

ALPHABETS IMAGE IDENTIFICATION USING ADVANCED LOCAL BINARY PATTERN AND CHAIN CODE ALGORITHM

Daniel Setiawan Cahyono¹, Shinta Estri Wahyuningrum²

^{1,2}Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Katholik
Soegijapranata

¹daniel28sc@gmail.com, ²shinta@unika.ac.id

Abstract

Optical Character Recognition (OCR) is a method for computer to process an image that contains some text and then try to find any characters in that image, then convert it to digital text. In this research, Advanced Local Binary Pattern and Chain Code algorithm will be tested to identify alphabets in the image. Several method image preprocessing are also needed, such as image transformation, image rescaling, grayscale conversion, edge detection and edge thinning.

Keywords: ocr, albp, chain code, image identification

Pendahuluan

Proses komputer membaca sebuah tulisan pada gambar disebut sebagai optical character recognition (OCR)[6], dalam OCR komputer dapat karakter yang ada pada gambar lalu di olah menjadi tulisan digital / informasi yang nantinya dapat di ubah bila di perlukan. Dalam proses OCR tersebut terdapat berbagai macam metode atau algoritma, antara lain algoritma untuk menormalisasi gambar atau image preprocessing, algoritma pencari fitur atau ciri pada gambar (feature extraction), algoritma pengidentifikasi atau pengklasifikasi objek. Pada jurnal ini membahas pendeteksian bentuk huruf yang ada pada sebuah gambar menggunakan algoritma feature extraction Advanced Local Binary Pattern (ALBP) dan juga algoritma feature extraction Chain-Code. Dan pada penelitian ini menggunakan berbagai metode image preprocessing untuk menormalisasi semua gambar, antara lain menyamakan ukuran gambar (image resizing), mengubah warna gambar menjadi keabu-abuan(grayscale color), setelah itu mencari garis tepi pada objek, lalu menipiskan garis tepi menjadi hanya 1 pixel tipisnya. Hasil dari perhitungan rumus ALBP disimpan dalam bentuk histogram, sedangkan hasil dari Chain-Code akan berupa sekumpulan nilai arah dari garis tepi object.

Pada jurnal ini akan menguji dengan objek gambar huruf besar A – Z, dan akan menguji pada posisi gambar huruf normal / standart, huruf yang telah terkena efek tools rotasi, shear, perpective dari aplikasi GIMPP. Lalu juga akan menguji coba apakah algoritma ALBP dan Chain-Code masih dapat mendeteksi gambar huruf bila gambar

tersebut di beri efek blur dan juga noise. Hasil dari pengujian ini akan berupa besaran akurasi yang didapat dari algoritma ALBP dan Chain-Code.

Pengumpulan Gambar

Gambar huruf yang digunakan pada penilitan ini di buat dengan aplikasi GIMPP dengan ketentuan:

- Ukuran gambar 1000x1000.
- Warna latar putih.
- Jenis huruf adalah Times New Roman.
- Ukuran huruf adalah 500 dan berwarna hitam.
- Huruf A sampai Z.

Untuk gambar latihan yang akan disimpan ke database, terdapat 26 gambar, dan bentuk huruf normal tanpa efek apapun.

Untuk gambar uji coba, terdiri dari gambar huruf yang sudah di ubah / di manipulasi menggunakan tools dari GIMPP antara lain tools rotation, shear , perspective. Cara memanipulasi gambar pun secara bebas besaran perubahannya. Lalu ada pula gambar huruf normal, yang akan di beri efek blur dan noise. Total gambar yang akan di uji adalah 130 gambar.

Image Pre-Processing

Gambar – gambar yang sudah terkumpul harus di sesuaikan terlebih dahulu untuk membantu proses komputasi citra. Tahapan image pre-processing yang digunakan yaitu:

1. Mengubah ukuran gambar menjadi 500x500
2. Menjadikan warna pada gambar menjadi abu – abu
3. Menggunakan Sobel Edge Detector untuk mencari garis tepi pada objek yang ada di gambar
4. Proses thinning atau penipisan ukuran garis menjadi hanya 1 pixel tebal nya dengan menggunakan algoritma zhang-suen.

Proses kerja algoritma zhang-suen memiliki dua-subiterasi, subiterasi pertama, akan menghapus pixel berwarna hitam bila memenuhi kondisi :

1. $2 \leq B(P1) \leq 6$
2. $A(P1) = 1$
3. $P2 \times P4 \times P6 = 0$

$$4. P_4 \times P_6 \times P_8 = 0$$

Dimana $A(P_1)$ adalah banyak pola pixel yang bertransisi (0-1), P_2-P_8 adalah 8 tetangga dari P_1 , $B(P_1)$ adalah banyak tetangga yang memiliki nilai 1. Pada subiterasi kedua, akan menghapus pixel berwarna hitam bila memenuhi kondisi:

1. $2 \leq B(P_1) \leq 6$
2. $A(P_1) = 1$
3. $P_2 \times P_4 \times P_8 = 0$
4. $P_2 \times P_6 \times P_8 = 0$

Iterasi ini akan terus dilakukan sampai tidak ada lagi perubahan[5].

Ekstraksi ciri

Gambar yang telah melewati proses image pre-processing akan di ekstrak cirinya menggunakan algoritma Advanced Local Binary Patterns (ALBP) dan juga algoritma Chain-Code. Berikut adalah penjelasan dari cara kerja algoritma ALBP:

Advanced Local Binary Patterns adalah pengembangan dari algoritma ekstraksi ciri Local Binary Pattern, yang memiliki keunggulan lebih tahan terhadap efek rotasi, atau dapat mengidentifikasi gambar meskipun terkena efek rotasi. Rumus Advanced Local Binary Patterns pada dasarnya sama dengan rumus Local Binary Patterns. Cara

$$LBP(m, R) = \sum_{i=0}^{p-1} u(t_i - t_c) 2^i$$

Rumus 1: Local Binary Patterns

kerja rumus LBP adalah dengan membandingkan setiap nilai pixel tengah dengan membandingkan tiap pixel pada citra dengan nilai pixel pada 8 tetangga yang ada di sekitarnya. Bila nilai pixel tetangganya lebih besar sama dengan dari nilai pixel tengah maka akan mendapat nilai 1, bila lebih kecil maka akan mendapat nilai 0. Nilai – nilai yang diperoleh akan di susun menjadi bilangan 8bit binari [1].

1	5	6
9	4	7
8	2	3

Gambar 1: Contoh pixel 3x3

Nilai 4 adalah nilai tengah / pusat, yang akan dibandingkan dengan setiap nilai tetangga sekitarnya searah jarum jam dimulai dari pojok kiri atas. Dari contoh Gambar 1

diatas, maka akan menghasilkan susunan 8 angka biner 01110011 (*rumus 1*) dengan nilai desimal 115.

$$ALBP(m, R) = \min(Cir(LBP(m, R), n))$$

Rumus 2: ALBP

Setelah mendapatkan nilai LBP, nilai tersebut akan di proses dengan rumus ALBP, dengan cara menggeser searah jarum jam susunan nilai biari hasil LBP diatas satu persatu lalu menyimpan nilai minimal yang dihasilkan. Pada contoh diatas, susunan angka tersebut akan digeser terus sampai berputar menempati semua titik. Hasil penggeseran pertama adalah 10111001 dengan nilai desimal 185, lalu di geser lagi menjadi 11011100 dengan nilai desimal 220, terus lakukan pergeseran sampai 7 kali, lalu simpan nilai terkecilnya. Untuk contoh di atas akan mendapatkan nilai minimal dengan susunan binari 00110111 dengan nilai desimal 55. nilai 55 tersebut akan di simpan kedalam histogram. Histogram adalah nilai frekuensi kemunculan nilai hasil perhitungan ALBP. Proses perhitungan akan terus dilanjutkan hingga akhir pixel [2].

Cara kerja algoritma Chain-Code adalah pertama dengan mencari posisi titik mulai awal, posisi titik mulai awal adalah posisi pixel hitam pertama yang di temukan. Dari posisi tersebut, progam akan mencari apakah ada pixel hitam yang terhubung pada tetangganya, bila ada maka posisi pointer akan bergerak ke posisi titik pixel hitam tersebut dan mencatat arah nya, lalu titik tersebut akan di beri keterangan bahwa sudah di lewati, titik yang telah di lalui tidak boleh di lalui lagi. Cara ini akan di lakukan terus menerus sampai tidak ada lagi titik hitam yang terhubung / yang dapat di tempati. Hasil akhir dari proses Chain-Code adalah daftar susunan nilai arah pergerakan pointer [3].

Proses Identifikasi

Pada proses ini terdapat 2 cara untuk identifikasi, karena proses ekstraksi ciri menggunakan algoritma ALBP dan Chain-Code menghasilkan output yang berbeda. Untuk mengidentifikasi hasil dari algoritma ALBP akan menggunakan rumus Chi-Square. Chi-Square akan menghitung seberapa jauh perbedaan nilai histogram nai ALBP pada gambar di database dengan gambar yang di uji, berikut adalah rumus Chi-Square.

$$X^2(S, M) = \sum_{b=1}^B \frac{(S_b - M_b)^2}{S_b + M_b}$$

Rumus 3: Chi-Square

S = Sample data (Histogram gambar database)

M = Model data (Histogram gambar uji)

b = Indek histogram

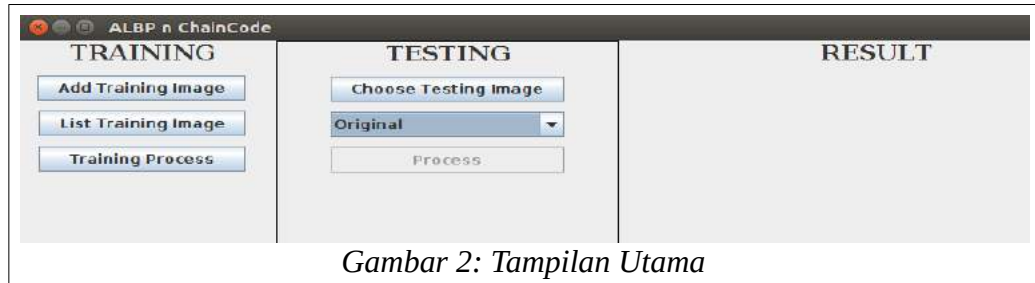
hasil Nilai terkecil menandakan bahwa gambar tersebut paling mirip / sama [1].

Untuk mengidentifikasi hasil dari algoritma Chain-Code akan menggunakan proses perbandingan. Hasil kumpulan nilai arah dari gambar uji dan database akan

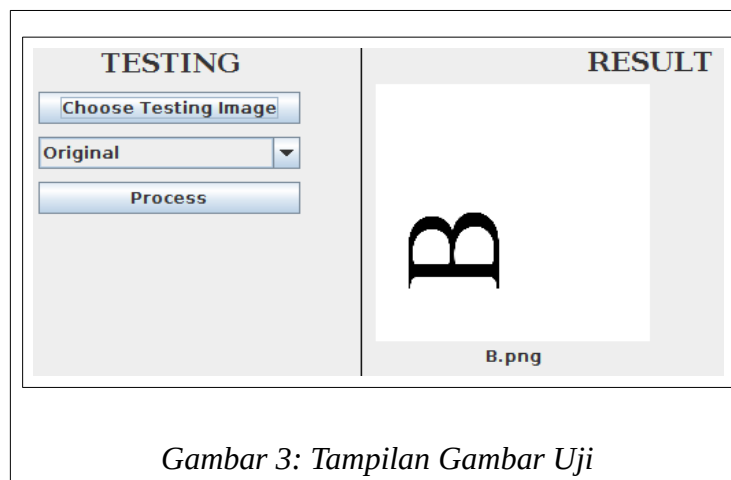
dibandingkan dengan cara di cocokkan satu persatu apakah urutan dan nilai nya sama. Semakin besar nilai nya, artinya gambar tersebut lebih identik.

Tampilan Program

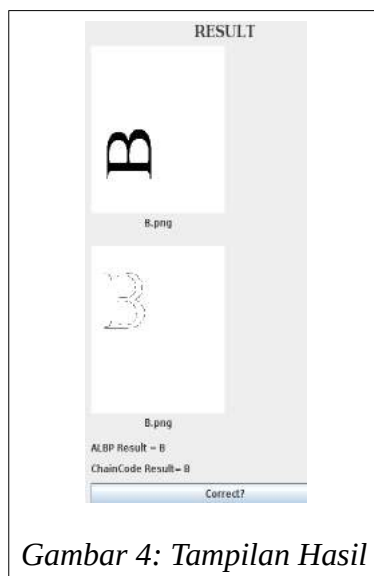
Berikut ini adalah tampilan dari aplikasi program yang telah di buat.



Terdapat 3 bagian utama, yaitu training, testing dan result. Kolom training terdiri dari tombol 'Add Training Image' untuk menambahkan gambar ke database, tombol 'List Training Image' untuk menampilkan daftar gambar – gambar yang ada di database, 'Training Process' untuk memulai proses ekstraksi ciri pada setiap gambar di database. Kolom testing terdiri dari tombol 'Choose Testing Image' untuk memilih gambar yang akan di uji, lalu dibawahnya ada ComboBox yang berisi pilihan efek 'Original, Blur, Noise'. Tombol 'Process' akan dapat di klik setelah pengguna memilih gambar dan memilih efek gambar.

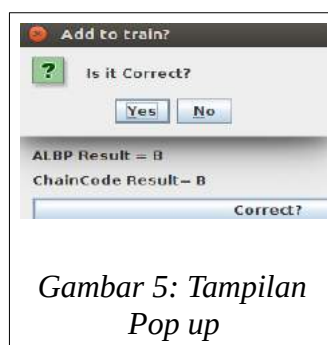


Di atas adalah tampilan setelah pengguna memilih gambar uji. Gambar uji akan muncul pada kolom result, dan nama file akan di munculkan.



Gambar 4: Tampilan Hasil

Setelah menklik tombol 'Process' maka hasil gambar yang identik dengan gambar uji akan di tampilkan di bawahnya, dan terdapat keterangan kategori hasil proses algoritma ALBP dan juga algoritma Chain-Code. Dan di paling bawah terdapat tombol 'Correct?' yang memunculkan pop up pertanyaan 'Is it Correct?'



Gambar 5: Tampilan Pop up

Lalu pengguna akan di beri pertanyaan apakah gambar yang di uji ingin di masukan ke dalam database, agar kedepan hasil identifikasi akan menjadi lebih akurat



Gambar 6: Tampilan 'Add to Train'

Pengujian

Progam pada penelitian ini dibuat dengan menggunakan bahasa Java, dan akan dilakukan pengujian dengan beberapa kumpulan gambar, yaitu gambar huruf normal, gambar huruf yang telah di rotasi, gambar huruf yang telah di miringkan, gambar huruf

yang telah di beri efek perspective, gambar huruf yang di buramkan, gambar huruf yang di beri noise. Pengujian di lakukan dengan cara setiap 1 gambar uji akan dibandingkan dengan setiap gambar yang ada pada database, dan nilai yang paling cocok berarti gambar uji tersebut memiliki kategori yang sama dengan gambar yang ada di database.






Dibawah ini adalah semua gambar huruf asli yang akan di gunakan sebagai gambar latih.

Tabel 1: Gambar Latih

A	B	C	D	E	F	G
H	I	J	K	L	M	N
O	P	Q	R	S	T	U
V	W	X	Y	Z		

Dibawah ini adalah table contoh gambar yang di gunakan sebagai gambar uji.

Tabel 2: Gambar Uji

Gambar Asli	Gambar Terotasi	Gambar Miring	Gambar Perspektif	Gambar Buram	Gambar Noised
A					

Gambar asli adalah gambar huruf yang masih standar tidak diberi efek apapun, gambar asli di gunakan sebagai gambar latih yang ada di database, dan juga gambar uji. Gambar terotasi adalah gambar asli yang di rotasi dari rotasi 0' sampai 360' secara bebas dan acak. Gambar miring adalah gambar asli yang di beri efek menggunakan Shear tools GIMPP, besar nya kemiringan adalah bebas dan acak. Gambar perspektif adalah gambar asli yang diberi efek menggunakan perspective tools GIMPP, yang akan merubah bentuk seakan dilihat dari sudut yang berbeda. Penggunaan perspective tools ini bebas tanpa ketentuan. Gambar buram adalah gambar asli yang sudah melalui proses image pre-processing lalu diberi efek buram. Gambar noised adalah gambar asli yang sudah melalui proses image pre-processing lalu di beri efek noise and salt. Berikut ini adalah tabel akurasi algoritma ALBP dan algoritma Chain-Code dari pengujian yang telah dilakukan

Tabel 3: Hasil Pengujian

Kondisi	Akurasi ALBP	Akurasi Chain-Code
Asli	100%	100%
Gambar Terotasi	65,38%	76,92%
Gambar Miring	50%	96,15%
Gambar Perspektif	61,53%	73,07%
Gambar Buram	19,23%	0%
Noised Image	0%	0%
Rata – rata Akurasi	49,35%	57,68%

Kesimpulan

1.1 Kesimpulan

Algoritma Advanced Local Binary Pattern dan Chain-Code dapat digunakan untuk mendeteksi bentuk huruf pada sebuah gambar. Pada kondisi asli hasil akurasi dari Advanced Local Binary Pattern dan Chain-Code sangat tinggi sampai 100%. Saat gambar yang di uji terkena berbagai macam efek, contohnya rotasi, kemiringan, perspektif, hasil akurasi yang paling tinggi dari kedua algoritma tersebut adalah Chain-Code. Kesamaan dari kedua algoritma tersebut adalah keduanya tidak dapat mendeteksi huruf bila gambar yang diuji diberi efek buram dan noise.

1.2 Saran

Saran untuk penelitian selanjutnya adalah untuk mencoba mengidentifikasi kata yang terdapat pada gambar dengan menggunakan algoritma Chain Code. Dan untuk mendapatkan akurasi yang lebih baik bisa menambahkan algoritma machine learning seperti SVM atau Backpropagation.

Daftar Pustaka

- [1] Md. Rahim A., Md. Hosain N. M., Wahid T., Md. Azam S., "Face Recognition Using Local Binary Patterns (LBP)", Global Journal of Computer Science And Technology Grapich & Vision, 2013.
- [2] Liao S. And Chung A. C. S., "Texture Classification By Using Advanced Local Binary Patterns And Spatial Distribution Of Dominant Patterns", ICASSP, 2007.
- [3] Sleit A. And Jobay R., "A Chain Code Approach For Recognizing Basic Shapes", CSIT, 2006.
- [4] Widiarti A. T., "Comparing Hilditch, Rosenfeld, Zhang-Suen, and Nagendrasad - Wang-Gupta Thinning Algorithms for Javanese Character Image", World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering, 2011.
- [5] Fakhrina F. A., Rahmadwati, Wijono, "Thinning Zhang-Suen dan Stentiford untuk Menentukan Ekstraksi Ciri(Minutiae) Sebagai Identifikasi Pola Sidik Jari Whorl dan Loop", Teknologi Elektro Vol 15 No 2, 2016.
- [6] Eikvil L., "OCR Optical Character Recognition", Norsk Regnesentral, P.B. 114 Blindern,1993.