Model Deformable untuk Mengenali Angka Tulisan Tangan

Maya Rini Handayani
Fakultas Teknologi Informasi dan Komunikasi, Universitas Semarang
email : m4y4_h4nd4@yahoo.com

Abstrak : Paper ini membahas tentang penggunaan deformable model untuk pengenalan angka hasil tulisan tangan yang memiliki gaya tulisan yang bervariasi. Setiap bentuk karakter angka dimodelkan sebagai *B-spline* dan pixel-pixel data dimodelkan dengan fungsi Gaussian. Parameter-parameter model yang berhubungan dengan spline dan fungsi Gaussian, diestimasi menggunakan inferensi Bayesian. Pada *Bayesian framework*, proses klasifikasi merupakan proses pemilihan model. Pendekatan ini telah diuji dengan angka kesalahan sekitar 4%.

Kata kunci : deformable, pixel, titik, angka, tulisan

PENDAHULUAN

Template matching (pencocokan template) telah lama digunakan untuk pengenalan karakter hasil cetakan. Untuk mengenali karakter tulisan tangan yang memiliki bentuk yang sangat bervariasi, diperlukan jumlah template yang sangat banyak untuk merepresentasikan semua perubahan bentuk (*deformation*) yang mungkin terjadi, atau digunakan model yang dapat berubah bentuk (*deform*).

Alternatif kedua tersebut memerlukan sumber daya lebih sedikit, dan lebih mudah dikelola. Dengan *deformable model* (model yang dapat berubah bentuk), telah berhasil didemonstrasikan pengenalan dengan hasil akurasi yang tinggi. Database pada paper ini telah diuji dengan menggunakan database NIST yang berisi sekitar 10000 digit.

Dengan pendekatan ini dapat diperoleh hasil dengan angka kesalahan 4.05% dimana angka penolakan 9.88%. Database NIST merupakan database yang dikembangkan oleh *National Institute of Standards and Technology*, yaitu sebuah lembaga penelitian yang membuat dan mengembangkan perangkat lunak untuk perhitungan-perhitungan yang akurat.

Pendekatan *deformable model* ini menarik karena memungkinkan untuk mengintegrasikan proses segmentasi dan pengenalan. Namun aplikasinya masih terbatas karena diperlukannya kompleksitas komputasional yang tinggi.

PEMODELAN

Deformable model biasanya dirumuskan dengan kriteria deformasi model (*model deformation criterion*) dan kriteria ketidakcocokan data (*data mismatch criterion*). Dengan distribusi Gibbs, dapat diperoleh interpretasi probabilistik.

Pada *Bayesian framework*, kriteria deformasi model dapat dipandang sebagai distribusi prior untuk parameter-parameter model, dan kriteria data mismatch dipandang sebagai fungsi kemungkinan (*likelihood function*). Dengan demikian pencocokan yang optimal akan berkaitan dengan estimasi parameter-parameter, yang dilakukan dengan cara mendeformasi model angka secara iteratif untuk dicocokkan dengan karakter inputnya, sedangkan pengklasifikasian berkaitan dengan pilihan model yang digunakan.

1. Kriteria Deformasi Model (Prior Distribution)

Karakter tulisan tangan dimodelkan sebagai *cubic B-spline*, yaitu suatu teorema spline yang ditemukan oleh Isaac Jacob Schoenberg, yang masing-masing diparameterkan sebagai suatu himpunan dari *k* titik kontrol.

Derajat deformasi, yang dinyatakan dengan kriteria deformasi model $E_w(\mathbf{w})$ dari model ke-i H_i, didefinisikan sebagai jarak *Mahalanobis* dari vektor titik kontrol, $\mathbf{w} \in$

$\Re^{2k}$, dari vektor lokasi awal yang telah ditentukan, $\mathbf{h}_i \in \Re^{2k}$. Jarak Mahalanobis adalah metode pengukuran jarak pada pencocokan kemiripan data yang menghitung jarak antara 2 titik dalam ruang dimensi p. Penghitungan ini sama dengan jarak *Euclidean* bila varian matriksnya adalah identik.

Dalam hal ini vektor-vektor dibentuk dengan menggabungkan koordinat x dan y dari seluruh titik kontrol k, yaitu, $\mathbf{w} = [x_1, y_1, x_2, y_2, \ldots, x_k, y_k]^t$. Secara khusus, kriteria deformasi model dinyatakan sebagai :

$$E_w(\mathbf{w}) = \frac{1}{2}(\mathbf{w}-\mathbf{h}_i)^t \sum_i^{-1}(\mathbf{w}-\mathbf{h}_i) \quad \ldots\ldots(1)$$

di mana $\sum_i$ adalah matriks kovarian dari $\mathbf{w}$ untuk H_i dan $\mathbf{w}^t$ menyatakan transpose dari $\mathbf{w}$. Selanjutnya, distribusi prior dari $\mathbf{w}$ diperoleh dari

$$p(\mathbf{w} \mid \alpha, H_i) = \frac{1}{Z_w(\alpha)} \exp(-\alpha E_w(\mathbf{w})) .(2)$$

di mana $Z_w(\alpha) = \left(\frac{2\pi}{\alpha}\right)^k |\sum_i|^{\frac{1}{2}}$. Komponen dari $\mathbf{h}_i$ dan $\sum_i$ dihitung dengan estimasi kemungkinan maksimum pada saat tahap pelatihan.

Seringkali orang menulis angka dalam bentuk yang berbeda meskipun termasuk kelas angka yang sama. Variasi yang terjadi kadang-kadang tidak dapat direpresentasikan dengan deformasi satu model angka, misalnya "7" dan "7" untuk kelas angka tujuh. Hal ini menimbulkan pemikiran bahwa dengan menggunakan beberapa model untuk masing-masing kelas mungkin akan memberikan hasil yang lebih baik.

Untuk membuat pengkelasan angka secara otomatis dengan menggunakan data latihan merupakan hal yang tidak mudah. Untuk itu model awal dibuat secara manual berdasarkan penelitian pada variasi yang umumnya terjadi pada tulisan tangan yang sebenarnya.

Model awal ini selanjutnya dapat diperbaiki secara otomatis dengan algoritma seperti *K-means clustering*, dimana pelatihan dilakukan pada model tertentu yang paling mendekati kecocokan. Algoritma K-means clustering adalah algoritma untuk mengklasifikasikan obyek menjadi group-group yang berbeda berdasarkan atribut menjadi partisi K.

2. Kriteria Data Mismatch (Likelihood Function)

Penyebaran dari pixel berwarna hitam dimodelkan menggunakan gabungan berbobot sama dari Gaussian dengan rata-ratanya yang ditempatkan di sepanjang spline. Ketidakcocokan *(mismatch)* antara model dengan data diukur dengan kriteria data mismatch yang didefinisikan dengan:

$$E_D(\mathbf{w}) = -\log\left[\prod_l^N \frac{1}{N_g}\sum_j^{N_g} \exp\left(-\beta \frac{\|\mathbf{m}_j - \mathbf{y}_l\|^2}{2}\right)\right] .(3)$$

Kemudian fungsi kemungkinan *(likelihood function)* diperoleh dengan

$$p(\mathbf{D} \mid \mathbf{w}, \beta, H_i) = \frac{1}{Z_D(\beta)} \exp(-E_D(\mathbf{w})) .(4)$$

dimana $Z_D(\beta) = (2\pi/\beta)^N$, $\mathbf{m}_j = \mathbf{S}_j^t \mathbf{w}$, N adalah banyaknya pixel berwarna hitam, N_g adalah banyaknya Gaussian pada spline, $\mathbf{S}_j$ adalah matriks 2k x 2 yang berisi koefisien cubic B-spline, β adalah inverse varian dari Gaussian untuk pemodelan tebal garis karakter, $\mathbf{m}_j \in \Re^{2k}$ adalah rata-rata dari Gaussian ke-j, dan $\mathbf{y}_l \in \Re^{2k}$ adalah lokasi dari suatu pixel berwarna hitam secara individu.

Disebabkan gaya tulisan tangan setiap orang bervariasi, frame model yang digunakan akan sering berbeda dengan frame data karakter input yang akan dikenali. Untuk mengatasi variasi seperti ini, digunakan *affine transform* $\{\mathbf{A},\mathbf{T}\}$ untuk memetakan vektor titik kontrol $\mathbf{w}$ dalam frame model ke rata-rata Gaussian $\mathbf{m}_j$ dalam frame data sehingga $\mathbf{m}_j^t = \mathbf{S}_j^t(A\mathbf{w} + \tau)$, dimana A adalah matriks diagonal 2k x 2k dengan k submatriks $\mathbf{A}$ ditempatkan pada diagonalnya, dan τ adalah vektor 2k x 1 yang dibentuk dengan menggabungkan k

vektor **T**. Pada dasarnya, **A** digunakan untuk menentukan perubahan ukuran, kemiringan, dan rotasi dari model karakter sedangkan **T** untuk pergeseran.

3. Kriteria Fungsi Gabungan (Posterior Distribution)

Selanjutnya dilakukan penggabungan kriteria deformasi model dan kriteria data mismatch, yaitu dengan

$$E_M(\mathbf{w}) = \alpha E_w(\mathbf{w}) + E_D(\mathbf{w}) \text{ .....(5)}$$

dan distribusi posterior dari **w** didefinisikan sebagai

$$p(\mathbf{w}|\mathbf{D},\alpha,\beta,H_i) = \frac{1}{Z_M(\alpha,\beta)} \exp(-E_M(\mathbf{w})) \text{ .(6)}$$

dimana $Z_M(\alpha,\beta) = \int \exp(-E_M(\mathbf{w}))d\mathbf{w}$.

Distribusi posterior ini kemudian digunakan dalam proses pencocokan dan klasifikasi. Parameter pengaturan α adalah untuk menyeimbangkan pengaruh antara model dengan data.

PENCOCOKAN DAN PENGKLASIFIKASIAN

Tahap selanjutnya yaitu menentukan langkah-langkah untuk pencocokan dan klasifikasi. Langkah tersebut yaitu :

1. Estimasi Titik Kontrol

Estimasi *maximum a posteriori* (MAP) dari vektor titik kontrol w* diperoleh dengan memaksimalkan nilai $p(\mathbf{w} \mid \mathbf{D},\alpha,\beta,H_i)$ di persamaan (6). Di sini digunakan algoritma *expectation-maximization* (EM) yang telah dikenal efisien untuk menghitung hal semacam ini. Algoritma EM merupakan proses iteratif yang terdiri dari dua langkah yaitu :

a. Langkah *Expectation* untuk mengestimasi data yang tidak ada berdasarkan data yang ada.

b. Langkah berikutnya yaitu *Maximization* untuk menentukan estimasi kemungkinan terbesar.

Konvergensi algoritma ini pada titik maksimum lokal telah terbukti dengan baik dan digunakan juga untuk estimasi MAP.

2. Estimasi Parameter Pengaturan dan Ketebalan Garis Karakter

Dengan memaksimalisasi fungsi densitas probabilitas posterior $p(\alpha,\beta \mid \mathbf{D},H_i)$, estimasi MAP dari α dan β dapat ditentukan. Untuk itu harus dipenuhi :

$$\alpha^* = \gamma / (2E_w(\mathbf{w}^*))$$

dan

$$\beta^* = (2N - \gamma) / (2E'_D(\mathbf{w}^*))$$

dimana $\gamma = 2k - \alpha Trace(\nabla\nabla E_M(\mathbf{w}^*)^{-1})$.

Disebabkan tidak ada solusi bentuk tertutup untuk α^* dan β^*, estimasi titik kontrol dan langkah-langkah estimasi $\{\alpha^*,\beta^*\}$ diimplementasikan secara iteratif dimana pada kedua persamaan di atas α dan β digunakan sebagai kriteria konvergensi. Di samping itu diperlukan juga nilai awal untuk α dan β. Gambar 1 menunjukkan contoh beberapa langkah proses pencocokan.

Karakter kecil pada bagian kiri atas masing-masing gambar adalah model yang digunakan. Gambar (a) menunjukkan posisi awal dari model, gambar (b) adalah estimasi dari parameter-parameter awal *affine transform* melalui algoritma EM, dan gambar (c) adalah posisi finalnya.

Gambar 1: Langkah pencocokan dengan deformable model.

3. Pengklasifikasian

Untuk pengklasifikasian diperlukan perhitungan $p(\mathbf{D} \mid H_i)$ berdasarkan pada $\{\mathbf{w}^*, \alpha^*, \beta^*\}$ untuk setiap model H_i, yang didapat dari :

$$p(\mathbf{D} \mid H_i) \alpha \frac{Z_M(\alpha^*, \beta^*)}{Z_w(\alpha^*) Z_D(\beta^*)} \sqrt{\frac{2}{\gamma}} \sqrt{\frac{2}{2N-\gamma}} \quad (7)$$

Akhirnya, pengklasifikasian dapat ditentukan pada saat $i^* = \max_i p(\mathbf{D} \mid H_i)$. Agar dapat dilakukan penolakan terhadap input-input yang bersifat ambigu, digunakan aturan penolakan berdasarkan probabilitas posterior kelas, yaitu $p(H_i \mid \mathbf{D})$, dimana

$$p(H_i \mid \mathbf{D}) = \frac{p(\mathbf{D} \mid H_i) P(H_i)}{\sum_j p(\mathbf{D} \mid H_j) P(H_j)} \quad \text{.....(8)}$$

Input yang memiliki probabilitas posterior kelas kurang dari threshold kemantapan D akan ditolak.

HASIL DAN PEMBAHASAN

Percobaan telah dilakukan untuk mengenali angka di dalam NIST Special Database 1. Data yang digunakan berisi 23451 digit (hasil tulisan 200 orang yang berbeda), di mana 11600 digit (hasil tulisan 100 orang) digunakan untuk pelatihan dan sisanya digunakan untuk pengujian. Masing-masing digit merupakan pola biner berukuran 32x32. Hasil percobaannya tampak pada tabel 1.

Nilai threshold kemantapan untuk masing-masing kelas digit dipilih dengan cermat sedemikian hingga angka penolakan hampir sama untuk untuk masing-masing kelas digit. Perlu diperhatikan bahwa jika angka penolakan dinaikkan, reliabilitas akan naik dan angka kesalahan akan turun. Pada angka penolakan 9.88%, dapat dicapai angka kesalahan 4.05%.

Tabel 1. Hasil Percobaan pada Angka Penolakan yang Berbeda-beda

Reliabilitas	92.90%	93.01%	94.18%	95.50%	96.20%	96.44%
Angka pengenalan	92.90%	92.83%	89.82%	86.10%	80.30%	78.10%
Angka penolakan	0%	0.20%	4.63%	9.88%	16.58%	19.02%
Angka kesalahan	7.08%	6.98%	5.55%	4.05%	3.14%	2.88%

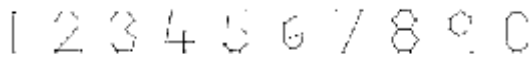
Akurasi dari pengenalan dapat ditingkatkan lebih lanjut dengan menambahkan lebih banyak model untuk masing-masing kelas digit dan membuat representasi model yang lebih bagus, misalnya dengan memodelkan juga pixel berwarna putih.

Meskipun unjuk kerja implementasi ini belum mencapai tingkatan puncak (yaitu dengan sekitar 1% angka kesalahan dan 5% angka penolakan) dan nilai komputasional yang tinggi masih menjadi masalah untuk komputer berkecepatan rendah, pendekatan ini tetap memiliki beberapa keunggulan yang menarik.

Pertama, sifatnya yang *model-driven* (diatur dengan model) sehingga yang dihitung bukan hanya keanggotaan kelas digit dari karakter inputnya, namun juga nilai parameter-parameter modelnya. Hal ini memungkinkan dibuatnya database tulisan tangan online.

Yang kedua, dan ini yang lebih penting, kebanyakan algoritma OCR dapat bekerja pada kasus karakter-tersambung hanya jika informasi segmentasinya tersedia, tetapi segmentasi itu sendiri merupakan masalah yang sulit tanpa adanya informasi pengklasifikasian. Jadi pendekatan model-driven yang diusulkan ini sangat potensial untuk digunakan mengintegrasikan proses segmentasi dan pengenalan tulisan tangan yang bersambung.

Sebagai perbandingan, Hastie-Tibshirani (1994) juga menggunakan *deformable model* untuk pengenalan angka tulisan tangan, namun terdapat perbedaan pada pendekatan model karakter yang digunakan. Hastie-Tibshirani tidak menggunakan *spline*, melainkan menggunakan kurva yang terdiri dari potongan-potongan garis



Gambar 2: Model karakter digit 0-9 dari Hastie

lurus untuk membuat model karakter. Contoh model karakter ini dapat dilihat pada gambar 2.

Model ini memiliki k elebihan dalam hal efisiensi dan kecepatan karena kompleksitas komputasionalnya lebih kecil. Namun dalam hal akurasi tampaknya masih kalah jika dibandingkan dengan metode yang lain.

KESIMPULAN

Pengenalan angka tulisan tangan dengan menggunakan *deformable model*, dapat diperoleh akurasi yang cukup tinggi. Hasil percobaan membuktikan bahwa dengan metode ini dapat diperoleh angka kesalahan 4.05% dengan angka penolakan 9.88%. Parameter-parameter model tidak perlu diinputkan oleh pemakai karena akan diestimasi secara otomatis berdasarkan data input. Metode ini dapat dikembangkan lebih lanjut untuk melakukan pengenalan huruf alfabet tulisan tangan.

DAFTAR PUSTAKA

1. Cheung, K.W., Yeung, D.Y., Chin, R.T., (1995), "A Unified Framework for Handwritten Character Recognition Using Deformable Models", *Proceedings of the Second Asian Conference on Computer Vision*, vol. I, Singapore, ftp://ftp.cs.ust.hk/pub/dyyeung/paper/yeung.accv95.ps.gz.
2. Cheung, K.W., Yeung, D.Y., Chin, R.T., (1996), "Recognition of Handwritten Digits Using Deformable Models", *Proceedings of the Fifth International Workshop on Frontiers in Handwriting Recognition*, England, ftp://ftp.cs.ust.hk/pub/dyyeung/paper/yeung.iwfhr96.ps.gz.
3. Hastie, T.J dan Tibshirani, R., (1994), "Handwritten Digit Recognition via Deformable Prototypes", *AT&T Bell Laboratories Technical Report*, USA, http://www-stat.stanford.edu/~hastie/Papers/zip.proto.ps.
4. Firebaugh, Morris W., (1989), "Artificial Intelligence: A Knowledge-Based Approach", PWS-KENT Publishing Company, USA.