

JARINGAN SYARAF TIRUAN LEARNING VECTOR QUANTIZATION UNTUK APLIKASI PENGENALAN TANDA TANGAN

Difla Yustisia Qur'ani¹, Safrina Rosmalinda²

¹Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia

Jl. Kaliurang Km. 14 Yogyakarta 55501

E-mail: difla.yustisia@gmail.com, rosma2203@yahoo.com

ABSTRAKS

Jaringan Syaraf Tiruan *Learning Vector Quantization (LVQ)* memiliki kemampuan untuk melatih pola-pola yang merupakan *neural network based on competition*. Untuk itu, dalam penelitian ini, akan dibuktikan kemampuan *JST LVQ* tersebut dengan mengembangkan aplikasi untuk mengidentifikasi pola tanda tangan. Pada penelitian ini, digunakan pula metode deteksi tepi (*edge detection*) untuk mengolah citra yang digunakan sebagai masukan jaringan syaraf tiruan ini agar diperoleh garis tepi dari citra tanda tangan. Ini akan membantu menandai bagian yang menjadi detil citra. Dari hasil uji coba program menunjukkan bahwa *JST LVQ* dapat mengenali pola tanda tangan dengan ketepatan 98% pada data uji. Kemungkinan ketidaktepatan ini disebabkan posisi image uji berbeda agak jauh dengan citra-citra yang dilatih.

Kata Kunci: Jaringan Syaraf Tiruan, Learning Vector Quantization, pengenalan tanda tangan

1. PENDAHULUAN

Tanda tangan adalah sebuah bentuk khusus dari tulisan tangan yang mengandung karakter khusus dan bentuk-bentuk tambahan yang sering digunakan sebagai bukti verifikasi identitas seseorang. Sebagian tanda tangan dapat dibaca, namun banyak pula tanda tangan yang tidak dapat dibaca (*unreadable*). Kendati demikian, sebuah tanda tangan dapat ditangani sebagai sebuah citra sehingga dapat dikenali dengan menggunakan aplikasi pengenalan pola pada pengolahan citra (OZ, 2003).

Karena tanda tangan merupakan mekanisme primer untuk *authentication* dan *authorization* dalam transaksi legal, kebutuhan akan penelitian pada pengembangan aplikasi pengenalan dan verifikasi tanda tangan yang otomatis dan efisien meningkat pada tahun-tahun terakhir ini (Özgündüz, 2005).

Pengenalandaan verifikasi tanda tangan meliputi dua bagian yang berbeda tetapi berkaitan erat satu sama lain. Yang pertama adalah identifikasi dari pemilik tanda tangan, sedangkan yang kedua adalah keputusan apakah tanda tangan tersebut asli atau dipalsukan. Selain itu, bergantung pada kebutuhannya, pengenalan tanda tangan dibagi menjadi dua kelas yang berbeda yaitu pengenalan dan verifikasi tanda tangan *online* dan *offline*. Pada pengenalan *online*, dibutuhkan beberapa alat bantu khusus yang digunakan untuk mengukur kecepatan dan tekanan tangan ketika membuat tanda tangan. Di lain pihak, hampir semua system pengenalan tanda tangan *offline* bergantung pada teknik pengolahan citra dan *feature extraction* (OZ, 2003).

Sejumlah pendekatan telah digunakan untuk sistem pengenalan tanda tangan. Justino, Bortolozzi and Sabourin mengimplementasikan system pengenalan tanda tangan *offline* dengan menggunakan Hidden Markov Model. Vélez, Sánchez and Moreno mengajukan system verifikasi

tanda tangan *offline* dengan menggunakan *compression networks* dan *positional cuttings*. Arif dan Vincent (2003) memberi perhatian lebih pada *data fusion* dan teori pembuktian Dempster-Shafer, menggunakan metode *Possibility theory* dan *Borda count*. Chalechale and Mertins menggunakan distribusi segmen garis pada sketsa untuk pengenalan tanda tangan Persia. Sansone and Vento (2000) meningkatkan performa system pengenalan tanda tangan dengan sebuah *serial three stage multi-expert system*.

Pada penelitian ini, digunakan pendekatan Jaringan Syaraf Tiruan *Learning Vector Quantization* untuk pengenalan tanda tangan yang dikombinasikan dengan *Edge Detection Method* pada pengolahan citra tandatangan untuk menandai bagian yang menjadi detil citra tanda tangan.

2. PENGOLAHAN DATA

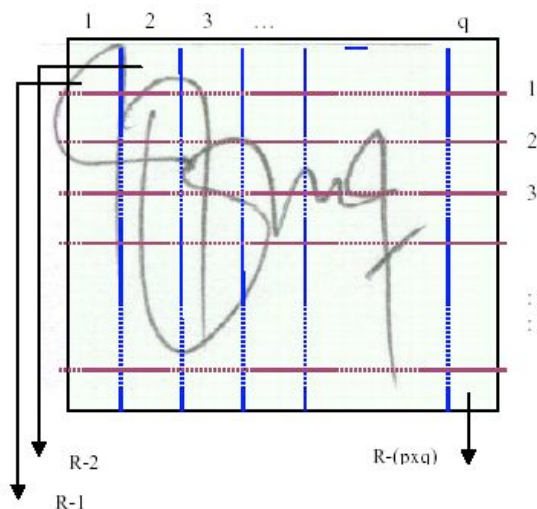
2.1 Preprocessing

Data yang digunakan dalam penelitian ini adalah sekumpulan citra untuk pembelajaran (*learning data set*) dan sekumpulan citra untuk pengujian (*testing data set*) yang diperoleh dari hasil scanning tanda tangan.

Citra tanda tangan yang digunakan berukuran 340x272 pixel. Citra tersebut mengalami proses konversi ke *grayscale* jika merupakan citra *true color*, selanjutnya citra akan dikenai proses deteksi tepi berbasis gradien dengan operator Roberts, yang dilanjutkan dengan proses operasi negasi.

Citra analisis merupakan citra grayscale dengan skala 8 bit yang memiliki intensitas warna berkisar antara 0 sebagai nilai minimum sampai 255 yang merupakan nilai maksimum. Citra analisis kemudian dikonversi ke dalam barisan bilangan biner sepanjang 340x272 untuk masing-masing citra. Untuk setiap pixel dengan nilai intensitas warna ≤ 237 diberi nilai 1, sedangkan untuk nilai intensitas

warna > 237 diberi nilai 0, sehingga akan diperoleh barisan nilai yang terdiri dari 0 dan 1 sepanjang 92480 elemen. Kemudian dilakukan proses pembagian *region* (wilayah) sehingga menghasilkan 10x8 *region* yang berisi penjumlahan nilai pixel dalam *region* yang masing-masingnya berukuran 34x34 pixel.



Gambar 1. Pembagian *region* dari $m \times n$ (340x272) pixel menjadi $p \times q$ (10x8)

2.2 Edge Detection Method

Edge detection atau deteksi tepi digunakan untuk melihat apakah suatu *edge* atau tepi melewati atau berada di dekat suatu titik (*pixel*) dalam sebuah citra (Sigit, dkk., 2005). Tepi adalah batas antara dua daerah dengan sifat tingkat keabuan yang relatif berbeda.

Tujuan dari deteksi tepi adalah menandai bagian yang menjadi detil citra, dan memperbaiki detil citra yang kabur karena *error* atau efek proses akuisisi citra (Sigit, dkk., 2005).

Batas suatu obyek semakin nampak bersamaan dengan ketidak kontinuitasan intensitas dalam suatu citra. Eksperimen pada sistem penglihatan manusia menunjukkan bahwa suatu citra sangat penting, bahkan suatu obyek dapat dikenali hanya dari garis bentuk kasarnya saja (*croude outline*). Kenyataan ini merupakan konsep prinsip untuk merepresentasikan suatu obyek melalui batasnya.

Tepi lokal (*local edge*) adalah daerah kecil dalam suatu citra yang tingkat keabuan lokalnya berubah dengan sangat cepat dengan cara yang sederhana (misalnya monotonik). Operator tepi (*edge operator*)

adalah operator matematis (atau yang hasil komputasinya ekuivalen), yang dengan perluasan kecil, didesain untuk mendeteksi keberadaan tepi lokal dalam suatu fungsi citra (Low, 1991). Pada intinya, ide yang mendasari sebagian algoritma pendeteksian tepi adalah perhitungan suatu operator derivatif lokal.

Dalam teknik deteksi tepi ini, diasumsikan bahwa tepi adalah pixel yang memiliki nilai gradien tinggi. Gradien itu sendiri adalah ukuran besarnya perubahan intensitas yang terjadi.

Pada citra digital $f(x,y)$ turunan berarah sepanjang sekian obyek akan bernilai maksimum pada arah normal dari kontur tepian yang bersesuaian. Sifat ini dipergunakan sebagai dasar pemanfaatan operator gradien sebagai deteksi tepi. Gradien suatu citra $f(x,y)$ pada lokasi x,y data didefinisikan sebagai vektor.

$$\begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix} \quad (1)$$

Operator gradien menghitung intensitas perubahan tingkat keabuan dan arah dimana perubahan terjadi. Hal ini dihitung berdasarkan perbedaan nilai pixel-pixel yang bertetangga.

Operator Robert Cross adalah operator gradien yang sederhana berukuran 2 x 2. Operator ini menyediakan pendekatan paling sederhana dari magnitude gradien, yang dapat ditunjukkan dengan fungsi matematis berikut ini, yaitu:

$$G[f(i,j)] = [f(i,j) - f(i+1,j+1)] + [f(i+1,j) - f(i,j+1)] \quad (2)$$

Mask konvolusi untuk Operator Roberts adalah:

1	1
-1	-1

Gambar 2 menunjukkan algoritma deteksi tepi dengan operator Robert yang diikuti dengan proses negasi. Pengurangan 255 terhadap nilai hasil penjumlahan 2 konvolusi dari citra dimaksudkan untuk menegasikan, karena hasil konvolusi biasanya berada di sekitar nilai 0 (nilai kecil), sehingga jika ditampilkan, citra tepi kebanyakan berwarna gelap (hitam).

```

1.Untuk x=1 sampai w-2 kerjakan 2
2.Untuk y=1 sampai h-2 kerjakan 3
3.Kerjakan 4-12
4.Konvolusi1 = 0
5.Untuk u = 0 sampai 1 kerjakan 6
6.Untuk v = 0 sampai 1 kerjakan 7-
7.Konvolusi1= konvolusi1+
mask1[u,v]*nilai_citra_lama[x-u,y-v]
8.Konvolusi2 = 0
9.Untuk u = 0 sampai 1 kerjakan 10
10.Untuk v = 0 sampai 1 kerjakan 11
11.Konvolusi2 = konvolusi2 +
mask2[u,v]*nilai_citra_lama[x-
u,y-v]
12.Nilai_citra_baru[x,y]=255-
(round(abs(konvolusi1)

```

Gambar 2. Algoritma Konvolusi Deteksi Tepi dengan operator Robert dan operasi negasi

3. JARINGAN SYARAF TIRUAN

Jaringan Syaraf Tiruan (JST) adalah sistem pemroses informasi yang memiliki karakteristik unjuk kerja tertentu yang menyerupai jaringan syaraf biologis (Fausett, 1994). JST telah dikembangkan sebagai generalisasi model matematika dari aspek kognitif manusia atau syaraf biologis, yaitu didasarkan pada asumsi-asumsi bahwa :

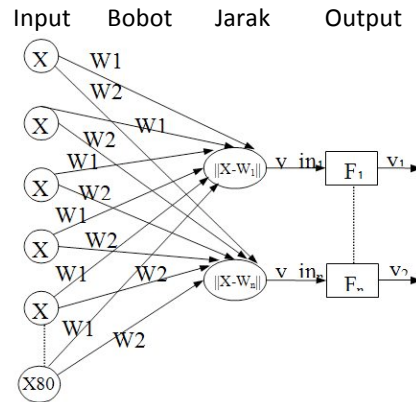
- a. Pemrosesan informasi terjadi pada elemen-elemen yang disebut neuron.
- b. Sinyal-sinyal merambat di antara neuron melalui interkoneksi.
- c. Setiap interkoneksi memiliki bobot yang bersesuaian yang pada kebanyakan jaringan syaraf berfungsi untuk mengalikan sinyal yang dikirim.
- d. Setiap neuron menerapkan fungsi aktivasi (biasanya tidak linear) pada masukan jaringan untuk menentukan sinyal keluarannya

3.1 Learning Vector Quantization untuk pengenalan tanda tangan

Learning Vector Quantization (LVQ) adalah suatu metode untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor input. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor-vektor input. Jika dua vektor input mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor input tersebut ke dalam kelas yang sama. (Kusumadewi, 2003).

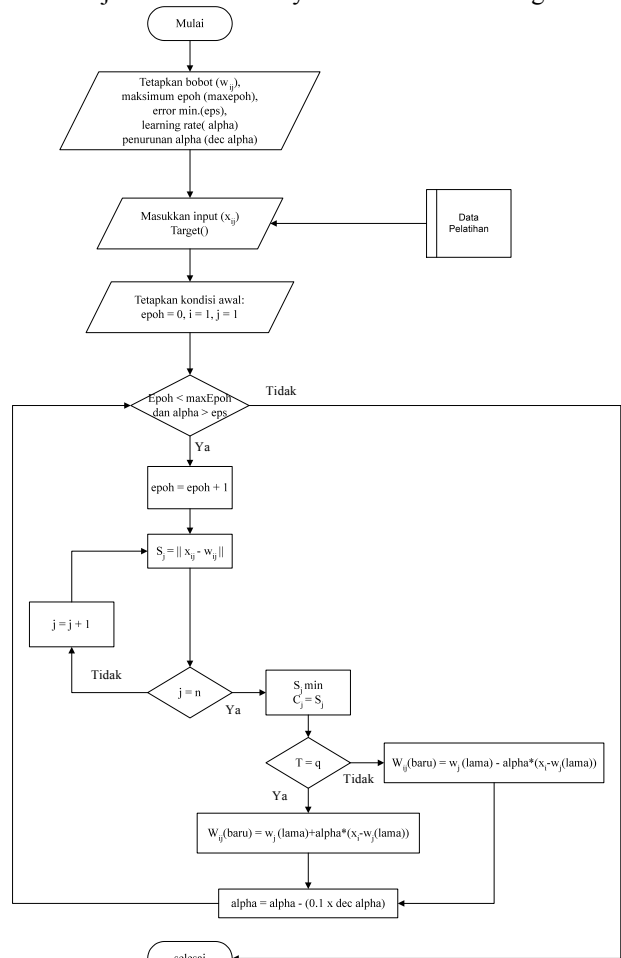
Pada Gambar 3, ditunjukkan arsitektur LVQ yang digunakan pada penelitian ini. Terdapat 80 unit neuron input, dan sejumlah n output. Lapisan input

diambil dari pengolahan citra tandatangan, dan lapisan output merepresentasikan target pemilik tandatangan.



Gambar 3. Arsitektur Jaringan LVQ

Pada jaringan LVQ, pembelajaran atau pelatihan jaringan harus dilakukan terlebih dahulu. Pembelajaran akan menyesuaikan bobot dengan



Gambar 4. Algoritma Pembelajaran Learning Vector Quantization

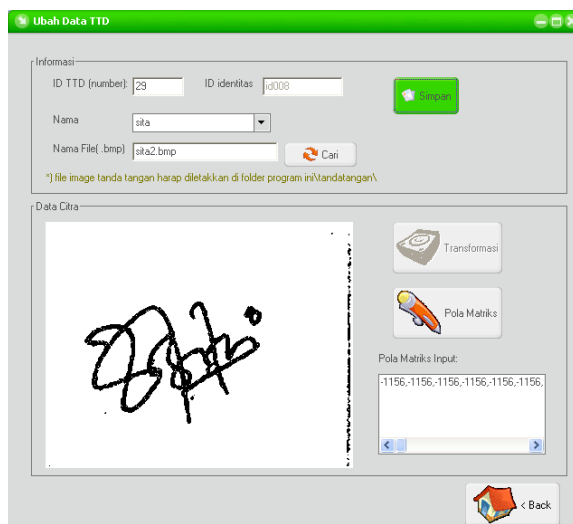
pola-pola yang dipelajari dari data. Algoritma pembelajaran LVQ dapat dilihat pada Gambar 4.

Setelah pembelajaran selesai dijalankan, barulah pengujian bias dilakukan. Pengujian dilakukan dengan menghitung jarak antara input pengujian (citra pengujian yang telah diolah menjadi vektor input) dengan bobot akhir dari masing-masing kelas (output) pemilik tanda tangan. Kelas yang memiliki jarak terdekat dengan vektor input akan menjadi pemenang.

4. IMPLEMENTASI DAN HASIL

4.1 Aplikasi pengenalan tanda tangan

Aplikasi yang dikembangkan dalam penelitian ini menggunakan halaman antar muka pengolahan data, pembelajaran, dan pengujian. Pada halaman pengolahan data, dapat dimasukkan data citra tanda tangan hasil *scanning* ke dalam basis data. Tampilan halaman pengolahan data ditunjukkan pada Gambar 5. Sebelum disimpan ke dalam basis data, citra tanda tangan akan dioleh terlebih dahulu hingga dihasilkan vektor input (pola matriks input) yang merepresentasikan citra tersebut.

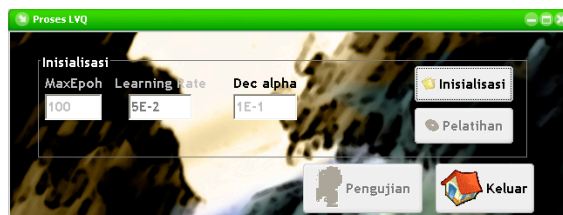


Gambar 5. Tampilan halaman pengolahan data tanda tangan

Tampak pada Gambar 5, citra tanda tangan telah ditransformasi untuk menghapus warna dan garis semu pada citra, sekaligus mengubahnya menjadi bentuk biner, pembagian region, serta pembentukan vektor input.

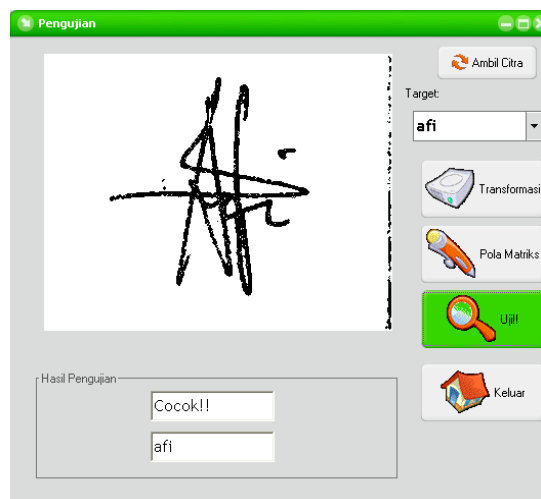
Data yang telah disimpan, dapat digunakan dalam pelatihan jaringan syaraf tiruan LVQ. Proses pelatihan diawali dengan tahap inisialisasi yang tampilannya ditunjukkan pada Gambar 6. Pada halaman ini, kita dapat menentukan parameter-parameter yang digunakan.

Setelah tahap inisialisasi selesai, akan dijalankan prosedur pelatihan menggunakan data tanda tangan yang telah disimpan. Proses ini akan menghasilkan bobot akhir yang mewakili masing-masing kelas pemilik tanda tangan.



Gambar 6. Tampilan halaman pelatihan

Setelah didapat bobot akhir, dapat dilakukan proses pengujian. Ini dilakukan dengan cara mengambil citra yang akan diujikan, mengolah citra tersebut hingga menjadi vektor input pengujian. Kemudian jarak Euclidean antara vektor input tersebut dengan masing-masing bobot kelas akan dihitung. Kelas yang memiliki jarak terkecil dengan vektor input akan dipilih menjadi pemenang. Ini berarti tanda tangan yang diujikan dikenali sebagai tanda tangan kelas pemenang. Gambar 7 menunjukkan tampilan halaman pengujian.



Gambar 7. Tampilan halaman pengujian

4.2 Hasil Pengujian

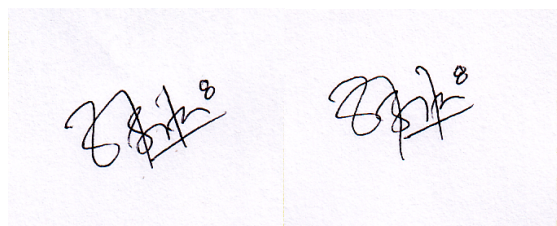
Pengujian aplikasi ini dilakukan dengan menggunakan 54 citra tanda tangan dari 9 orang. Rangkuman hasil pengujian dapat dilihat pada Tabel 1.

Tabel 1. Hasil pengujian aplikasi

No	Pemilik	Jumlah citra	Dikenali dengan benar
1	Afi	6	6
2	Chandra	6	6
3	Dinan	6	6
4	Dini	6	6
5	Dita	6	6
6	Novi	6	6
7	Shenchan	6	6
8	Sita	6	5
9	Yang	6	6

Tingkat ketepatan pengenalan tanda tangan yang dilakukan aplikasi ini terhadap data yang diujikan adalah 98%.

Dari keseluruhan data yang diujikan, terdapat satu citra tanda tangan yang tidak dapat dikenali dengan benar. Hal ini disebabkan oleh posisi tanda tangan yang berbeda dengan posisi pada citra pelatihan. Gambar 8 menunjukkan perbedaan posisi tersebut.



Gambar 8. Salah satu citra tanda tangan yang dilatih (kiri) dan citra pengujian yang tidak dikenali dengan baik (kanan)

Pada Gambar 8, kita dapat melihat bahwa posisi tanda tangan pada citra pengujian lebih ke atas daripada citra yang dilatih.

5. KESIMPULAN DAN SARAN

Dalam penelitian ini, dikembangkan aplikasi pengenalan tanda tangan menggunakan jaringan syaraf tiruan Learning Vector Quantization yang digunakan untuk melatih sejumlah citra tanda tangan. Pada pengujian, tingkat ketepatan pengenalan tanda tangan sebesar 98%.

Terdapat satu citra tanda tangan yang tidak dapat dikenali dengan benar. Kesalahan ini disebabkan oleh perbedaan posisi tanda tangan pada citra. Oleh karena itu, disarankan untuk pengembangan berikutnya, dapat ditambahkan proses normalisasi (pemotongan sisi luar citra) pada citra sehingga kesalahan pengenalan akibat perbedaan posisi ini dapat dihindari.

PUSTAKA

Arif M., dan Vincent N. (2003) *Comparison of Three Data Fusion Methods For An Off-Line Signature Verification Problem*, Laboratoire d'Informatique, Université de François Rabelais

Chalechale A., dan Mertins A. (2003) *Line Segment Distribution of Sketches for Persian Signature Recognition*, IEEE Proc. TENCON, vol. 1, pp. 11-15

Fausett, L., 1994, *Fundamentals of Neural Network: Architectures, Algorithms, and Applications*, Prentice-Hall, Inc., New Jersey.

Justino E.J.R., Bortolozzi F., Sabourin R. (2001) *Off-line Signature Verification Using HMM for Random, Simple, and Skilled Forgeries*, ICDAR

2001, International Conference on Document Analysis and Recognition, vol. 1, pp. 105-110

Kusumadewi, S. (2003) *Artificial Intelligence (Teknik dan Aplikasinya)*, Graha Ilmu, Yogyakarta

Low, A., (1991) *Introduction Computer Vision and Image Processing*, Mc Graw Hill Book Company, London.

OZ C., Ercal F., Demir Z. (2003) *Signature Recognition and Verification with ANN*, Third International Conference on Electrical and Electronics Engineering, Bursa, Turkey

Özgündüz E., Şentürk T., Karşlıgil M. E. (2005) *Off-line Signature Verification and Recognition by Support Vector Machine*, 13th European Signal Processing Conference, Antalya, Turkey

Sansone and Vento (2000) *Signature Verification: Increasing Performance by a Multi-Stage System*, Pattern Analysis & Applications, vol. 3, pp. 169-181

Sigit R., Basuki A., Ramadijanti N., Pramadihanto D. (2005) *Step by Step Pengolahan Citra Digital*, Penerbit Andi, Yogyakarta

Vélez J. F., Sánchez A., Moreno A. B., (2003) *Robust Off-Line Signature Verification Using Compression Networks And Positional Cuttings*, Proc. 2003 IEEE Workshop on Neural Networks for Signal Processing, vol. 1, pp. 627-636