada kelas background). Perhitungan *weak classifier* ditampilkan pada persamaan (1).

$$h_t(x) = \frac{1}{2} \log \frac{P(f(x, u_t, \theta_t)|O)+\epsilon}{P(f(x, u_t, \theta_t)|B)+\epsilon} \quad (1)$$

Pada persamaan (1), $P(f(x, u_t, \theta_t)|O)$ adalah probabilitas jumlah kemunculan fitur *ferns* pada kelas objek (motif), sedangkan $P(f(x, u_t, \theta_t)|B)$ adalah probabilitas jumlah kemunculan fitur *ferns* pada kelas background. Variabel x adalah citra objek yang diidentifikasi, variabel u_t adalah posisi pusat dari subset *ferns* dalam *weak classifier t*, dan variabel θ_t adalah 7 pasang bin HOG yang dibandingkan dalam *weak classifier t*. Jika hasil *output weak classifier* bernilai negatif maka dinyatakan sebagai *background*, sedangkan jika bernilai positif, maka dinyatakan sebagai objek (motif).

Selanjutnya karena dalam setiap iterasi t dipilih 5 posisi subset *ferns* secara random, maka diperoleh 5 *weak classifier* yang masing-masing memiliki posisi subset *ferns* yang berbeda. Kemudian dalam setiap iterasi t hanya dipilih satu *weak classifier* yang paling diskriminan, sehingga dibutuhkan *bhattacharya distance* untuk menghitung nilai diskriminan dari masing-masing komponen *weak classifier*. Perhitungan *bhattacharya distance* ditampilkan pada persamaan (2).

$$Q(u_t, \theta_t) = 2 \sum_{z=0}^{2^M-1} \sqrt{P(z|O)P(z|B)} \quad (2)$$

Pada persamaan (2), $Q(u_t, \theta_t)$ adalah nilai *bhattacharya*, sedangkan M adalah jumlah pasangan fitur *ferns*, sehingga $2^M - 1$ adalah $2^7 - 1$. Kemudian $P(z|O)$ adalah probabilitas kemunculan nilai z dalam kelas motif dikali bobot data *training* x_i , sedangkan $P(z|B)$ adalah probabilitas kemunculan nilai z dalam kelas background dikali bobot data *training* x_i . Bobot nilai z pada iterasi t pertama adalah 1/semua sampel *training*. Setelah didapatkan nilai *bhattacharya distance* dari komponen masing-masing *weak classifier*, maka *weak classifier* yang memiliki komponen dengan nilai *bhattacharya distance* terkecil dipilih menjadi *weak classifier* pada iterasi t . Pada *bhattacharya distance*, jika nilai semakin mendekati 0 maka semakin diskriminan.

Selanjutnya dilakukan perubahan bobot berdasarkan *weak classifier* yang terpilih dalam iterasi t saat ini. Perhitungan perubahan ditampilkan pada persamaan (3).

$$W_{t+1}(i) = \frac{W_t(x_i) \exp(y_i h_t(x_i))}{\sum_{i=1}^N W_t(x_i) \exp(y_i h_t(x_i))} \quad (3)$$

Pada persamaan (3), w adalah bobot dan w_{t+1} merupakan bobot untuk iterasi t berikutnya. Kemudian y_i adalah label hasil klasifikasi *weak classifier* pada iterasi t . Label + menandai citra x_i sebagai motif sedangkan label - menandai citra x_i sebagai background. Nilai bobot ini akan digunakan sebagai bobot citra x_i pada iterasi t berikutnya.

Algoritma 1. Boosted Random Ferns

Input : dataset *training* dari kelas O , memiliki label $\{+1, -1\}$ yang digunakan sebagai indikasi sampel adalah kelas O atau *background B*

Output : Model klasifikasi objek H_{C_j}

- $\theta = \{F_1, \dots, F_M\}$: adalah sekumpulan *random ferns*.
- T : adalah jumlah *weak classifiers* yang dipakai.

1. Lakukan inisiasi distribusi bobot terhadap sampel *training* ke $W_1(x_i) = \frac{1}{N}$, di mana $i = 1, 2, \dots, N$.
2. Buat *list* posisi subset random *ferns* u_l sejumlah L subset *ferns* yang digunakan.
3. Buat *list* θ sebanyak R dengan isi pasangan-pasangan bin HOG.
4. **For** $t = 1$ sampai T **do**
5. **For** $l = 1$ sampai L **do**
6. **For** $r = 1$ sampai R **do**
7. Gunakan distribusi weight saat ini (W_t), pasangan bin HOG dalam θ_r , dan posisi citra u_l dalam perhitungan *bhattacharyya distace* Q .
8. **End**
9. **End**
10. *Weak classifier* dengan kombinasi θ_r dan posisi citra u_l yang memiliki nilai *bhattacharyya distace* Q terkecil menjadi *weak classifier* h_t .
11. h_t digunakan dalam perubahan distribusi weight W^{t+1} .
12. **End**
13. Dari semua *weak classifier-weak classifier t*, lakukan penyusunan *strong classifier*. Semua kombinasi pasangan bin HOG θ_r dengan posisi citra u_l yang memiliki nilai *Bhattacharyya distace* yang paling mendekati 0 pada setiap iterasi t disimpan untuk digunakan dalam proses *testing*.

Setelah iterasi t berakhir, *weak classifier* dari setiap iterasi t digabung menjadi *strong classifier*. Semua klasifikasi *weak classifier* divoting untuk mendapatkan klasifikasi *strong classifier*. *Strong classifier* selanjutnya disebut model klasifikasi motif. Karena digunakan 6 jenis motif batik, maka dibutuhkan 6 *strong classifier* untuk menjadi 6 model klasifikasi motif. Keseluruhan poses pembangunan *weak classifier*, perubahan *weak classifier* dan pembangunan *strong classifier* ditampilkan pada Algoritma 1.

B. Testing

Pada proses *testing*, digunakan *input* dataset *testing* motif batik. Setiap citra pada dataset *testing* motif batik mempunyai satu atau beberapa motif batik. Digunakan 64 citra batik sebagai dataset *testing* yang terdiri dari 22 citra motif kombinasi dan 42 citra motif satuan. Citra motif kombinasi adalah citra yang mengandung lebih dari satu motif batik dalam setiap citra, citra ini antara lain adalah 3 citra kombinasi semen lunglungan, 2 citra kombinasi parang-kawung-lunglungan, 7 citra kombinasi parang-nitik, 4 citra kombinasi parang-megamendung, 3 citra kombinasi kawung-nitik, dan 3 citra kombinasi parang-semen. Sedangkan citra motif satuan adalah citra yang mengandung satu motif batik saja dalam setiap citra, citra ini terdiri dari 6 citra nitik, 6 citra parang, 6 citra lunglungan, 9 citra kawung, 9 citra megamendung, serta 6 citra semen. Gambar 4 menampilkan contoh citra *testing* motif kombinasi, dan Gambar 5 menampilkan contoh data *testing* motif satuan.

Untuk mengetahui kebenaran dari hasil klasifikasi terhadap data *testing*, setiap citra pada dataset *testing* diberi *ground truth* dalam bentuk *bounding box* dalam semua citra *testing*. Dikarenakan proses klasifikasi masing-masing model klasifikasi motif batik dikerjakan terpisah, *ground truth* untuk masing-masing motif juga dipisah. Gambar 6 menampilkan contoh tampilan *ground truth* secara terpisah, dengan bagian a adalah *ground truth* (ditandai dengan kotak biru) dari motif semen sedangkan bagian b adalah *ground truth* dari motif parang.

Selanjutnya dilakukan *testing* terhadap citra *testing* menggunakan model-model klasifikasi motif yang mana masing-masing berfungsi untuk mengidentifikasi motif sesuai modelnya. Sebagai contoh, misalkan model klasifikasi motif lunglungan berfungsi untuk mengidentifikasi motif lunglungan sedangkan model klasifikasi motif semen berfungsi untuk mengidentifikasi motif semen. Begitu pula halnya dengan model klasifikasi motif yang lain. Masing-masing model klasifikasi motif akan mengidentifikasi motif dan *background*.



Gambar 4. Contoh motif batik kombinasi



Gambar 5. Contoh motif batik satuan



Gambar 6. Contoh *ground truth*.(a) *Ground truth* motif semen. (b) *Ground truth* motif parang.

TABEL II
CONTOH HASIL PREDIKSI

Nama Citra	Hasil <i>testing</i> Label hasil klasifikasi	
	(P_j)	(C_j)
Batik 1	kawung lunglungan megamendung	kawung megamendung
Batik 2	lunglungan	lunglungan

Background dalam proses ini merupakan region-region citra yang bukan termasuk motif yang bersesuaian dengan modelnya. Yang termasuk *background* juga yaitu motif-motif lain yang tidak sesuai model klasifikasi motif. Sebagai contoh, jika menggunakan klasifikasi motif lunglungan, yang dianggap sebagai objek motif hanyalah motif lunglungan. Sehingga region citra yang tidak termasuk motif lunglungan atau bukan motif apapun, dianggap sebagai *background*. Model klasifikasi motif memiliki *output* berupa hasil identifikasi motif pada *region* citra.

Proses identifikasi model-model klasifikasi motif dilakukan pada *sliding window* dalam citra. Semua *region* citra yang berada di dalam *sliding window* selanjutnya diidentifikasi apakah merupakan motif atau *background*. *Sliding window* yang digunakan mempunyai ukuran piksel 50x50. Pergeseran *sliding window* dilakukan dari posisi piksel kiri atas ke kanan bawah dalam citra *testing*. Kemudian secara overlap pada *sliding window* dilakukan pergeseran dengan jarak setiap 4 piksel.

Sliding window membentuk *region* citra yang akan diidentifikasi motif batik menggunakan masing-masing model klasifikasi motif. Model klasifikasi motif akan menjalankan perhitungan masing-masing *weak classifier-weak classifier*. Kemudian dilakukan voting terhadap hasil *weak classifier-weak classifier* untuk menentukan apakah terdapat motif batik pada *region* tersebut.

Ketika motif batik ditemukan dalam *sliding window* sesuai model klasifikasi motif, *sliding window* diberi warna kuning, posisinya disimpan dan citra diberi label motif. *Sliding window* dengan warna kuning kemudian disebut sebagai *bounding box*. *Bounding box* dari *sliding window* selanjutnya dipadankan terhadap *bounding box ground truth*. Ketika posisi *sliding window* ada di piksel sebelah kanan bawah dari citra, pergeseran *sliding window* dilakukan pada citra selanjutnya. *Output* model klasifikasi motif masing-masing citra terbagi menjadi dua opsi. Satu, motif berhasil diidentifikasi, *outputnya* adalah *bounding box* dengan isi motif. Dua, motif tidak berhasil diidentifikasi, *outputnya* adalah kosong.

Selanjutnya dilakukan proses *bootstrapping* dari hasil identifikasi model-model klasifikasi motif. *Bootstrapping* digunakan untuk menambahkan dataset baru ke dalam dataset *training* [8]. Dataset baru ini didapatkan dengan cara menambahkan *region* citra berlabel *false* positif ke dalam dataset *training* negatif. Juga dengan menambahkan *region* citra berlabel tidak terdeteksi ke dataset *training* positif. *Region* citra berlabel *false* positif adalah hasil identifikasi model klasifikasi motif yang tidak cocok terhadap *ground truth*. Sedangkan yang berlabel tidak terdeteksi, adalah *ground truth* dengan keterangan tidak teridentifikasi sebagai motif oleh model klasifikasi motif.

Kemudian label-label hasil klasifikasi model-model klasifikasi motif diuji menggunakan *tanimoto distance* yang biasa digunakan menguji klasifikasi multilabel. Jumlah kategori yang berhasil diidentifikasi dengan benar oleh algoritma klasifikasi multilabel dapat dievaluasi menggunakan *tanimoto distance* [13]. Persamaan *tanimoto distance* ($tanimoto_j$) ditampilkan pada persamaan (4).

$$tanimoto_j = \frac{|P_j| + |C_j| - 2|P_j \cap C_j|}{|P_j| + |C_j| - |P_j \cap C_j|} \quad (4)$$

Pada persamaan (4), $P_j \cap C_j$ adalah jumlah kategori yang sama dari (P_j) dengan (C_j). (P_j) adalah jumlah kategori yang dihasilkan klasifikasi model klasifikasi motif, dan (C_j) adalah jumlah kategori sesungguhnya. Jika P_j dan C_j memiliki jumlah sama, yang terjadi adalah *tanimoto_j* menjadi bernilai 0 (nilai persamaan tertinggi). Tabel II menampilkan contoh *tanimoto distance*.

Citra Batik 1 pada Tabel II memiliki $P_j=3$ (kawung, lunglungan, megamendung) dan $C_j=3$ (kawung, megamendung) sedangkan irisan $P_j \cap C_j=2$ (kawung, megamendung). Berdasarkan nilai P_j , C_j dan $P_j \cap C_j$ maka nilai $tanimoto_j$ dari citra Batik 1 adalah sebagai berikut :

$$tanimoto_j \text{ citra Batik 1} = \frac{3 + 2 - 2(2)}{3 + 2 - 2} = 0.33$$

Sedangkan untuk citra Batik 2 memiliki $P_j=1$ (lunglungan) dan $C_j=1$ (lunglungan) sedangkan irisan $P_j \cap C_j=1$ (lunglungan). Berdasarkan P_j , C_j dan $P_j \cap C_j$ maka nilai $tanimoto_j$ dari citra Batik 2 adalah sebagai berikut :

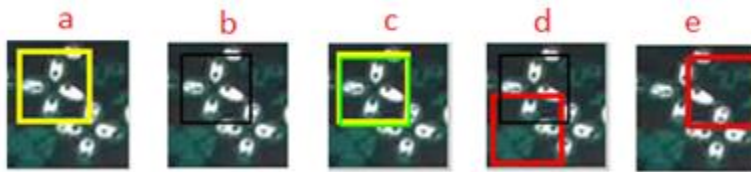
$$tanimoto_j \text{ citra batik 2} = \frac{1 + 1 - 2(1)}{1 + 1 - 1} = 0$$

C. Skenario Pengujian

Skenario pengujian digunakan untuk menguji hipotesa penelitian dan performa sistem. Pada skenario pengujian, variasi jumlah subset *ferns* yang digunakan ada empat yakni 5, 10, 15 dan 20. Kemudian variasi jumlah *weak classifier* yang digunakan juga empat yakni 100, 200, 300 dan 400. Selanjutnya variasi iterasi *boosttrapping* yang digunakan ada enam yakni iterasi 0, 1, 2, 3, 4, dan 5. Skenario pengujian terbagi ke dalam dua jenis pengujian, yakni pengujian data *testing* dan pengujian *boosttrapping*. Dalam pengujian data *testing*, semua variasi parameter jumlah subset *ferns* dikombinasikan dengan semua variasi jumlah *weak classifier*. Kombinasi tersebut menghasilkan 16 skenario pengujian. Sedangkan dalam pengujian *boosttrapping*, hanya digunakan satu variasi jumlah subset *ferns* yakni 5 dan satu variasi jumlah *weak classifier* yakni 100, yang kemudian dikombinasi dengan 6 variasi iterasi *boosttrapping*. Kombinasi tersebut menghasilkan 6 skenario pengujian.

III. HASIL DAN PEMBAHASAN

Hasil pengujian data citra *testing* beserta pembahasan dari pengujian yang dilakukan diterangkan pada bab ini. Pengujian terhadap data *testing* dilakukan untuk membuktikan kemampuan dari klasifikasi multilabel motif batik dari sistem yang dibangun beserta akurasinya. Pengujian dilakukan terhadap masing-masing data *testing* yang diberi label-label motif batik hasil dari klasifikasi sistem. *Tanimoto distance* digunakan sebagai penghitung akurasi multilabel dimana nilai persamaan akan tinggi jika nilainya mendekati 0 (jika nilainya 0, maka label yang dibandingkan adalah sama). Parameter pengujian terhadap data *testing* adalah sebagai berikut jumlah file benar,



Gambar 7. Ilustrasi hasil klasifikasi motif. (a) *Bounding box default* hasil dari klasifikasi motif. (b) *Bounding box default ground truth* tanpa hasil klasifikasi. (c) *Bounding box* hasil dari klasifikasi motif beririsan terhadap *bounding box ground truth* lebih dari 50%, *bounding box ground truth* berubah warna jadi hijau. (d) *Bounding box* hasil dari klasifikasi motif beririsan terhadap *bounding box ground truth* lebih dari 50%, *bounding box* hasil klasifikasi motif berubah warna jadi merah. (e) *Bounding box* hasil klasifikasi motif tanpa *ground truth*, *bounding box* hasil klasifikasi motif berubah warna jadi merah.

TABEL III
CONTOH PELABELAN HASIL DETEKSI MOTIF

No	Hasil tampilan <i>testing</i>	Label hasil klasifikasi	Label sesungguhnya	Nilai <i>tanimoto distance</i>
1		-	semen	1
2		kawung-nitik	kawung	0.5
3		semen-nitik	semen	0.5
4		semen	semen	0
5		parang-nitik	parang-nitik	0
6		kawung	kawung	0

TABEL IV
PENGUJIAN PADA DATA TESTING MENGGUNAKAN 100 WEAK CLASSIFIER

Jumlah Ferns	Jumlah citra benar	Jumlah citra salah	Rata-rata nilai tanimoto	Waktu testing (detik)	Waktu training (detik)
5	57	7	0.068	1788	121
10	55	9	0.082	1762	156
15	55	9	0.073	1772	191
20	59	5	0.060	1771	225

TABEL V
PENGUJIAN PADA DATA TESTING MENGGUNAKAN 200 WEAK CLASSIFIER

Jumlah Ferns	Jumlah citra benar	Jumlah citra salah	Rata-rata nilai tanimoto	Waktu testing (detik)	Waktu training (detik)
5	60	4	0.047	3545	245
10	58	6	0.060	3778	341
15	59	5	0.052	3814	410
20	58	6	0.060	3495	480

TABEL VI
PENGUJIAN PADA DATA TESTING MENGGUNAKAN 300 WEAK CLASSIFIER

Jumlah Ferns	Jumlah citra benar	Jumlah citra salah	Rata-rata nilai tanimoto	Waktu testing (detik)	Waktu training (detik)
5	60	4	0.047	5217	332
10	58	6	0.068	5181	403
15	60	4	0.047	5489	518
20	59	5	0.052	5470	628

TABEL VII
PENGUJIAN PADA DATA TESTING MENGGUNAKAN 400 WEAK CLASSIFIER

Jumlah Ferns	Jumlah citra benar	Jumlah citra salah	Rata-rata nilai tanimoto	Waktu testing (detik)	Waktu training (detik)
5	59	5	0.052	8586	395
10	59	5	0.055	8332	597
15	59	5	0.055	7894	740
20	59	5	0.060	6663	866

TABEL VIII
HASIL PENGUJIAN BOOSTRAPPING

Jumlah iterasi	Jumlah file benar	Jumlah file salah	Rata-rata nilai tanimoto	Waktu testing (detik)	Waktu training (detik)
0	57	7	0.068	1788	121
1	57	7	0.052	2010	119
2	62	2	0.013	2025	128
3	59	5	0.047	2032	135
4	60	4	0.047	2033	144
5	61	3	0.023	2010	148

jumlah file salah, jumlah rata-rata nilai tanimoto distance, waktu untuk melakukan *testing* dan waktu untuk melakukan *training*.

Pada pengujian data *testing*, identifikasi motif dalam masing-masing citra *testing* dilakukan menggunakan model-model klasifikasi dengan memanfaatkan *sliding window* berukuran 50x50 piksel, yang menyesuaikan dengan ukuran data training. Model klasifikasi akan merubah *sliding window* menjadi *bounding box* berwarna kuning seperti oleh bagian a pada Gambar 7 ketika terdapat motif yang teridentifikasi. Selanjutnya label motif diberikan pada citra yang teridentifikasi terdapat motif. Kemudian *bounding box ground truth* diadapatkan terhadap *bounding box* hasil dari model klasifikasi. *Bounding box ground truth* memiliki ukuran yang sama dengan *sliding window* yakni 50x50 piksel.

Dalam pelabelan motif, jika tidak sepadan terhadap *bounding box ground truth*, label sesuai hasil klasifikasi model klasifikasi motif tetap diberikan pada citra namun dianggap sebagai *false positif*. *Bounding box ground truth* memiliki warna *default* hitam, seperti ditampilkan oleh bagian b pada Gambar 7. Kemudian seandainya *bounding box* hasil klasifikasi beririsan dengan *bounding box ground truth* dengan nilai irisan lebih besar sama dengan 50% dari ukuran *bounding box*, maka *bounding box ground truth* akan berubah warna menjadi hijau seperti ditampilkan oleh bagian c pada Gambar 7. Hanya saja jika besar irisan lebih kecil dari 50% atau bahkan tidak beririsan (terjadi *false positif*), maka *bounding box* hasil klasifikasi akan berwarna merah dan *bounding box ground truth* tetap berwarna hitam seperti ditampilkan oleh bagian d pada Gambar 7. Sedangkan ketika model

klasifikasi salah mengidentifikasi *background* sebagai motif (dari hasil klasifikasi sebenarnya tidak terdapat *ground truth*) seperti ditampilkan oleh bagian e pada Gambar 7. Yang terjadi adalah warna merah menjadi warna *bounding box* hasil klasifikasi dan label motif tetap diberikan pada citra. Tabel III menampilkan contoh pelabelan hasil deteksi motif batik.

Pada Tabel III, label pada hasil *testing* nomor 4, 5 dan 6 memiliki hasil klasifikasi yang sama persis dengan label sebenarnya sehingga nilai *tanimoto distance* bernilai 0. Kemudian pada hasil *testing* nomor 1, label hasil klasifikasi tidak sama dengan label sebenarnya sehingga nilai *tanimoto distance* bernilai 1. Sedangkan pada hasil *testing* nomor 2 dan 3, label hasil klasifikasi hanya sama pada satu label, sedangkan label lainnya salah sehingga nilai *tanimoto distance* bernilai 0.5.

A. Pengujian Terhadap Data Testing

Tabel IV, Tabel V, Tabel VI dan Tabel VII menampilkan hasil pengujian pada data *testing* dengan *weak classifier* yang masing-masing berjumlah 100, 200, 300 dan 400.

Dari Tabel IV, Tabel V, Tabel VI dan Tabel VII, pada pengujian terhadap 200 *weak classifier* (pada 5 *subsets ferns*) serta 300 *weak classifier* (pada 5 dan 15 *subsets ferns*) diperoleh rata-rata nilai *tanimoto distance* terbaik. Rata-rata nilai *tanimoto distance* yang diperoleh adalah sama, yakni 0.047 dengan citra benar sejumlah 60 dan citra salah sejumlah 4.

Dalam Tabel IV, Tabel V, Tabel VI dan Tabel VII diketahui bahwa rata-rata nilai *tanimoto distance* yang terburuk bernilai 0.082 sedangkan yang terbaik bernilai 0.047. Semua rata-rata nilai *tanimoto distance* berada di bawah 0.100 sehingga bisa dikatakan klasifikasi multilabel motif batik berhasil melakukan klasifikasi motif dengan baik. Untuk jumlah citra yang dilabeli dengan benar paling sedikit adalah 55 citra, sedangkan yang paling banyak adalah 60 citra dari total 64 citra *testing*. Dari sisi jumlah citra yang dilabeli dengan benar, akurasi terendah yakni 85,9% (55 citra dilabeli dengan benar) dan akurasi tertinggi 93,7% (60 citra dilabeli dengan benar).

Jumlah *subsets ferns* dan jumlah *weak classifiers* berpengaruh terhadap jumlah waktu yang dibutuhkan untuk proses *training* dan *testing*. Semakin banyak yang digunakan, waktu yang dibutuhkan juga semakin banyak. Namun dalam *testing*, jika jumlah *weak classifier* sama tapi digunakan berbagai variasi jumlah *subset ferns* alokasi waktu yang dibutuhkan rata-rata tidak berbeda jauh.

B. Pengujian Bootstrapping

Terdapat 6 variasi iterasi dalam pengujian *bootstrapping*, yakni pengujian pada iterasi ke-0 (tanpa iterasi) serta iterasi ke-1 hingga iterasi ke-5. Parameter dalam pengujian *bootstrapping* antara lain jumlah file benar, jumlah file salah, jumlah rata-rata nilai *tanimoto distance*, waktu untuk melakukan *testing* dan waktu untuk melakukan *training*. Hanya kombinasi 5 *subset ferns* dan 100 *weak classifier* yang dipakai pada pengujian *bootstrapping*. Hasil dari pengujian ditampilkan pada Tabel VIII.

Pada Tabel VIII, pada iterasi *bootstrapping* ke-2 dengan nilai 0.013 dan ke-5 dengan nilai 0.023, kedua nilai tersebut lebih baik dari nilai tertinggi dalam pengujian *testing* (0.047). Jumlah file yang dideteksi secara benar terbanyak terdapat pada iterasi ke-2 yakni 62 file. Semakin banyak iterasi *bootstrapping* yang digunakan maka semakin lama waktu *training*. Alasannya adalah karena penambahan jumlah iterasi *bootstrapping* maka akan menambah jumlah data *training*. Untuk waktu *testing* dalam iterasi *bootstrapping* rata-rata adalah dua ribu detik, alasannya adalah kombinasi jumlah *subset ferns* dan *weak classifier* yang digunakan oleh masing-masing iterasi *bootstrapping* adalah sama.

IV. KESIMPULAN DAN SARAN

Dari penelitian ini, diperoleh kesimpulan:

- a. Kombinasi metode klasifikasi *Boosted Random Ferns* dan *Histogram of Oriented Gradient* terbukti berhasil melakukan klasifikasi multilabel motif batik.
- b. Hasil klasifikasi multilabel motif batik menunjukkan akurasi yang baik dengan jumlah data benar berada pada kisaran 55 sampai 62 data yang diklasifikasi dengan benar dari 64 data citra *testing* yang diujikan.

Untuk pengembangan penelitian lebih lanjut disarankan penggunaan jumlah jenis motif batik untuk mengetahui performa sistem yang dibangun, atau bisa juga digunakan dataset multilabel lain yang memiliki karakteristik variasi bentuk seperti dataset motif batik.

DAFTAR PUSTAKA

- [1] Riesmala, C. P., Rizal, A., & Novamizanti, L. (2012). Pengenalan Motif Batik dengan Analisis Struktur dan Warna Pada Citra Digital. Tugas Akhir, Telkom University, Bandung..
- [2] Rangkuti, A. H. (2013). Klasifikasi Motif Batik Berbasis Kemiripan Ciri Dengan Wavelet Transform Dan Fuzzy Neural Network, ComTech, 5(1), 361–372.

- [3] Kurniawardhani, A., Suciati, N., & Arieshanti, I. (2014). Klasifikasi Citra Batik Menggunakan Metode Ekstraksi Ciri yang Invariant Terhadap Rotasi. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 12(2), 48.
- [4] Sulistyono A.S., P. (2016). Sistem Pengenalan Pola Motif Batik Pada Perangkat Android Dengan Jaringan Syaraf Tiruan, Tugas Akhir, Universitas Gadjah Mada, Yogyakarta.
- [5] Dalal, N., & Triggs, W. (2005). Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR05*, 1(3), 886–893.
- [6] Asiri, N. M., AlHumaidi, N., & AIOsaim, N. (2015). Combination of Histogram of Oriented Gradients and Hierarchical Centroid for Sketch-Based Image Retrieval. *2015 Second International Conference on Computer Science, Computer Engineering, and Social Media (CSCESM)*, 149–152.
- [7] Randa, A. F., Suciati, N., & Navastara, D. A. (2016). Implementasi Metode Kombinasi Histogram Of Oriented Gradients Dan Hierarchical Centroid Untuk Sketch Based Image Retrieval. *Jurnal Teknik ITS*, 5(2), 311-316.
- [8] Villamizar, M., Andrade-Cetto, J., Sanfeliu, A., & Moreno-Noguer, F. (2012). Bootstrapping Boosted Random *Ferns* for discriminative and efficient object classification. *Pattern Recognition*, 45(9), 3141–3153.
- [9] Nilogiri, A., Suciati, N., & Purwitasari, D. (2012). Klasifikasi Kansei Multi Label Dengan Probabilistic Neural Network Pada Citra Batik Menggunakan Klasifikasi Kansei Multi Label Dengan Probabilistic Neural Network Pada Citra Batik Menggunakan Kombinasi Fitur Warna, Tekstur, Dan Bentuk. *Seminar Nasional Manajemen Teknologi, XVI*, 1–9.
- [10] Villamizar, M., Moreno-Noguer, F., Andrade-Cetto, J., & Sanfeliu, A. (2010). Shared random *ferns* for efficient detection of multiple categories. *Proceedings - International Conference on Pattern Recognition*, 388–391.
- [11] Tomasi, C. (2012). Histograms of Oriented Gradients. *Computer Vision Sampler*, 1–6.
- [12] Villamizar, M., Andrade-cetto, J., & Sanfeliu, A. (2017). Boosted Random *Ferns* for Object Detection, 8828(c).
- [13] Oliveira, E., Ciarelli, P. M., Badue, C., Goiabeiras, C. De, Ferrari, A. F., & Postal, C. (2008). a KNN based Approach and a PNN Algorithm for a Multi-Label Classification Problem, Eighth International Conference on Intelligent Systems Design and Applications.