

Pengenalan Tulisan Tangan Angka menggunakan *Self Organizing Maps* (SOM)

Gita Fadila Fitriana

Fakultas Informatika, Rekayasa Perangkat Lunak, Institut Teknologi Telkom Purwokerto, Jawa Tengah, Indonesia

Email: gita@ittelkom-pwt.ac.id

Submitted: 24/06/2021; Accepted: 29/06/2021; Published: 30/06/2021

Abstrak—Tulisan tangan merupakan pengenalan pola karakter. Pengenalan pola karakter sangat menarik untuk dilakukan penelitian. Dalam pengenalan pola karakter, banyak jenis karakter dapat dikenali oleh komputer dan diselesaikan dengan berbagai algoritma. Berbagai macam Pengenalan pola telah berhasil diterapkan di berbagai bidang seperti pengenalan suara, deteksi wajah, pengenalan sidik jari, dan pengenalan tulisan tangan. Pengenalan tulisan tangan dibagi menjadi dua macam yaitu pengenalan tulisan tangan secara online dan pengenalan tulisan tangan secara offline. Pengenalan tulisan tangan secara online membutuhkan peralatan elektronik khusus dan tulisan tangan ditangkap dengan tablet yang sensitive terhadap tekanan. Pengenalan tulisan tangan secara offline tidak memerlukan mesin khusus dikarenakan data tulisan tangan yang dimasukkan dari teks tertulis sebelumnya seperti gambar scan oleh scanner. Beberapa metode yang dikembangkan untuk mengenali tulisan tangan dengan tingkat akurasi yang berbeda-beda. Penelitian ini menggunakan ekstraksi ciri United Moment Invariant (UMI) dan Self Organizing Maps (SOM). Berdasarkan hasil percobaan perangkat lunak untuk keseluruhan data set pengujian, data primer menghasilkan tingkat akurasi sebesar 88% untuk 50 gambar dan data sekunder pertama menghasilkan tingkat akurasi sebesar 98,2% untuk 500 gambar. Namun untuk percobaan data sekunder kedua sebanyak 50 data uji, tingkat akurasi sebesar 90%. Percobaan data sekunder ketiga sebanyak 500 data uji, tingkat akurasi sebesar 89%. Bila dilihat dari nilai akurasi tersebut, maka data primer memiliki tingkat akurasi yang lebih rendah bila dibandingkan dengan kedua data sekunder dengan jumlah yang berbeda. Tingkat akurasi yang dihasilkan dari percobaan data set yang berbeda membuktikan bahwa karakter tulisan tangan memiliki tingkat variasi yang tinggi dan tidak konsisten. Hal ini disebabkan oleh ketebalan dan bentuk tulisan yang tidak konsisten di setiap orang serta kebiasaan yang mempengaruhi karakter tulisan tangan seseorang. Data primer merupakan data yang diambil langsung dan melalui proses pemindaian serta masih memiliki banyak derau pada citra tulisan tangan angka. Sedangkan data sekunder sudah mengalami proses citra keabuan sehingga citra tulisan tangan bersih dari derau.

Kata Kunci: SOM; UMI; Tulisan Tangan; Akurasi; Data Uji

Abstract—Handwriting is character pattern recognition. Character pattern recognition is exciting to do research. In character pattern recognition, many types of characters can be recognized by computers and solved by various algorithms. Various kinds of pattern recognition have been successfully applied in multiple fields such as voice recognition, face detection, fingerprint recognition, and handwriting recognition. Handwriting recognition is divided into two types, namely online handwriting recognition and offline handwriting recognition. Online handwriting recognition requires special electronic equipment, and handwriting is captured on a pressure-sensitive tablet. Offline handwriting recognition does not need a particular machine because handwriting data is entered from previously written text such as images scanned by a scanner. Several methods have been developed to recognize handwriting with varying degrees of accuracy. This research uses the feature extraction of United Moment Invariant (UMI) and Self Organizing Maps (SOM). Based on the results of the software experiment for the entire test data set, the primary data yielded an accuracy rate of 88% for 50 images, and the first secondary data paid an accuracy rate of 98.2% for 500 images. However, for the second secondary data experiment with 50 test data, the accuracy rate is 90%. The third secondary data experiment was 500 test data. The accuracy rate was 89%. When viewed from the accuracy value, the primary data has a lower level of accuracy when compared to the two secondary data with different amounts. The story of accuracy resulting from experimenting with varying data sets proves that handwritten characters have a high and inconsistent level of variation. This is caused by the thickness and form of writing that is not consistent in each person and habits that affect the character of one's handwriting. Primary data is data that is taken directly and through a scanner process and still has a lot of noise in the handwritten image of numbers. At the same time, the secondary data has undergone a grey image process so that the handwritten image is clean from noise.

Keywords: SOM; UMI; Handwriting; Accuration; Test Data

1. PENDAHULUAN

Tulisan tangan merupakan pengenalan pola karakter. Pengenalan pola karakter sangat menarik untuk dilakukan penelitian. Dalam pengenalan pola karakter, banyak jenis karakter dapat dikenali oleh komputer dan diselesaikan dengan berbagai algoritma. Berbagai macam Pengenalan pola telah berhasil diterapkan di berbagai bidang seperti pengenalan suara, deteksi wajah, pengenalan sidik jari, dan pengenalan tulisan tangan [1]. Pengenalan tulisan tangan dibagi menjadi dua macam yaitu pengenalan tulisan tangan secara online dan pengenalan tulisan tangan secara offline [2]. Pengenalan tulisan tangan secara online membutuhkan peralatan elektronik khusus dan tulisan tangan ditangkap dengan tablet yang sensitive terhadap tekanan. Pengenalan tulisan tangan secara offline tidak memerlukan mesin khusus dikarenakan data tulisan tangan yang dimasukkan dari teks tertulis sebelumnya seperti gambar scan oleh scanner.

Beberapa metode yang dikembangkan untuk mengenali tulisan tangan dengan tingkat akurasi yang berbeda-beda. Metode tersebut meliputi: pengenalan pola angka tulisan tangan menggunakan Backpropagation dengan tingkat akurasi sebesar 70,7% [3], pengenalan tulisan tangan angka dinamis menggunakan jaringan syaraf tiruan persepton multi layer dengan tingkat akurasi sebesar 83,9% [4] dan pengenalan angka tulisan tangan secara online menggunakan *Support Vector Machine* (SVM) dengan tingkat akurasi sebesar 96,8% [5].



Pengenalan pola karakter tulisan tangan sulit dikenali karena sifat karakter tulisan tangan yang selalu mengalami perubahan dengan berbagai variasi bentuk distorsi. Tulisan tangan tidak mudah dikenali oleh komputer. Komputer membutuhkan beberapa proses seperti pra-pengolahan, ekstraksi ciri dan klasifikasi untuk dapat mengenali tulisan tangan. Fakta menunjukkan bahwa hingga saat ini belum ada metode yang dapat mengenali tulisan tangan dari berbagai bahasa manapun dengan tingkat keakuratan 100%.

Tahapan pertama pada pengenalan tulisan tangan adalah tahap prapengolahan yang meliputi penghilangan noise, nilai ambang, dan sebagainya. Pada tahapan kedua yaitu ekstraksi ciri, pemilihan ekstraksi ciri yang tepat dapat menjadi faktor penting dalam mencapai kinerja pengenalan yang akurat. Metode yang dapat digunakan untuk pengenalan tulisan tangan salah satunya adalah *United Moment Invariant* (UMI). Kelebihan metode UMI dapat diterapkan dalam kondisi diskrit dan juga dapat diterapkan berdasarkan wilayah dan batas serta meningkatkan kecepatan perhitungan ekstraksi ciri tulisan tangan. Pada penelitian pengenalan mengenai tulisan tangan huruf Devnagri menggunakan ekstraksi ciri UMI, akurasi pengenalan mencapai 94,56% [6]. Penelitian tentang pengenalan karakter tulisan tangan telah ada sejak tahun 1980-an. Pengenalan tulisan tangan menggunakan metode pengelompokan sangat penting seperti mengenali kode pos pada surat untuk surat pos penyortiran, memproses sejumlah cek di bank, entri numerik secara manual misalnya formulir pajak dan seterusnya [7]. Pengenalan tulisan tangan bertugas mengubah bahasa yang diwakili dalam bentuk spasial tanda grafis ke dalam representasi simbolik [8].

Sebuah sistem pengenalan dapat berupa "online" atau "offline". Sistem pengenalan online apabila jika urutan temporal poin ditelusuri oleh pena yang tersedia, seperti elektronik asisten data pribadi yang membutuhkan pengguna untuk menulis pada layar dengan menggunakan stylus. Sistem pengenalan offline apabila jika diterapkan pada teks tertulis sebelumnya, seperti setiap gambar scan oleh scanner. Masalah online biasanya lebih mudah daripada masalah secara offline karena informasi lebih lanjut tersedia pada sistem pengenalan online [9][10]. Penelitian ini menggunakan tulisan tangan secara offline.

Pengenalan tulisan tangan offline umumnya diamati lebih sulit daripada pengenalan tulisan tangan secara online. Proses pengenalan standar dasarnya sama: urutan fitur yang diekstraksi dari data, kemudian dicocokkan dengan urutan label [11]. Dalam kasus offline hanya gambar teks saja yang tersedia. Karena lebih mudah untuk penggalan fitur yang relevan.

Perbedaan lainnya adalah antara mengenali karakter terisolasi atau katakata, dan pengenalan seluruh jajaran teks [12]. Contoh implementasi adalah rekayasa perangkat lunak Optical Offline Character Recognition (OCR) pada scanner yang bekerja pada karya-karya hasil pindai. Masalah yang terdapat pada tulisan tangan adalah ketidakpastian seperti variasi dalam tulisan tangan selama periode waktu, kesamaan dalam teks, variasi dalam gaya penulisan. Ini telah membuat tulisan tangan teeks menjadi wilayah kerja penelitian [13]. Offline Handwritten Digit Recognition Using Neural Network, penelitian [14][15] menyajikan pengenalan tulisan tangan angka secara offline. Ekstraksi ciri global features extraction yang digunakan pada penelitian ini dan jaringan syaraf tiruan multi layer perceptron untuk mengenali tulisan tangan, penelitian ini menggunakan data latih sebanyak 250 data dan data uji sebanyak 500 data dan menghasilkan tingkat akurasi sebesar 91,2%. Penelitian pengenalan tulisan tangan angka menggunakan metode *Self Organizing Maps* (SOM) dengan akurasi pengenalan mencapai 85,74% [16]. Berdasarkan uraian di atas, pengembangan penelitian tulisan tangan belum optimal dengan akurasi yang tinggi mencapai 100%. Untuk menyelesaikan masalah tersebut, maka akan dikembangkan sebuah perangkat lunak untuk mengenali tulisan tangan yang berfokus pada angka dengan menerapkan metode UMI dan SOM.

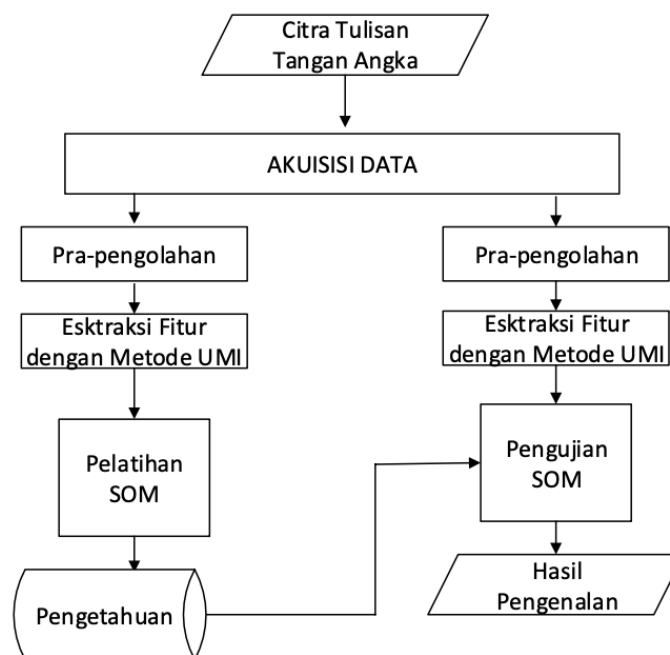
2. METODOLOGI PENELITIAN

2.1 Analisis Akuisisi Data Citra Tulisan Tangan

Objek pengenalan pada tugas akhir ini adalah data tulisan tangan berbentuk citra. Jenis data yang digunakan berupa data primer dan data sekunder. Data primer diperoleh langsung dari tulisan tangan beberapa mahasiswa. Data primer diambil dari 5 mahasiswa, masing-masing mahasiswa menuliskan angka 0-9 sebanyak 10 kali, sehingga menghasilkan 500 buah tulisan tangan angka. Mahasiswa menuliskan angka tersebut pada sebuah form pengambilan data primer dapat dilihat pada Tahap selanjutnya, citra tulisan tangan dilakukan proses scanner. Hasil scanner berupa citra tulisan tangan angka dalam bentuk digital dengan format JPEG dan resolusi 106 x 97 piksel. Kemudian citra tulisan tangan angka tersebut diubah menjadi format BMP dan resolusi 28x28 piksel. Karena, format BMP tidak terkompresi pikselnya. Sehingga, nilai piksel yang tersimpan masih murni. Sedangkan, data sekunder diperoleh dari MNIST data set (<http://yann.lecun.com/exdb/mnist>). Citra tulisan tangan angka dari MNIST berupa format BMP dan resolusi 28x28 piksel. Proses perubahan data primer berupa format dan resolusi, agar citra yang diproses menghasilkan pengenalan yang baik. Maka tahapan selanjutnya dilakukan pra-pengolahan.

2.2 Tahapan Penelitian

Masalah pengenalan tulisan tangan dalam penelitian ini adalah bagaimana mengenali tulisan tangan khususnya angka (0-9). Komputer harus melalui beberapa proses untuk melakukan pengenalan. Tahapan proses pengenalan tulisan tangan angka yaitu citra tulisan tangan khususnya angka, pra-pengolahan, ekstraksi fitur dan metode pengelompokan. Berikut diagram proses pengenalan tulisan tangan angka dengan SOM disajikan pada Gambar 1.

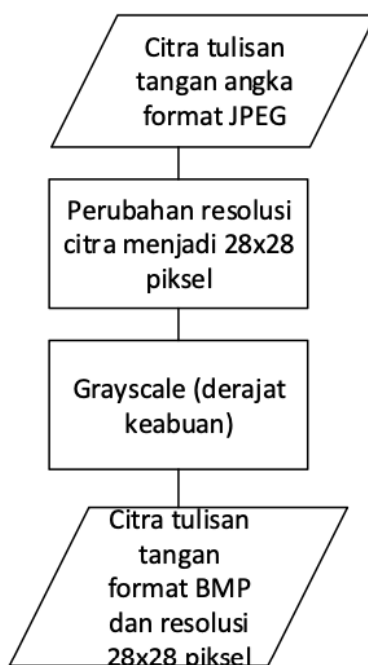


Gambar 1. Diagram Proses Pengenalan Tulisan Tangan Angka dengan SOM

Pada Gambar 1, diagram proses pengenalan citra tulisan tangan dimulai dari tahap akuisisi data. Citra tulisan tangan angka diproses menggunakan alat scanner. Kemudian tahap selanjutnya yaitu tahapan pra-pengolahan. Tahapan pra-pengolahan ini digunakan untuk memperbaiki citra tulisan tangan angka dari derau agar citra tulisan tangan angka dapat dikenali. Setelah tahapan prapengolahan, nilai dari hasil tahapan pra-pengolahan menjadi input pada tahapan ekstraksi fitur menggunakan metode UMI. Kemudian, proses pengelompokan citra tulisan tangan angka menggunakan SOM. SOM terdapat dua proses dalam pengelompokan yaitu pelatihan dan pengujian. Tahapan awal SOM yaitu melatih citra tulisan tangan angka. Setelah citra tulisan tangan dilatih, kemudian pengetahuan SOM dari hasil pelatihan disimpan. Pengetahuan SOM tersebut digunakan untuk proses pengujian SOM. Sehingga, citra tulisan tangan angka dapat dikenali.

2.3 Analisis Pra-pengolahan

Citra tulisan tangan harus terlebih dahulu melalui proses pra-pengolahan sebelum dilakukan tahapan ekstraksi ciri dan dikelompokkan dengan SOM. Berikut tahapan proses pra-pengolahan pada data primer pada Gambar 2.



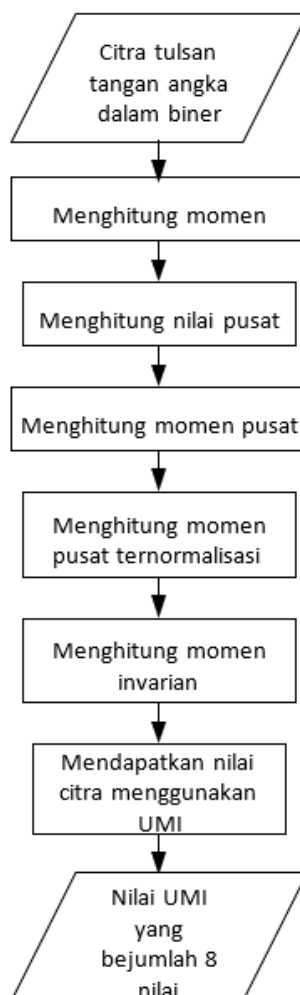
Gambar 2. Tahapan Pra-pengolahan Data Primer

Pada tahapan pra-pengolahan, format awal citra tulisan tangan data primer yaitu JPEG dan resolusi 106x97 piksel. Kemudian, perubahan resolusi menjadi 28x28 piksel menggunakan perangkat lunak Adobe Photoshop CS3. Tahapan selanjutnya grayscale (derajat keabuan) menggunakan perangkat lunak Adobe Photoshop CS3. Citra tulisan tangan menggunakan resolusi menjadi 28x28 piksel dan grayscale agar sama dengan data sekunder yang diperoleh dari MNIST data set. Selanjutnya, data diubah formatnya menjadi BMP agar nilai piksel murni karena tidak terkompresi nilai piksel pada BMP. Semua nilai piksel tersebut nantinya akan dilakukan proses thresholding. Proses thresholding mengubah warna hitam dan putih. Warna hitam bernilai 1 dan putih bernilai 0. Namun, tahapan thresholding tidak menggunakan perangkat lunak Adobe Photoshop CS3. Setelah tahapan thresholding, selanjutnya proses ekstraksi fitur dibahas pada subbab analisis ekstraksi fitur menggunakan UMI. Berikut Gambar 3 menunjukkan citra tulisan tangan yang telah dilakukan tahapan thresholding.



Gambar 3. (a). Citra Tulisan Tangan Tanpa Prapengolahan dan Citra Tulisan Tangan Prapengolahan

Tahapan selanjutnya ekstraksi fitur, menggunakan UMI. Jumlah nilai UMI sebanyak 8 nilai didapatkan untuk setiap citra tulisan tangan. Diagram alir ekstraksi fitur tulisan tangan angka menggunakan UMI dapat dilihat pada Gambar 4. Gambar 4 merupakan langkah-langkah terperinci dari proses ekstraksi fitur tulisan tangan angka.



Gambar 4. Langkah-langkah Ekstraksi Fitur dengan Metode UMI



Berikut penjelasan dari gambar 4:

1. Citra tulisan tangan dalam biner

Pada tahapan ini masukan citra untuk perhitungan UMI menggunakan citra tulisan tangan dalam bentuk biner. Proses menentukan nilai 0 dan 1 menggunakan Persamaan 1.

$$f(x, y) = \begin{cases} 1, I(x, y) \geq D \\ 0, I(x, y) < D \end{cases} \quad (1)$$

Nilai $f(x, y)$ akan menjadi 1 apabila nilai $I(x, y)$ melebihi dalam batasan nilai ambangnya, apabila nilai $I(x, y)$ kurang dari batasan nilai ambang maka $f(x, y)$ akan bernilai 0.

2. Menghitung momen

Pada tahapan ini dilakukan perhitungan nilai momen dari suatu citra tulisan tangan. Nilai momen yang akan dihitung yaitu nilai momen m_{00} , m_{10} , dan m_{01} .

3. Menghitung titik pusat

Setelah diketahui nilai momen maka proses selanjutnya adalah menghitung nilai titik pusat, dimana nilai titik pusat ini yang akan menghasilkan nilai \bar{x} dan \bar{y} dari sebuah citra tulisan tangan.

4. Menghitung momen pusat

Proses menghitung momen pusat ini dilakukan setelah mendapatkan nilai momen, dan nilai titik pusat sebuah citra tulisan tangan. Perhitungan momen pusat akan menghasilkan nilai μ'_{20} , μ'_{02} , μ'_{11} , μ'_{30} , μ'_{03} , μ'_{21} , dan μ'_{12} .

5. Menghitung momen pusat ternormalisasi

Pada proses ini, akan dilakukan perhitungan momen pusat ternormalisasi yang akan menghitung nilai η_{20} , η_{02} , η_{11} , η_{30} , η_{03} , η_{21} , dan η_{12} . Perhitungan ini dapat dilakukan apabila nilai-nilai dari perhitungan momen pusat sudah diketahui.

6. Menghitung momen invarian

Proses perhitungan momen invarian ini, akan dilakukan suatu perhitungan yang menghitung nilai ϕ_1 , ϕ_2 , ϕ_3 , ϕ_4 , ϕ_5 , ϕ_6 , dan ϕ_7 . Sebelum perhitungan momen invarian dilakukan, nilai momen invarian ternormalisasi harus didapatkan terlebih dahulu.

7. Mendapatkan ciri citra dengan UMI

Perhitungan ini adalah proses akhir dari proses UMI yaitu menghitung nilai UMI, nilai UMI didapat dari perhitungan θ_1 , θ_2 , θ_3 , θ_4 , θ_5 , θ_6 , θ_7 , dan θ_8 . Perhitungannya akan menghasilkan delapan buah nilai dari masing-masing θ (Theta), dimana persamaan menghitung UMI.

8. Nilai UMI yang berjumlah 8 nilai

Hasil dari proses ekstraksi fitur UMI mendapatkan 8 nilai setiap citra tulisan tangan angka. 8 nilai tersebut menjadi *input neuron* pada proses pengelompokan menggunakan SOM.

Dalam pengenalan pola dan pengolahan citra untuk membedakan bentuk merupakan hal yang sangat penting. Karena keunikan pencahayaan, sudut pandang, ruang posisi, skala, dan lokasi bentuk dalam gambar harus diubah. Jadi diperlukan untuk menemukan nilai-nilai fitur, serta harus menjaga RTS (rotasi, translasi dan skala) invarian dalam kondisi apapun. Invarian adalah alat yang sangat berguna untuk pengenalan pola [17].

2.4 Analisis Metode Self Organizing Maps

Algoritma Self Organizing Maps melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tetapi tidak sama) dengan pola yang dipakai selama pelatihan. Algoritma SOM memiliki keuntungan yaitu hasil pengenalan tulisan tangan menggunakan SOM menunjukkan hasil yang baik serta waktu pelatihan yang cepat (Agarwal, 2011).

2.4.1 Proses Pelatihan SOM

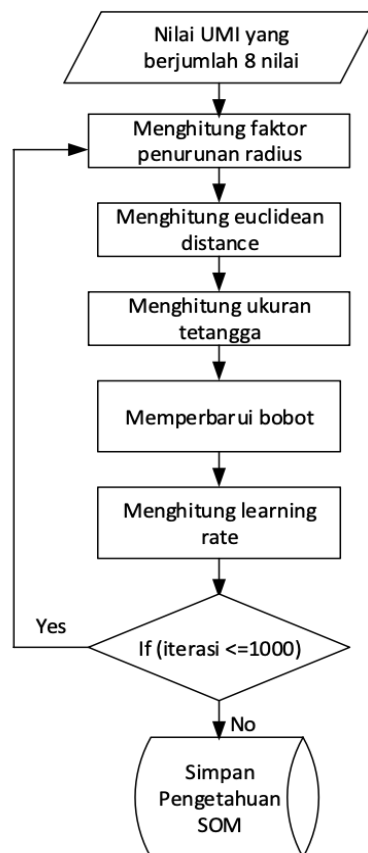
Pola untuk proses pelatihan citra tulisan tangan hasil ekstraksi fitur UMI menghasilkan 8 nilai. 8 nilai UMI yang menjadi masukan untuk SOM. Berikut Tabel 1 memperlihatkan atribut yang digunakan pada SOM untuk pengenalan citra tulisan tangan angka.

Tabel 1. Atribut JST SOM

Atribut	Nilai
Input Neuron	8
Output/Kohonen Neuron	1600 Neuron
Learning Rate	0.6
Batas Maksimum Iterasi	1000 iterasi

Output neuron berjumlah 1600 neuron. Hal ini dikarenakan bentuk SOM bertopologi dengan bentuk topologi rectangular. Apabila ukuran peta dengan tinggi dan lebar sebesar 40x40 maka neuron output sebanyak 1600 neuron. Ukuran peta dipengaruhi oleh jumlah data yang dilatih. Semakin banyak data yang dilatih maka semakin besar ukuran

peta SOM. Learning rate yang digunakan sebesar 0.6. Pemilihan nilai untuk kecepatan belajar harus seoptimal mungkin agar didapatkan proses belajar yang cepat. Apabila menggunakan learning rate kurang dari 0.6 maka hasil *cluster* tidak maksimal sebab banyak data citra tulisan tangan yang salah mengenali. Tetapi, apabila nilai learning rate lebih dari 0.6 maka hasil *cluster* akan banyak muncul NULL atau tidak dikenali sama sekali. Batas maksimum iterasi sebanyak 1000, karena pelatihan SOM belum menemukan kondisi yang sesuai jika jumlah iterasi lebih kecil dari 1000. Berdasarkan atribut dapat dijadikan parameter untuk melakukan proses pelatihan SOM. Berikut Gambar 5 langkah-langkah SOM.



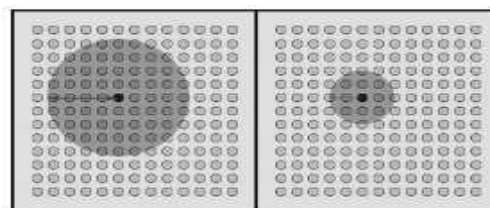
Gambar 5. Langkah-langkah SOM

1. Nilai UMI yang berjumlah 8 nilai

Nilai UMI tersebut menjadi *input neuron* pada SOM untuk proses pelatihan citra tulisan tangan angka.

2. Menghitung faktor penurunan radius

Proses perhitungan faktor penurunan radius untuk menentukan daerah di sekitar bobot pemenang. Pada rumusan tersebut terdapat ketetapan waktu. Tujuan ketetapan waktu agar seiring jalannya iterasi, maka *learning rate* mengecil dan radius makin mengecil. Hal ini dapat memperkecil jangkauan bobot pemenang untuk memperbarui bobot tetangga disekitarnya. Berikut Gambar 6 merupakan gambar penurunan radius.



Gambar 6. Penurunan Radius

3. Menghitung *euclidean distance*

Setelah didapatkan nilai *input neuron* berupa nilai UMI sebanyak 8 nilai. Node yang memiliki jarak minimum antara *input neuron* dengan bobot dinyatakan sebagai BMU (*Best Matching Unit*). Nilai bobot yang digunakan pada iterasi pertama adalah nilai bobot *random*. Untuk iterasi selanjutnya, perhitungan Euclidean distance menggunakan bobot terakhir dari hasil proses memperbarui bobot.

4. Menghitung ukuran tetangga

Setelah diperoleh BMU, kemudian menghitung ukuran tetangga untuk mengukur jarak BMU dengan bobot tetangga yang akan diperbarui bobotnya.

5. Memperbarui bobot

Setelah diperoleh jarak antara BMU dengan bobot tetangga yang akan diperbarui, lalu bobot tetangga tersebut diperbarui bobotnya.

6. Menghitung *learning rate*

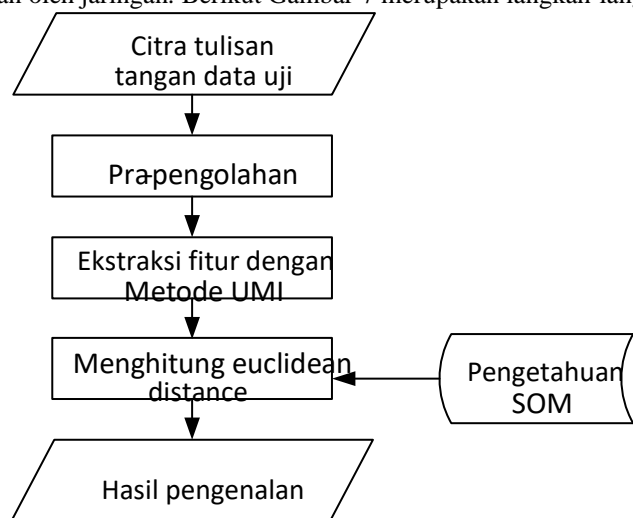
Setelah semua bobot diperbarui maka proses iterasi pertama telah selesai. Kemudian nilai *learning rate* juga diperbarui. Nilai *learning rate* semakin menurun seiring berjalannya iterasi hingga mencapai nilai 0. Jika iterasi belum mencapai 1000, maka dilakukan kembali proses tahap pertama hingga menghitung *learning rate* sampai iterasi terakhir.

7. Simpan pengetahuan SOM

Setelah tahapan SOM selesai hingga iterasi ke 1000, maka hasil bobot terakhir disimpan sebagai pengetahuan. Pengetahuan ini digunakan untuk proses pengujian SOM.

2.4.2 Proses Pengujian SOM

Proses pada fase pengujian hampir sama dengan proses pelatihan hanya saja pada proses pengujian, hanya pencarian winning neuron yang akan dilakukan oleh jaringan. Berikut Gambar 7 merupakan langkah-langkah pengujian SOM.



Gambar 7. Langkah-langkah Pengujian SOM

Citra tulisan tangan data uji berbeda dengan data latih. Pola tulisan tangan yang dikenali diproses dengan tahap pra-pengolahan dan ekstraksi fitur dengan metode UMI terlebih dahulu. Hasil ekstraksi fitur berupa 8 nilai UMI menjadi masukan untuk *input neuron* pada SOM.

Pada pengujian SOM, tahapan yang dilakukan menghitung Euclidean distance. Node yang memiliki jarak minimum antara *input neuron* citra yang akan dikenali dengan bobot. Bobot didapatkan dari pengetahuan yang merupakan hasil dari proses pelatihan SOM yang dapat dilihat pada Gambar 7. Node yang memiliki jarak minimum dinyatakan sebagai pemenang. Pola yang dimiliki neuron pemenang adalah representasi dari hasil pengenalan SOM.

3. HASIL DAN PEMBAHASAN

Perangkat lunak yang dibangun mampu melakukan ekstraksi ciri, pembelajaran dan pengenalan terhadap citra tulisan tangan. Setiap proses yang dilakukan agar perangkat lunak dapat berjalan dengan baik. Untuk melakukan pengenalan tulisan tangan, citra tulisan tangan yang akan dilatih diberikan label berupa nama yang akan dikenali.

3.1 Aktor dan Tujuan

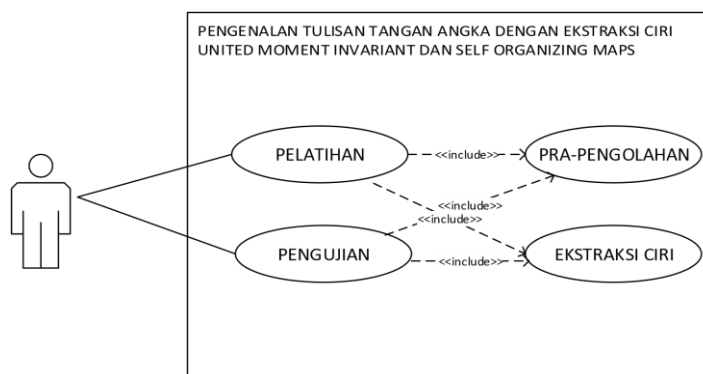
Aktor yang menggunakan perangkat lunak adalah pengguna. Berikut deskripsi aktor dapat dilihat pada Tabel 2.

Tabel 2. Aktor dan Deskripsi

Aktor	Deskripsi
Pengguna	Aktor dengan peran ini mempunyai wewenang untuk melakukan ekstraksi, melakukan pelatihan dan pengenalan terhadap citra tulisan tangan.

3.2 Diagram Use Case

Diagram *use case* adalah diagram yang digunakan untuk menggambarkan nama-nama *use case* dan aktor, dan hubungannya. Diagram *use case* dapat dilihat pada Gambar 8 berikut :



Gambar 8. Diagram Use Case Pengenalan Tulisan Tangan

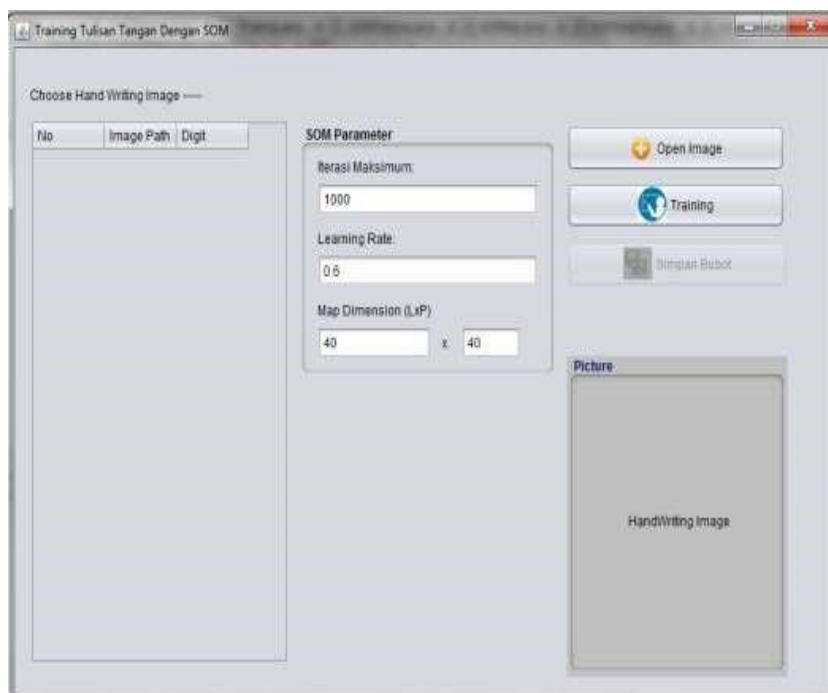
3.3 Implementasi Antar Muka

Implementasi antar muka pada perangkat lunak meliputi fUtama, fTraining dan fTesting. Gambar 9 menunjukkan halaman utama yaitu fUtama.



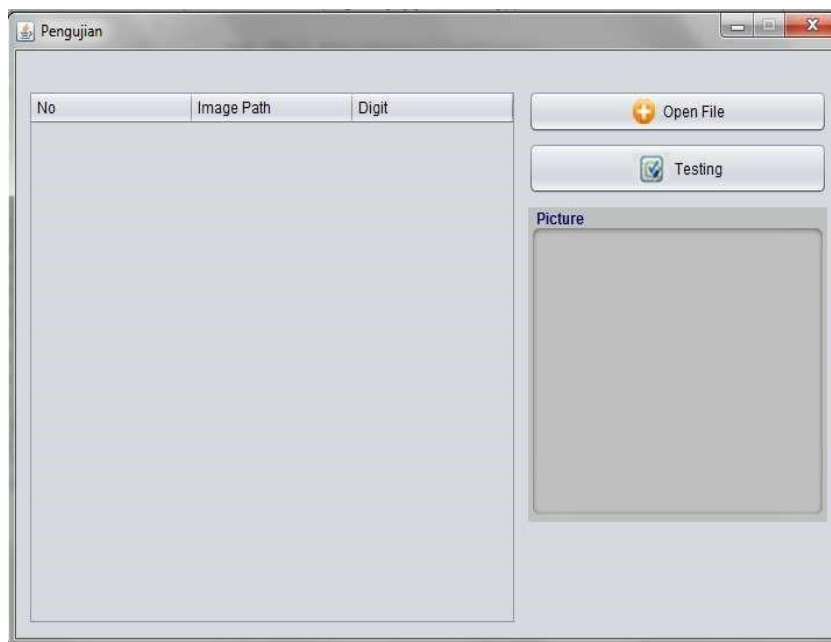
Gambar 9. Antar Muka Form Utama

Gambar 9 menunjukkan hasil implementasi antar muka perangkat lunak untuk Menu fTraining.



Gambar 10. Antar Muka Menu Form Pelatihan

Gambar 10 menunjukkan hasil implementasi antar muka perangkat lunak untuk Menu FormTesting.



Gambar 11. Antar Muka Menu Form Penguujian

3.4 Hasil Percobaan Perangkat Lunak

Pada subbab sebelumnya telah dijelaskan bahwa percobaan dilakukan dengan menggunakan data primer dan data sekunder angka tulisan tangan. Untuk itu pada subbab ini akan dijabarkan hasil dari percobaan kedua data tersebut.

3.4.1 Hasil Percobaan Data Primer

Percobaan data primer dilakukan dengan menggunakan 500 data gambar angka yang ditulis tangan. Dalam pengujian data primer, 500 data tersebut dibagi menjadi 450 data latih dan 50 data uji, setiap angka 0-9 terdiri dari 5 data uji. Hasil akurasi pengenalan untuk data primer pada data uji dapat dilihat pada Tabel 6

Tabel 3. Hasil Akurasi Pengenalan Data Primer pada Data Uji

No	Nama	Total Data Uji	Hasil Akurasi Pengenalan		Akurasi (%)
			Dikenali	Tidak Dikenali	
1	Angka 0	5	5	0	100
2	Angka 1	5	3	2	60
3	Angka 2	5	5	0	100
4	Angka 3	5	3	2	80
5	Angka 4	5	4	1	60
6	Angka 5	5	4	1	80
7	Angka 6	5	5	0	100
8	Angka 7	5	5	0	100
9	Angka 8	5	5	0	100
10	Angka 9	5	5	0	100
Total			44	6	88

3.4.2 Hasil Percobaan Data Sekunder

Percobaan data sekunder dilakukan dengan 3 jumlah data yang berbeda. Data sekunder pertama (SD-1) dilakukan dengan menggunakan 1500 gambar angka tulisan tangan dari basis data MNIST. Gambar yang digunakan dibagi menjadi 2 data yaitu 1000 data latih dan 500 data uji, setiap angka 0-9 terdiri dari 50 data uji. Data sekunder kedua (SD-2) dilakukan dengan menggunakan 500 gambar angka tulisan tangan. Gambar yang digunakan dibagi menjadi 2 data yaitu 450 data latih dan 50 data uji. Data sekunder ketiga (SD-3) dilakukan dengan menggunakan 1000 data gambar angka tulisan tangan. Gambar yang digunakan dibagi menjadi 2 data yaitu 500 data latih dan 500 data uji. Hasil akurasi pengenalan data sekunder pertama (SD-1) pada data uji dapat dilihat pada Tabel 7, data sekunder kedua (SD-2) pada data uji dapat dilihat pada Tabel 8, dan data sekunder ketiga (SD-3) pada data uji dapat dilihat pada Tabel 9.

Tabel 4. Hasil Akurasi Pengenalan Data Sekunder Pertama (SD-1) pada Data Uji.

No	Nama	Total Data Uji	Hasil Akurasi Pengenalan		Akurasi (%)
			Dikenali	Tidak Dikenali	
1	Angka 0	50	50	0	100
2	Angka 1	50	45	5	90
3	Angka 2	50	50	0	100
4	Angka 3	50	50	0	100
5	Angka 4	50	48	2	96
6	Angka 5	50	50	0	100
7	Angka 6	50	49	1	98
8	Angka 7	50	49	1	98
9	Angka 8	50	50	0	100
10	Angka 9	50	50	0	100
Total			491	9	98,2

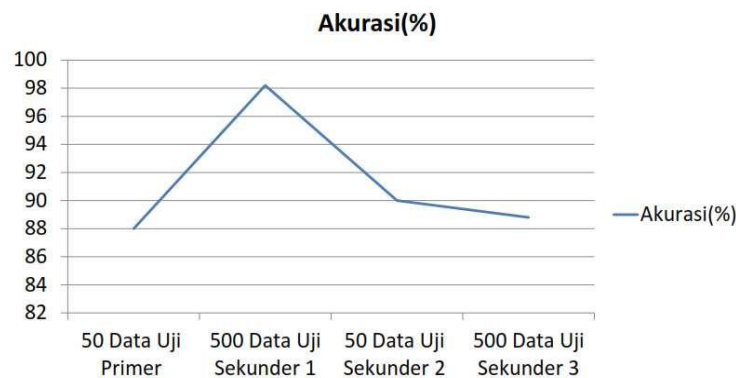
Tabel 5. Hasil Akurasi Pengenalan Data Sekunder Kedua (SD-2) pada Data Uji.

No	Nama	Total Data Uji	Hasil Akurasi Pengenalan		Akurasi (%)
			Dikenali	Tidak Dikenali	
1	Angka 0	5	5	0	100
2	Angka 1	5	4	1	80
3	Angka 2	5	5	0	100
4	Angka 3	5	4	1	80
5	Angka 4	5	3	2	60
6	Angka 5	5	5	0	100
7	Angka 6	5	4	1	80
8	Angka 7	5	5	0	100
9	Angka 8	5	5	0	100
10	Angka 9	5	5	0	100
Total			45	5	90

Tabel 6. Hasil Akurasi Pengenalan Data Sekunder Ketiga (SD-3) pada Data Uji.

No	Nama	Total Data Uji	Hasil Akurasi Pengenalan		Akurasi (%)
			Dikenali	Tidak Dikenali	
1	Angka 0	50	42	8	84
2	Angka 1	50	43	7	86
3	Angka 2	50	46	4	92
4	Angka 3	50	42	8	84
5	Angka 4	50	40	10	80
6	Angka 5	50	44	6	88
7	Angka 6	50	45	5	90
8	Angka 7	50	47	3	94
9	Angka 8	50	48	2	96
10	Angka 9	50	47	3	94
Total			445	55	89

Pada hasil percobaan tersebut, digambarkan dalam bentuk grafik hasil percobaan pengenalan tulisan tangan angka dapat dilihat pada Gambar 12 dibawah ini.



Gambar 12. Grafik Hasil Percobaan Pengenalan Tulisan Tangan Angka.

Pada gambar 12, grafik hasil percobaan tulisan tangan angka menggunakan 2 data yaitu primer dan sekunder. Data sekunder yang pertama mengalami peningkatan akurasi daripada data lainnya. Hal ini disebabkan karena jumlah data sekunder pertama untuk data latih lebih banyak daripada jumlah data latih yang lainnya. Jumlah data latih pada data sekunder pertama berjumlah 1000 data. Sedangkan data primer berjumlah 450 data latih, data sekunder kedua berjumlah 450 data latih dan data sekunder ketiga berjumlah 500 data latih.

3.5 Analisis Hasil Akurasi Data Primer dan Data Sekunder

Berdasarkan hasil percobaan perangkat lunak untuk keseluruhan data set pengujian, data primer menghasilkan tingkat akurasi sebesar 88% untuk 50 gambar dan data sekunder pertama menghasilkan tingkat akurasi sebesar 98,2% untuk 500 gambar. Namun untuk percobaan data sekunder kedua sebanyak 50 data uji, tingkat akurasi sebesar 90%. Percobaan data sekunder ketiga sebanyak 500 data uji, tingkat akurasi sebesar 89%. Bila dilihat dari nilai akurasi tersebut, maka data primer memiliki tingkat akurasi yang lebih rendah bila dibandingkan dengan kedua data sekunder dengan jumlah yang berbeda. Tingkat akurasi yang dihasilkan dari percobaan data set yang berbeda membuktikan bahwa karakter tulisan tangan memiliki tingkat variasi yang tinggi dan tidak konsisten. Hal ini disebabkan oleh ketebalan dan bentuk tulisan yang tidak konsisten di setiap orang serta kebiasaan yang mempengaruhi karakter tulisan tangan seseorang. Data primer merupakan data yang diambil langsung dan melalui proses pemindai serta masih memiliki banyak derau pada citra tulisan tangan angka. Sedangkan data sekunder sudah mengalami proses citra keabuan sehingga citra tulisan tangan bersih dari derau. Data sekunder juga telah diuji oleh berbagai peneliti sehingga data sekunder telah memiliki pengetahuan yang didapatkan dari proses pembelajaran data sekunder. Tentunya hal ini memungkinkan tingkat akurasi yang dihasilkan oleh data primer lebih rendah.

4. KESIMPULAN

Berdasarkan hasil percobaan perangkat lunak untuk keseluruhan data set pengujian, data primer menghasilkan tingkat akurasi sebesar 88% untuk 50 gambar dan data sekunder pertama menghasilkan tingkat akurasi sebesar 98,2% untuk 500 gambar. Namun untuk percobaan data sekunder kedua sebanyak 50 data uji, tingkat akurasi sebesar 90%. Percobaan data sekunder ketiga sebanyak 500 data uji, tingkat akurasi sebesar 89%. Bila dilihat dari nilai akurasi tersebut, maka data primer memiliki tingkat akurasi yang lebih rendah bila dibandingkan dengan kedua data sekunder dengan jumlah yang berbeda. Tingkat akurasi yang dihasilkan dari percobaan data set yang berbeda membuktikan bahwa karakter tulisan tangan memiliki tingkat variasi yang tinggi dan tidak konsisten. Hal ini disebabkan oleh ketebalan dan bentuk tulisan yang tidak konsisten di setiap orang serta kebiasaan yang mempengaruhi karakter tulisan tangan seseorang. Data primer merupakan data yang diambil langsung dan melalui proses pemindai serta masih memiliki banyak derau pada citra tulisan tangan angka. Sedangkan data sekunder sudah mengalami proses citra keabuan sehingga citra tulisan tangan bersih dari derau. Data sekunder juga telah diuji oleh berbagai peneliti sehingga data sekunder telah memiliki pengetahuan yang didapatkan dari proses pembelajaran data sekunder. Tentunya hal ini memungkinkan tingkat akurasi yang dihasilkan oleh data primer lebih rendah.

REFERENCES

- [1] K. Dutta, "Improving CNN-RNN hybrid networks for handwriting recognition," *Proc. Int. Conf. Front. Handwrit. Recognition, ICFHR*, vol. 2018, pp. 80–85, 2018, doi: 10.1109/ICFHR-2018.2018.00023.
- [2] C. Boufenar, "Investigation on deep learning for off-line handwritten Arabic character recognition," *Cogn. Syst. Res.*, vol. 50, pp. 180–195, 2018, doi: 10.1016/j.cogsys.2017.11.002.
- [3] A. Baldominos, "Evolutionary convolutional neural networks: An application to handwriting recognition," *Neurocomputing*, vol. 283, pp. 38–52, 2018, doi: 10.1016/j.neucom.2017.12.049.
- [4] S. Pengenalan, T. Tangan, A. Fadlil, P. Studi, T. Elektro, and U. A. Dahlan, "(Influence of Feature Extraction Complexity in Online Handwriting Recognition)," pp. 127–132.



- [5] S. V. M. Classifiers *et al.*, “Handwritten Assamese Numeral Recognizer Using.”
- [6] R. J. Ramteke, “Invariant Moments Based Feature Extraction for Handwritten Devanagari Vowels Recognition,” vol. 1, no. 18, pp. 1–5, 2010.
- [7] I. J. Kim and X. Xie, “Handwritten Hangul recognition using deep convolutional neural networks,” *Int. J. Doc. Anal. Recognit.*, vol. 18, no. 1, pp. 1–13, 2015, doi: 10.1007/s10032-014-0229-4.
- [8] G. F. Fitriana, “Handwriting digit recognition using united moment invariant feature extraction and self organizing maps,” 2014, doi: 10.1109/ICT-ISPC.2014.6923214.
- [9] P. Kumar, “A lexicon-free approach for 3D handwriting recognition using classifier combination,” *Pattern Recognit. Lett.*, vol. 103, pp. 1–7, 2018, doi: 10.1016/j.patrec.2017.12.014.
- [10] K. Xu, W. Long, Y. Sun, and Y. Lin, “A novel image feature extraction algorithm based on the fusion AutoEncoder and CNN,” *J. Phys. Conf. Ser.*, vol. 1646, no. 1, 2020, doi: 10.1088/1742-6596/1646/1/012039.
- [11] “United Moment Invariants for Shape Discrimination,” vol. 00, no. October, pp. 88–93, 2003.
- [12] S. F. Rashid, F. Shafait, and T. M. Breuel, “Scanning Neural Network for Text Line Recognition.”
- [13] O. P. Sharma, M. K. Ghose, K. B. Shah, and B. K. Thakur, “Recent Trends and Tools for Feature Extraction in OCR Technology,” no. 6, pp. 220–223, 2013.
- [14] N. K. Raman, S. Gandhi, and J. Khurana, “Study and Analysis of Devnagari Handwritten Character Recognition Techniques,” vol. 2, no. 6, pp. 2–6, 2013.
- [15] S. ur Rehman, S. Tu, O. ur Rehman, Y. Huang, C. M. S. Magurawalage, and C. C. Chang, “Optimization of CNN through novel training strategy for visual classification problems,” *Entropy*, vol. 20, no. 4, pp. 1–11, 2018, doi: 10.3390/e20040290.
- [16] K. Jia, X. Wang, and X. Tang, “Image transformation based on learning dictionaries across image spaces,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 367–380, 2013, doi: 10.1109/TPAMI.2012.95.
- [17] R. Ghosh, “RNN based online handwritten word recognition in Devanagari and Bengali scripts using horizontal zoning,” *Pattern Recognit.*, vol. 92, pp. 203–218, 2019, doi: 10.1016/j.patcog.2019.03.030.