

Implementasi Metode *Backpropagation* untuk Mengenali Teks pada *Natural Scene Image*

Imam Ghozali¹, Tibyani², Putra Pandu Adikara³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹imam.ghz17@gmail.com, ²tibyani@ub.ac.id, ³adikara.putra@ub.ac.id

Abstrak

Teks pada gambar dapat memuat berbagai informasi penting bagi manusia, mendeteksi teks pada natural scene image (NSI) menjadi suatu bagian penting pada penelitian mengenai text information extraction (TIE), dalam mendapatkan informasi pada NSI terdapat berbagai macam tantangan yang harus dihadapi, seperti menentukan region teks pada background yang kompleks hingga mengenali berbagai macam teks yang memiliki beraneka ragam bentuk, ukuran, warna dan lain sebagainya. Hasil dari TIE dapat diimplementasikan pada berbagai hal salah satunya, sistem yang penulis kembangkan diharapkan nantinya dapat membantu orang tunanetra mengenali label produk atau mengenali tulisan yang ada disekitar mereka. Untuk mengenali teks pada NSI merupakan gabungan dari 2 topik penelitian, topik yang pertama adalah penentuan region teks atau text detection dan topik kedua adalah mengenali karakter tulisan atau character recognition. Pada penelitian ini hanya akan di fokuskan pada pengenalan karakter teks, metode yang digunakan adalah backpropagation dari hasil pengujian yang telah dilakukan hasil yang didapatkan yaitu sebesar 44.00%, hal ini dipengaruhi oleh fitur yang diekstraksi dari hasil proses binerisasi, banyak citra huruf yang digunakan sebagai data latih memiliki noise yang menyebabkan rendahnya tingkat akurasi yang diperoleh.

Kata kunci: *Backpropagation, Natural Scene Image, Optical Character Recognition.*

Abstract

Text in images contain useful information. Text detection in Natural Scene Image (NSI) becomes an important part of research on text information extraction (TIE). For getting information on the NSI there are various challenges, such as determining the text region in a complex background to recognize various kinds of text that has a variety of shapes, sizes, colors and others. The results of TIE can be implemented in many ways, one of them system can to help blind people to recognize product label or recognize text is around them. To recognize text on the NSI is a composite of two research topics, the first topic is determination of the text region or text detection and second topic is recognizing character of text or character recognition. In this research will focus on recognizing character using backpropagation method. Results of tests is 44.00%, this is influenced by the features extracted from the results of image binerization, many image data has a noise.

Keywords: *Backpropagation, Natural Scene Image, Optical Character Recognition.*

1. PENDAHULUAN

Teks pada gambar dapat memuat berbagai informasi penting bagi manusia. Mendeteksi teks pada *natural scene image* (NSI) menjadi suatu bagian penting pada penelitian mengenai *text information extraction* (TIE). Dalam mendapatkan informasi pada NSI terdapat berbagai macam tantangan yang harus dihadapi, seperti menentukan *region* teks pada *background* yang kompleks hingga mengenali

berbagai macam teks yang memiliki beraneka ragam bentuk, ukuran, warna dan lain sebagainya. Hasil dari TIE dapat diimplementasikan pada berbagai hal salah satunya, sistem yang penulis kembangkan diharapkan nantinya dapat membantu orang tunanetra mengenali label produk atau mengenali tulisan yang ada disekitar mereka, untuk mengenali teks pada NSI merupakan gabungan dari 2 topik penelitian, topik yang pertama adalah penentuan *region* teks atau *text*

detection dan topik kedua adalah mengenali karakter tulisan atau *character recognition*.

Pada penelitian sebelumnya dilakukan penelitian dengan membandingkan antara metode metode LVQ dan *backpropagation* pada pengenalan huruf hijaiyah, dan hasilnya LVQ mencapai akurasi 58%, dan *Backpropagation* mencapai akurasi 98% (Mailani, et al., 2016), untuk *text recognition* pada *natural scene image* (NSI) telah dilakukan penelitian oleh (Ali & Foroosh, 2016) menggunakan metode *Rank-1 Tensor Decomposition* yang menghasilkan akurasi tertinggi mencapai 70%.

Berdasarkan pada penelitian sebelumnya *backpropagation* dapat mengenali karakter dengan tingkat akurasi yang tinggi namun belum dilakukan pada NSI yang memiliki variasi pada jenis teks, warna, ukuran dan lain sebagainya, sehingga kali ini peneliti memfokuskan diri mengimplementasikan metode *backpropagation* untuk mengenali teks pada NSI.

2. LANDASAN KEPUSTAKAAN

2.1 Citra

Citra adalah gambar bidang dua dimensi, atau dapat dikatakan merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi. Sumber cahaya menerangi objek, objek memantulkan kembali berkas cahaya tersebut, pantulan dari berkas cahaya tersebut ditangkap oleh mata, kamera dan lainnya sehingga kita dapat melihat (Munir, 2004).

2.2 Grayscale

Grayscale adalah warna piksel yang memiliki rentang keabuan atau berada pada rentang hitam dan putih, *grayscale* terdiri dari kombinasi 29,9% piksel merah, 58,7% piksel hijau, 11,4% piksel biru (Abdullah-al-mamun & Alam, 2015), mengubah warna menjadi *grayscale* dapat mempermudah pengolahan citra, hal ini dikarenakan tidak banyak warna yang di proses, untuk mengubah warna menjadi *grayscale* dapat menggunakan Persamaan 1:

$$Gray = (R \times 0.299 + G \times 0.587 + B \times 0.144) \quad (1)$$

2.3 Image Binarization

Image binarization atau binerisasi citra berfungsi untuk mengubah citra *grayscale* menjadi citra biner dengan tingkat kontras yang tinggi. Hal ini dikarenakan semua piksel yang lebih dari *threshold* akan diubah menjadi putih,

dan sisanya yang kurang dari *threshold* akan diubah menjadi hitam (Munir, 2004).

2.4 Segmentasi

Segmentasi berfungsi membagi citra ke dalam beberapa bagian. Proses ini digunakan untuk memisahkan antar karakter dalam suatu teks atau kalimat. Penelitian kali ini akan dikombinasikan 2 buah algoritme segmentasi untuk mendapatkan hasil yang maksimal yaitu *vertical projection* dan *drop fall algorithm*.

2.4.1 Vertical Projection

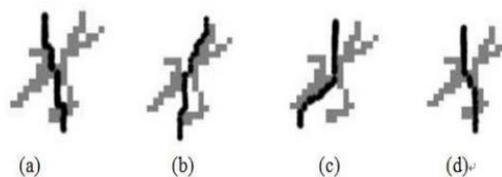
Vertical projection adalah metode yang menghitung jumlah piksel pada setiap kolom pada gambar, setelah citra diubah menjadi citra biner *Vertical projection* dapat digunakan untuk menghitung piksel hitam maupun piksel putih (Shanjana & James, 2014).

Algoritme:

1. Tentukan ukuran dari gambar [M N]
2. Hitung setiap jumlah nilai piksel perkolom dari 1 sampai N

2.4.2 Drop Fall Algorithm

Algoritme *Drop fall* dapat membuat *path* untuk segmentasi dari atas hingga bawah dari karakter, terdapat 3 komponen dalam *Drop fall* yaitu *start point*, *movement rules*, dan *direction* (Yang J. et al, 2016). *Direction* sendiri terdiri dari 4 jenis. *Descending-left*, *descending-right*, *ascending-left*, *ascending-right*, seperti yang terlihat pada Gambar 1.



Gambar 1 Jenis *Direction* pada *Drop Fall*
Sumber: Yang J. et al (2016)

Sedangkan pada *Drop fall* terdapat 6 *rule* yaitu:

1. If $n3 = 1$, then $next_pixel = n3$
2. If $n4 = 1$, then $next_pixel = n4$
3. If $n2 = 1$, then $next_pixel = n2$
4. If $n5 = 1$, then $next_pixel = n5$
5. If $n1 = 1$, then $next_pixel = n1$
6. Else $next_pixel = n3$

Nilai 1 menunjukkan piksel putih, sedangkan $n1$,

n_2, n_3, n_4, n_5 menunjukkan piksel.

2.5 Jaringan Saraf Tiruan *Backpropagation*

Jaringan saraf tiruan (JST) atau dapat disebut sebagai *artificial neural network* merupakan salah satu cabang ilmu dari kecerdasan buatan yang berdasar dari sel saraf pada makhluk hidup. JST dapat digunakan untuk belajar dan menghasilkan aturan atau operasi, contohnya adalah saat kita menginginkan *output* yang sempurna dari data yang dimasukkan, dengan JST kita dapat membuat prediksi mengenai *output* yang akan muncul atau mengenali karakteristik *input* (Kristanto, 2004). *Backpropagation* adalah salah satu dari algoritme *supervised learning*. *Backpropagation* memiliki 3 tahap yaitu perhitungan maju (*forward*), perhitungan mundur (*backward*) dan perubahan nilai bobot dan nilai bias. Secara umum berikut ini adalah algoritme dari *backpropagation*.

1. Inisialisasi semua bobot
2. Jika kondisi berhenti belum terpenuhi, lakukan langkah 3-8
3. Untuk setiap data pelatihan, lakukan langkah 4-7

Fase I: Propagasi Maju

4. Hitung keluaran unit di tersembunyi Z_j ($j=1,2,\dots,p$)

$$z_net_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \tag{2}$$

Keterangan:

- Z_net_j = *Input* jaringan ke Z_j
- V_{0j} = Bias pada *layer* j
- X_i = Unit *input* i
- V_{ij} = Nilai bobot dari unit X_i ke unit Z_i

Kemudian gunakan fungsi aktivasi untuk menghitung keluaran dari *input*.

$$z_j = f(z_net_j) = \frac{1}{1 + e^{-z_net_j}} \tag{3}$$

Keterangan:

Z_j = Unit ke- j pada *hidden layer*
Setelah itu kirimkan hasil tersebut ke unit-unit di *hidden layer*.

5. Hitung keluaran jaringan pada unit y_k ($k=1,2,\dots,m$)

$$y_net_k = w_{k0} + \sum_{i=1}^p z_i w_{ki} \tag{4}$$

Keterangan:

- Y_net_j = *Input* jaringan ke Y_j
 - W_{0j} = Bias pada *layer* j
 - W_{ij} = Nilai bobot dari unit X_i ke unit Z_i
- Kemudian gunakan fungsi aktivasi untuk menghitung keluaran.

$$y_k = f(y_net_k) = \frac{1}{1 + e^{-y_net_k}} \tag{5}$$

Keterangan:

Y_k = Unit ke- k pada *output layer*.

Fase II: Propagasi Balik

6. Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran y_k ($k = 1,2,\dots,m$)

$$\begin{aligned} \delta_k &= (t_k - y_k) f'(y_net_k) \\ &= (t_k - y_k) y_k (1 - y_k) \end{aligned} \tag{6}$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot. Hitung perubahan w_{kj} dengan laju percepatan α .

$$\Delta w_{kj} = \alpha \delta_k z_j \quad k=1, 2, m \quad j=1, 2, p \tag{7}$$

7. Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j ($j=1,2,\dots,p$)

$$\delta_net_j = \sum_{k=1}^m \delta_k w_{kj} \tag{8}$$

Faktor δ unit tersembunyi:

$$\begin{aligned} \delta_k &= \delta_net_j f'(z_net_j) \\ &= \delta_net_j z_j (1 - z_j) \end{aligned} \tag{9}$$

Hitung perubahan v_{kj}

$$\Delta v_{ij} = \alpha \delta_j z_j \quad j=1, 2, p \quad i=1, 2, n \tag{10}$$

Keterangan:

α = *Learning rate*

Fase III: Perubahan Bobot

8. Hitung semua perubahan bobot

$$\begin{aligned} w_{jk}(\text{baru}) &= \\ w_{kj}(\text{lama}) &+ \Delta w_{kj} \end{aligned} \tag{11}$$

$$v_{ij}(baru) = v_{ij}(lama) + \Delta v_{ij_{kj}} \tag{12}$$

9. Memeriksa stopping condition.

Ada 2 buah parameter yang dapat digunakan untuk memeriksa *stopping condition*, yang pertama adalah maksimum *epoch* dan yang kedua adalah *minimum error*. Jika salah satu dari *stopping condition* sudah terpenuhi maka proses akan dihentikan.

2.6 Euclidean Distance

Euclidean distance adalah salah satu metode dalam menentukan jarak dari 2 buah titik. Dari hasil penelitian yang dilakukan oleh (Greche, et al., 2017) dengan menggunakan *euclidean distance* dapat mendapatkan akurasi yang tinggi dengan menggunakan *hidden unit* yang lebih sedikit dari *manhattan distance*. Persamaan *euclidean distance* ditunjukkan pada Persamaan (13).

$$d_e(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \tag{13}$$

Keterangan:

- $d_e(p, q)$ = Merupakan jarak antara titik p dan q .
- p_i = Titik pertama
- q_i = Titik kedua

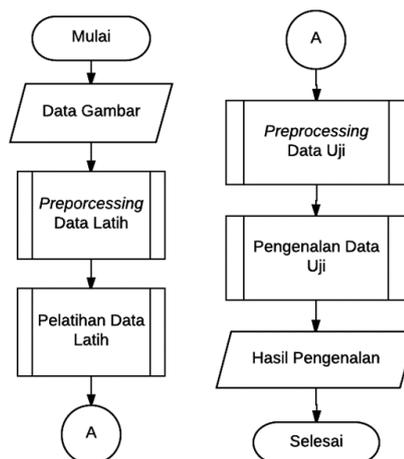
3. METODOLOGI

3.1. Pengumpulan Data

Pengumpulan data bertujuan untuk mendapatkan data yang akan digunakan pada penelitian, teknik pengumpulan data yang digunakan adalah sekunder. Data yang digunakan untuk pelatihan dan pengujian didapatkan dari dataset *International Conference Document Analysis and Recognition (ICDAR) 2003*.

3.2. Perancangan Sistem

Perancangan sistem adalah tahapan yang dilakukan untuk menjabarkan mengenai kinerja sistem secara terstruktur, dimulai dari *input* hingga *output* yang dihasilkan, *flowchart* kerja sistem ini dapat dilihat pada Gambar 2.



Gambar 2 Flowchart Sistem

Pada Gambar 2 dapat dilihat bahwa perancangan sistem terdiri dari tiga proses utama, yaitu:

1. Input

Input adalah masukan dari pengguna berupa gambar dari *Natural Scene image*.

2. Proses

Proses yang dilakukan pertama adalah *preprocessing*, kemudian dilakukan segmentasi untuk memisahkan satu karakter dengan yang lain, lalu dilakukan *training* karakter menggunakan *backpropagation*, dan terakhir dilakukan *recognition*.

3. Output

Output adalah keluaran dari sistem yang diharapkan yaitu berupa teks dari gambar

4. PENGUJIAN

4.1 Pengujian Parameter

Pada tahap pengujian ini akan dicari komposisi yang paling baik dari arsitektur *backpropagation* untuk mengenali karakter. Pengujian parameter akan terdiri dari 2 parameter yang diuji, yaitu jumlah *hidden unit* dan *learning rate*, akan dicari kombinasi dari keduanya yang menghasilkan akurasi terbaik. Pengujian akan dilakukan dengan cara melakukan pengenalan terhadap 26 karakter huruf.

4.1.1 Pengujian Jumlah Hidden Unit

Pengujian akan dilakukan untuk mengetahui pengaruh jumlah *hidden unit* dan terhadap tingkat akurasi. Pada pengujian ini

jumlah *input unit* yang digunakan adalah 96, Jumlah maksimum *epoch* adalah 7000, Jumlah *hidden unit* yang akan diuji adalah 10, 20, 30, 40, 60, 70, 80, 90, dan *learning rate* yang akan digunakan adalah 0.1. Pengujian akan dilakukan sebanyak 3 kali kemudian akan diambil rata-rata akurasi, dan hasil terbaik akan digunakan untuk pengujian akurasi. Hasil pengujian *hidden unit* dapat dilihat pada Tabel 1, 2, 3, dan Rata-rata hasil pengujian dapat dilihat pada Tabel 4.

Tabel 1 Pengujian Pertama

No	Hidden Unit	Dikenali	Akurasi
1	10	2	7.69%
2	20	1	3.85%
3	30	1	3.85%
4	40	2	7.69%
5	50	8	30.77%
6	60	4	15.38%
7	70	4	15.38%
8	80	4	15.38%
9	90	8	30.77%

Tabel 2 Pengujian Kedua

No	Hidden Unit	Dikenali	Akurasi
1	10	2	7.69%
2	20	1	3.85%
3	30	1	3.85%
4	40	2	7.69%
5	50	8	30.77%
6	60	4	15.38%
7	70	4	15.38%
8	80	4	15.38%
9	90	8	30.77%

Tabel 3 Pengujian Ketiga

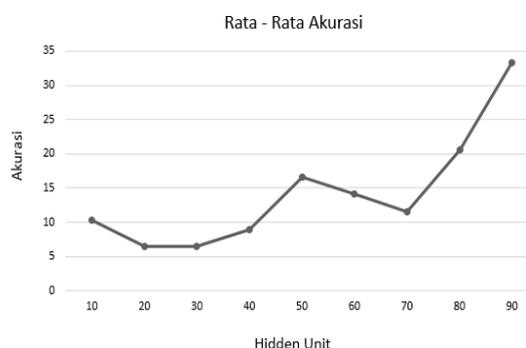
No	Hidden Unit	Dikenali	Akurasi
1	10	4	15.38%
2	20	2	7.69%
3	30	3	11.54%
4	40	4	15.38%
5	50	2	7.69%
6	60	4	15.38%
7	70	2	7.69%
8	80	5	19.23%
9	90	11	42.31%

Tabel 0 Rata-rata Hasil Pengujian

No	Hidden Unit	Akurasi
1	10	10.26%

2	20	6.41%
3	30	6.41%
4	40	8.97%
5	50	16.67%
6	60	14.10%
7	70	11.54%
8	80	20.51%
9	90	34.62%

Dari pengujian yang telah dilakukan menunjukkan bahwa jumlah *hidden unit* berpengaruh terhadap tingkat akurasi, namun tidak ada acuan pasti berapa jumlah *hidden unit* yang harus digunakan untuk mendapatkan hasil yang optimal, dikarenakan setiap BPNN memiliki jumlah *input* dan *output* masing-masing, sehingga jumlah *hidden unit* akan menyesuaikan dari arsitektur *input* dan *output* BPNN yang digunakan. Dari pengujian yang telah dilakukan menunjukkan bahwa semakin banyak *hidden unit* tidak akan menjamin menghasilkan hasil yang lebih baik dari pada *hidden unit* dengan jumlah yang sedikit, Grafik rata-rata hasil pengujian dapat dilihat pada Gambar 3.



Gambar 3 Grafik Rata-rata Hasil Pengujian

4.1.2 Pengujian Learning Rate

Pengujian *learning rate* digunakan untuk mengetahui pengaruh *learning rate* terhadap tingkat akurasi. Pada pengujian ini jumlah *input unit* yang digunakan adalah 96, Jumlah maksimum *epoch* adalah 7000, Jumlah *hidden unit* yang akan digunakan adalah 90 dan *learning rate* yang akan diuji adalah 0.1, 0.2, 0.3, 0.4, 0.5. Hasil pengujian *learning rate* akan ditunjukkan pada Tabel 5.

Tabel 5 Pengujian Learning Rate

Learning Rate	Percobaan			Rata-rata
	1	2	3	
0.1	30.77%	30.77%	42.31%	34.62%

0.2	15.38%	7.69%	11.54%	11.54%
0.3	30.77%	30.77%	7.69%	23.08%
0.4	23.08%	23.08%	11.54%	19.23%
0.5	30.77%	15.38%	34.62%	26.92%

Menurut hasil pengujian pada Tabel 5 menunjukkan bahwa *learning rate* berpengaruh terhadap tingkat akurasi, dari lima *learning rate* yang menghasilkan hasil terbaik adalah 0.1, dengan tingkat rata-rata akurasi sebesar 34.62%. Semakin kecil *learning rate* akan semakin lama kecepatan belajar hal ini dikarenakan perubahan bobot yang terjadi akan semakin kecil, sehingga akan membutuhkan waktu dan *epoch* yang lebih banyak untuk mencapai nilai minimum *error*.

4.2 Pengujian Akurasi

Pengujian akurasi dilakukan untuk mengetahui tingkat akurasi BPNN dalam mengenali teks pada NSI, dengan menggunakan hasil pengujian sebelumnya yaitu *learning rate* yang akan digunakan adalah 0.1 dan *hidden unit* yang digunakan adalah 90. Data untuk pengujian adalah 14 teks yang dipilih secara acak, dengan total terdiri dari 61 huruf. Hasil pengujian akurasi akan ditunjukkan oleh Tabel 6.

Tabel 6 Hasil Pengujian Akurasi

Data	Hasil	Dikenali	Akurasi
	YOC	2	66.67%
	POIPAPI	3	42.86%
	PAEI	1	25.00%
	FOW	2	66.67%
	CARCCP	3	50.00%
	EAIP	1	25.00%
	ECCAE	2	40.00%
	IEACF	2	40.00%
	BIEPIIIY	1	33.33%
	OICEI	2	40.00%
	PIPAPFPI PRT	3	30.00%
	EEIATE	4	66.67%

	FEPE	2	50.00%
	FIIOQFI	0	0%
Rata-rata			44.00%

Berdasarkan hasil akurasi rata-rata yang didapatkan adalah 44.00%, hasil yang didapatkan cukup kecil, hal ini dipengaruhi oleh bentuk huruf dan posisi yang berbeda beda untuk setiap karakter yang dikenali pada teks, sehingga mempersulit proses pengenalan, selain hal tersebut dikarenakan *background* yang beraneka ragam dan terdapat pula *noise* seperti cahaya yang tidak merata, atau juga terdapat bayangan, yang menyebabkan tidak semua proses *preprocessing* menghasilkan binerisasi dan ekstraksi fitur yang sempurna, contoh hasil binerisasi data latih ditunjukkan pada Tabel 7.

Tabel 7 Hasil Binerisasi

Data	Hasil Binerisasi

5. KESIMPULAN

Setelah melakukan penelitian mengenai pengenalan teks pada NSI menggunakan *Backpropagation*, terdapat beberapa kesimpulan yang dapat diambil antara lain.

1. *Backpropagation* dapat digunakan untuk mengenali huruf atau karakter pada NSI, dengan melalui beberapa tahap *preprocessing* antara lain *image binarization*, *crop*, *resize*, dan segmentasi pada citra. *image binarization* berfungsi untuk merubah citra menjadi biner sehingga memunculkan warna hitam dan putih saja, kemudian *crop* berfungsi untuk memotong *background* citra huruf sehingga hanya tersisa hurufnya saja, *resize* berfungsi untuk mengubah ukuran citra menjadi 8 x 12 piksel, piksel tersebut akan

digunakan sebagai fitur yang dimasukkan pada *neuron input*, dan yang terakhir segmentasi, segmentasi hanya digunakan pada proses pengenalan, berfungsi untuk memecah teks menjadi huruf.

2. Akurasi yang didapatkan cukup rendah yakni sebesar 44.00%, hal ini disebabkan karena pada saat pelatihan dan pengenalan, terdapat beberapa faktor yang mempengaruhi data yang digunakan antara lain bentuk huruf yang berbeda-beda, cahaya yang tidak merata, terdapatnya bayangan pada gambar, dan posisi huruf yang berbeda-beda menyebabkan hasil yang tidak maksimal. Jumlah *hidden unit* dan *learning rate* juga berpengaruh terhadap tingkat akurasi dari BPNN, pada penelitian kali ini didapatkan hasil terbaik dengan menggunakan 90 *hidden unit*, dan 0.1 *learning rate*.

dengan Pendekatan Algoritmik. Bandung: Informatika Bandung.

- C, S. & James, A., 2014. *Character Segmentation in Malayalam Handwritten Documents*. Unnao, International Conference on Advances in Engineering & Technology Research (ICAETR - 2014).

Yang, J., Guo, J. & Jiang, W., 2016. *A Novel Drop-fall Algorithm Based on Digital Features for Touching Digit Segmentation*, Shanghai: East China Normal University.

DAFTAR PUSTAKA

- Abdullah-al-mamun, M. & Alam, T., 2015. *An approach to empirical Optical Character recognition paradigm using Multi-Layer Perceptorn Neural Network*. Dhaka, 2015 18th International Conference on Computer and Information Technology (ICCIT).
- Ali, M. & Foroosh, H., 2016. *CHARACTER RECOGNITION IN NATURAL SCENE IMAGES USING RANK-1 TENSOR*. Phoenix, 2016 IEEE International Conference on Image Processing (ICIP).
- Desai, S. & Singh, A., 2016. *Optical character recognition using template matching*. Coimbatore, International Conference on Inventive Computation Technologies (ICICT) IEEE.
- Greche, L. et al., 2017. *Comparison Between Euclidean and Manhattan Distance Measure for Facial Expressions Classification*. Fez, 2017 International Conference on Wireless Technologies,
- Mailani, N. A., Amrizal, V. & Hakiem, N., 2016. *Comparative Analysis of the Accuracy of Backpropagation and Learning Vector Quantisation for Pattern Recognition of Hijaiyah Letters*. s.l., 6th International Conference on Information and Communication Technology for The Muslim World.
- Munir, R., 2004. *Pengolahan Citra Digital*