

DETEKSI KARAKTER HURUF ARAB DENGAN MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK

Ibnu Akil¹; Indra Chaidir²

Sistem Informasi
Universitas Bina Sarana Informatika
www.bsi.ac.id
ibnu.ial@bsi.ac.id¹, indra@bsi.ac.id²

Abstract— *In an all-digital world that doesn't mean there are no handwriting anymore. For example, reading a check at a bank still has to accept handwritten input. The problem is that many OCR applications cannot facilitate all languages, one of which is Arabic. Therefore we need an application that can identify Arabic handwritten hijaiyah letters. The purpose of this study is to develop an artificial intelligent application to detect Arabic characters using the Convolutional Neural Network method. The results of this study can be used as a basis for further development of OCR applications in many languages.*

Keywords: *Arabic Alphabet Recognition; Convolutional Neural Network; Detection*

Abstrak— Dalam dunia yang serba digital bukan berarti tidak ada lagi tulisan tangan. Contohnya seperti membaca cek di bank masih harus menerima input berupa tulisan tangan. Masalahnya banyak aplikasi OCR belum bisa memfasilitasi semua bahasa salah satunya adalah bahasa arab. Karenanya diperlukan aplikasi yang dapat mengidentifikasi huruf hijaiyah tulisan tangan bahasa arab. Tujuan dari penelitian ini adalah mengembangkan aplikasi artificial intelligent untuk mendeteksi karakter huruf arab dengan metode Convolutional Neural Network. Hasil penelitian ini dapat dimanfaatkan sebagai dasar pengembangan lebih lanjut aplikasi OCR dengan banyak bahasa.

Kata kunci: Pengenalan Huruf Arab; Convolutional Neural Network; Deteksi

PENDAHULUAN

Sejak masuknya era globalisasi maka batasan antara satu bangsa dengan bangsa lain menjadi sangat tipis, namun tetap saja ada gap yang tidak bisa kita pungkiri cukup besar pengaruhnya untuk melengkapi ide “globalisasi” yaitu bahasa. Meskipun penyebaran bahasa Inggris yang telah berhasil memfasilitasi keseragaman dunia, namun membahayakan keberadaan dari banyak bahasa dan kebudayaan (Irina, 2011).

Meskipun interface untuk input dan output sudah memfasilitasi berbagai bahasa yang tidak menggunakan English alphabet seperti bahasa China, Jepang, dan Arab, namun input berupa touchscreen keyboard tidak optimal untuk bahasa tersebut karena perbedaan layout (Bi, Smith, & Zhai, 2015), dan dalam beberapa hal masih diperlukan input dari tulisan tangan secara langsung yang biasanya menggunakan optical character reader (OCR), digital pen, dan tablet. Sebagai contoh adalah aplikasi untuk memproses cek secara otomatis yang digunakan di bank (Aqab & Tariq, 2020). Selain itu belum adanya standar tulisan yang meliputi multi-font, multi-style dan multi-size tulisan bahasa arab yang tercetak yang bisa digunakan untuk

mengevaluasi nilai dari sistem OCR berbahasa arab (Jaiem, Kanoun, Khemakhem, El Abed, & Kardoun, 2013).

Berdasarkan beberapa literature yang dipelajari terkait artificial intelligent untuk image processing dan image identification untuk character reader telah ada beberapa metode diantaranya:

Gradient Based Learning yang diajukan oleh Yann LeCun dan kawan-kawan. Gradient Based Learning menyediakan kerangka kerja untuk melakukan segmenter dan feature extractor dari suatu image menjadi raw data (pixel image). Metode ini secara sederhana adalah ekstensi dari back propagation dari neural network, dimana fungsi objektif secara efisien bisa diminimalisasi dengan menggunakan gradient descent. Convolutional network yang diajukan LeCun mengadopsi dari model LeNet-5 (LeCun, Haffner, Bottou, & Bengio, 1998).

Model AlexNet yang diajukan oleh Krizhevsky dan memenangkan image recognition contest termasuk model yang banyak dipergunakan. Model ini menurunkan overfitting pada fully connected layer dengan teknik “dropout” yang terbukti cukup efektif (Krizhevsky, Sutskever, & Hinton, 2017).

Penelitian yang dilakukan oleh Muhammad Aminur Rahaman mengadopsi model CNN yang dikembangkan oleh Alom dengan mengurangi jumlah convolutional layer dari empat menjadi dua, kemudian menambahkan dropout filter pada tiga layer dan mengganti Global Average Pooling (GAP) dengan dua fully connected layer. Jumlah class yang digunakan oleh Rahaman ada 60 class (Rahaman, Mahin, Ali, & Hasanuzzaman, 2019).

Sementara Sara Aqob dalam penelitiannya yang berjudul "Handwriting Recognition using Artificial Intelligence Neural Network and Image Processing", masih menggunakan model CNN dengan konfigurasi dua convolutional layer, dua subsampling atau max-pooling dan dua fully connected layer untuk kasus handwritten OCR (Aqab & Tariq, 2020).

Dari hasil Analisa terhadap penelitian sebelumnya Gradient Based Learning adalah model dasar CNN sementara AlexNet sudah dapat memproses high-resolution image dengan memanfaatkan GPU (Graphic Processing Unit) yang canggih. Sedangkan model CNN yang akan dikembangkan dalam penelitian ini mengadopsi dari model yang dikembangkan oleh Rahaman dan dapat diimplementasikan dengan teknologi CPU yang tidak terlalu tinggi untuk image resolusi rendah.

Tujuan utama dari paper ini adalah mengembangkan system yang dapat mengenali input berupa karakter huruf arab, dan menganalisa tingkat akurasi model adopsi *convolutional neural network (CNN)* dalam mengenali image.

BAHAN DAN METODE

Penelitian ini dirancang menjadi beberapa tahapan. Mulai dari studi literature mengenai image processing, character reader, artificial intelligent dan neural network. Sumber data berupa image Arabic Handwritten Characters Dataset (AHCD), didapat dari <https://www.kaggle.com/mloey1/ahcd1> yang merupakan hasil riset dari Ahmed El Sawy dan Mohammed Loey (El Sawy, Loey, & El Bakry, 2019).

Model CNN

Standar model CNN pertamakali dikembangkan oleh LeCun dan kawan-kawan dalam papernya yang berjudul Object Recognition with Gradient-Based Learning (LeCun, Haffner, Bottou, & Bengio, 1999), namun sebelum itu CNN telah ada dengan nama yang berbeda dikenalkan oleh Kunihiko Fukushima pada tahun 1980.

CNN terdiri dari multilayer artificial neurons, dimana artificial neurons merupakan fungsi matematika yang menghitung total bobot dari banyak masukkan dan mengeluarkan nilai aktivasi

(Dickson, 2020). Layer-layer tersebut berupa convolution, subsampling, dense atau fully connected dan output layer.

Untuk layer convolution menggunakan fungsi aktivasi ReLU (Rectified Linear Unit) adalah sebuah fungsi linear yang akan mengeluarkan nilai output langsung apabila input bernilai positif, selain itu maka output berupa 0 (Brownlee, 2019). Berikut adalah fungsi ReLU:

$$f(x) = x^+ = \max(0, x) \dots \dots \dots (1)$$

Setelah convolution layer biasanya ada subsampling layer. Neuron di lapisan ini menerima masukan dari bidang reseptif non-overlapping kecil dari layer sebelumnya. Setiap neuron menghitung jumlah inputnya, mengalikannya dengan koefisien, menambahkan bias yang dapat dilatih, dan meneruskan hasilnya melalui fungsi transfer non-linear (Scherer, Andreas, & Behnke, 2010).

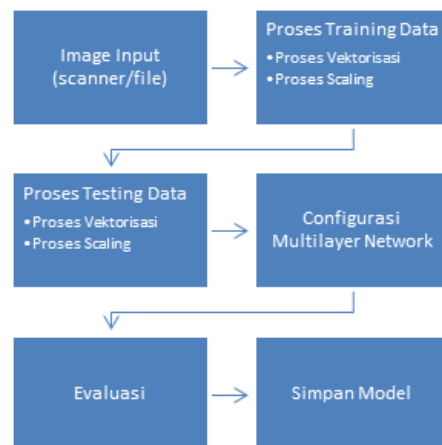
Sedangkan untuk output layer menggunakan fungsi aktivasi softmax.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \dots \dots \dots (2)$$

Library Deeplearning4j

Deeplearning4j adalah library deep-learning yang ditulis dengan bahasa Java untuk tingkat komersial pertama yang bersifat open source. Deeplearning4j merupakan framework dengan banyak dukungan algoritma deep-learning (Novet, 2014). Seluruh lisensinya berada dibawah Apache licence.

Tugas utama dari system alphabet character recognition adalah adalah membaca input character dan mengidentifikasinya secara tepat sebagai salah satu class yang sudah ditentukan (Klassen, 2001). Rancangan sistem yang akan dikembangkan dengan library deeplearning4j menggunakan model CNN dapat dilihat di gambar 1.



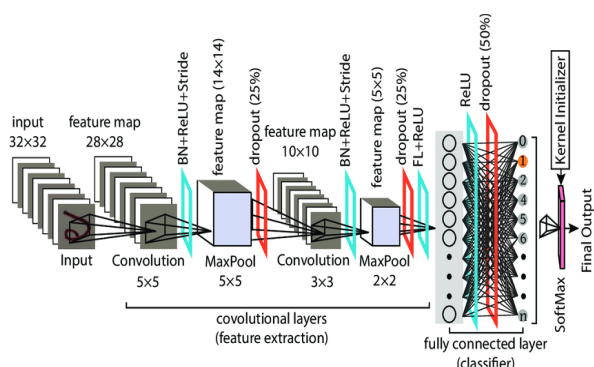
Sumber: (Akil & Chaidir, 2020)

Gambar 1. Rancangan Sistem yang dikembangkan

HASIL DAN PEMBAHASAN

1. Model CNN Pengembangan

Dalam rancangan riset ini mengadopsi model CNN yang dikembangkan oleh Muhammad Aminur Rahaman untuk membaca karakter/huruf Bangla karena memiliki resolusi input image yang sama berukuran 32x32 pixels (Rahaman et al., 2019).



Sumber: (Rahaman et al., 2019)

Gambar 2. Model CNN yang digunakan

Namun dari model pada gambar 2, ada beberapa modifikasi konfigurasi dari masing-masing layer untuk mendapatkan hasil training yang optimal.

2. Konfigurasi Layer

Model CNN ini berisi 6 layer dengan konfigurasi masing-masing layer perhatikan table 1 berikut:

Tabel 1. Konfigurasi CNN Layer

Layer	Deskripsi	Output	Konfigurasi
0	Input image	32x32	1 channel
1	Convolution Layer 1	28x28	32@5x5, Activation=ReLU, Stride=1
2	Max Pooling 1	14x14	32@3x3, stride=2
3	Convolution Layer 2	10x10	64@5x5, Activation=ReLU, stride=1
4	Max Pooling 2	5x5	64@2x2, stride=2
5	Dense Layer	1x1	Activation=ReLU
6	Output Layer	1x1	Activation=Softmax

Sumber: (Akil & Chaidir, 2020)

Layer 0 merupakan input yang berupa image berukuran 32x32 dengan format PNG 1 channel (hitam putih) atau 2D. Layer 1 adalah convolutional layer. Jumlah filter atau kernel adalah 5x5 pixel, dan stride (jumlah pixel yang digeser/slide) (Prabhu, 2018) 1 pixel. Sedangkan output untuk layer selanjutnya berjumlah 28 sesuai dengan jumlah class yang ada. Sebagai tambahan sebaiknya untuk

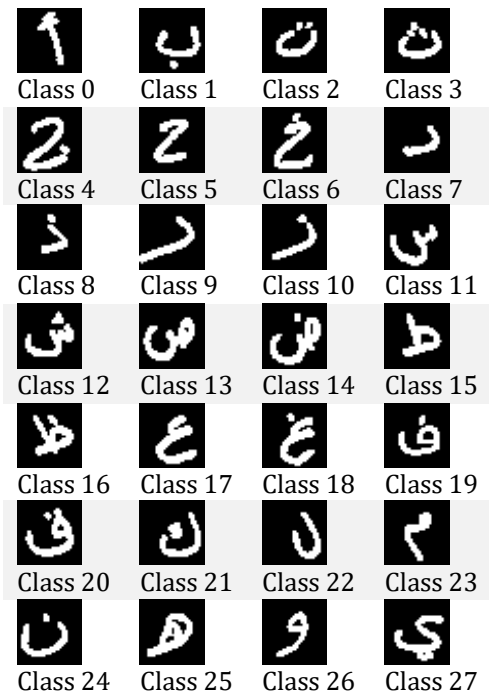
output layer berikutnya merupakan kelipatan dari class. Selanjutnya layer 2 merupakan subsampling layer yang berfungsi untuk downsampling dari output layer convolutional dengan resolusi 3x3 dan stride=2. Setelah subsampling layer yang pertama layer berikutnya adalah layer convolutional 2 dengan konfigurasi filter 5x5, stride 1 dan output untuk layer selanjutnya adalah jumlah class (28 x 10 = 280). Kemudian kembali menjadi subsampling layer pada layer ke 4 dengan konfigurasi kernel 2x2 dan stride=2. Pada posisi layer ke 5 ditempatkan dense layer. Dense layer adalah layer yang terkoneksi seluruhnya dari input layer sebelumnya (Sharma, 2020). Menggunakan fungsi aktivasi ReLU dan output jumlah class (28 x 28=784).

Kita dapat mengkonfigurasi layer dengan mudah menggunakan library deeplearning4j. Untuk coding digunakan Netbeans IDE dan Maven repository untuk projectnya.

```
MultiLayerConfiguration conf = new
NeuralNetConfiguration.Builder()
    .seed(seed)
    .l2(0.0005)
    .updater(new Nesterovs(new
MapSchedule(ScheduleType.ITERATION,
learningRateSchedule)))
    .weightInit(WeightInit.XAVIER)
    .list()
    .layer(new ConvolutionLayer.Builder(5,
5)//layer 1 Convolutional Layer
    .nIn(1)
    .stride(1, 1)
    .nOut(28)
    .activation(Activation.RELU)
    .build())
    .layer(new SubsamplingLayer.Builder(
SubsamplingLayer.PoolingType.MAX) //layer 2 Sub
Sampling Layer
    .kernelSize(3, 3)
    .stride(2, 2)
    .build())
    .layer(new ConvolutionLayer.Builder(5, 5)
//layer 3 Convolutional Layer
    .stride(1, 1)
    .nOut(280)//28*10
    .activation(Activation.RELU)
    .build())
    .layer(new SubsamplingLayer.Builder(
SubsamplingLayer.PoolingType.MAX)//layer 4 Sub
Sampling Layer
    .kernelSize(2, 2)
    .stride(2, 2)
    .build())
    .layer(new DenseLayer.Builder().activation(
Activation.RELU)//layer 5 Dense Layer
    .nOut(784)//28 * 28
    .build())
    .layer(new OutputLayer.Builder(
LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
//layer 6 Output Layer
    .nOut(outputNum)
    .activation(Activation.SOFTMAX)
    .build())
    .setInputType(InputType.convolutionalFlat(height
, width, channels)) // InputType.convolutional
for normal image
    .build();
```

3. Classification

Dalam penelitian ini ada 28 class yang merupakan huruf hijaiyah dari karakter bahasa arab, dapat dilihat pada gambar 3



Sumber: (Akil & Chaidir, 2020)

Gambar 3. Classification Image

Jumlah data training sebanyak 13.440 dengan ukuran pixel 32x32, dan data test sebanyak 3.360 dengan pixel yang sama.

4. Hasil Implementasi (Training)

Dari 3 epoch (jumlah run training) yang telah dilakukan bisa dilihat terdapat peningkatan akurasi dalam mengidentifikasi 28 class (huruf hijaiyah).

Table 2. Hasil Training

Epoch	1	2	3
Accuracy	0.8190	0.8726	0.9113
Precision	0.8340	0.8836	0.9144
Recall	0.8190	0.8726	0.9113
F1 Score	0.8189	0.8719	0.9113

Sumber: (Akil & Chaidir, 2020)

Berdasarkan hasil epoch di table 2, dapat kita lihat peningkatan akurasi, dimana pada epoch ke 3 akurasi adalah sebesar 0.9113 ini adalah hasil tertinggi yang didapat dengan model CNN diatas. Ada kemungkinan jika ditambah epoch nya akurasi akan bertambah, namun membutuhkan proses yang lebih lama. Perhatikan gambar 4 adalah matrix tingkat kesalahan dalam memprediksi dari masing-masing class.

Kesalahan yang terbesar atau sering terjadi adalah ketika memprediksi class dengan nomor 20 yaitu huruf ق yang dimungkinkan karena kemiripan dengan class 19 yaitu huruf ك. Sedangkan yang paling kecil tingkat kesalahannya adalah class 0 yaitu huruf ا.

```

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
-----
117 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 = 1
0 104 11 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 2 0 = 10
0 0 10 96 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 11 = 11
0 0 0 0 108 0 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 3 = 12
0 0 0 0 1 114 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 4 = 13
0 0 0 0 3 0 112 2 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 5 = 14
0 0 0 0 1 1 7 110 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 = 15
0 1 0 0 0 0 0 0 0 102 13 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 7 = 16
0 1 0 0 0 0 0 0 0 4 114 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 8 = 17
0 0 0 0 0 0 0 0 0 0 0 112 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 0 0 0 9 = 18
0 0 0 1 0 0 0 0 0 0 0 1 7 105 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0 0 10 = 19
0 0 0 0 0 0 0 0 0 0 0 0 1 0 116 0 0 0 0 0 0 0 2 0 0 0 0 1 0 0 0 0 0 0 11 = 2
0 0 0 0 0 0 2 0 1 0 0 0 0 0 1 108 3 0 0 0 0 4 0 0 1 0 0 0 0 0 0 0 0 0 12 = 20
0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 3 106 1 0 2 0 0 0 0 1 4 0 0 0 0 0 0 1 13 = 21
0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 112 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 14 = 22
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 117 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 15 = 23
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 118 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 16 = 24
0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 0 1 2 0 107 0 0 2 2 0 0 1 0 0 0 1 1 17 = 25
0 0 0 0 0 0 0 0 0 0 0 2 1 0 0 1 0 0 0 0 0 113 3 0 0 0 0 0 0 0 0 0 0 18 = 26
0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 111 0 0 0 0 0 0 0 1 0 0 19 = 27
0 0 0 0 0 1 0 0 0 0 0 1 0 0 2 0 0 0 0 1 0 0 0 0 115 0 0 0 0 0 0 0 0 20 = 28
0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 5 0 0 0 1 107 5 0 0 0 0 0 0 21 = 3
0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 111 0 0 0 0 0 0 22 = 4
0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 111 4 1 0 0 23 = 5
0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 116 0 0 0 24 = 6
0 0 0 0 0 0 0 0 0 0 0 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 107 0 0 25 = 7
0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 2 0 0 0 0 0 1 2 111 1 26 = 8
0 0 3 0 0 0 0 0 0 2 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 105 27 = 9

```

Sumber: (Akil & Chaidir, 2020)

Gambar 4. Confusion Matrix

Training ini dilakukan di PC dengan konfigurasi seperti berikut:

- Software:
 - OS: Windows 10 Pro
 - IDE: Apache Netbeans IDE 12
 - Java: jdk 1.8.0_25
- Hardware:
 - Processor: Processor Intel Pentium CPU G3220 @ 3.00 GHZ, GPU Intel HD 2 GB shared memory
 - Memory 8 GB
 - Hard Disk 500 GB

Jumlah waktu yang dibutuhkan untuk melakukan training sebanyak 3 epoch adalah: 05:17 menit.

KESIMPULAN

Dari hasil training dapat disimpulkan bahwa semakin banyak jumlah epoch akan semakin akurat namun banyak memakan sumberdaya CPU. Dalam proses training sudah dicoba beberapa fungsi aktivasi seperti; identity dan sigmoid, tapi hasil yang paling tinggi adalah dengan fungsi ReLU untuk layer convolutional. Sedangkan untuk subsampling tetap menggunakan fungsi max. Dari sumber data yang didapat masih ada kekurangan yaitu belum ada huruf sambung baik posisi didepan, ditengah dan dibelakang dari 28 class huruf hijaiyah tersebut yang membuat akan terdapat penambahan class yang cukup banyak, hal ini akan menjadi tantangan untuk penelitian selanjutnya.

REFERENSI

- Akil, I., & Chaidir, I. (2020). *Laporan Akhir Penelitian: Deteksi Karakter Huruf Arab Dengan Menggunakan Convolutional Neural Network*. Jakarta.
- Aqab, S., & Tariq, M. U. (2020). Handwriting recognition using artificial intelligence neural network and image processing. *International Journal of Advanced Computer Science and Applications*, 11(7), 137-146. <https://doi.org/10.14569/IJACSA.2020.0110719>
- Bi, X., Smith, B. A., & Zhai, S. (2015). *Multilingual Touchscreen Keyboard Design and Optimization CONTENTS 1. INTRODUCTION 2. BACKGROUND AND RELATED WORK 2.1 Input Methods for Different Languages. 2.2. QWERTY layout 2.3. Optimization Objectives 2.4. Optimization Scope 2.5. Optimization Methods*. 1-34.
- Brownlee, J. (2019). A Gentle Introduction to the Rectified Linear Unit (ReLU). Retrieved from Machine Learning Mastery website: <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>
- Dickson, B. (2020). What are convolutional neural networks (CNN)? Retrieved from bdtechtalks.com website: <https://bdtechtalks.com/2020/01/06/convolutional-neural-networks-cnn-convnets/>
- El Sawy, A., Loey, M., & El Bakry, H. (2019). Arabic Handwritten Characters Recognition using Convolutional Neural Network. *2019 10th International Conference on Information and Communication Systems, ICICS 2019*, 5, 147-151. <https://doi.org/10.1109/IACS.2019.8809122>
- Irina, G. (2011). Globalization and the World Crisis. *Researchgate*, pp. 135-166.
- Jaiem, F. K., Kanoun, S., Khemakhem, M., El Abed, H., & Kardoun, J. (2013). Database for Arabic printed text recognition research. *Image Analysis and Processing - ICIAP 2013, 8156 LNCS(PART 1)*, 251-259. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-41181-6_26
- Klassen, T. (2001). *Towards Neural Network Recognition Of Handwritten Arabic Letters* (Dalhousie University). Dalhousie University. Retrieved from <https://web.cs.dal.ca/~mheywood/Reports/TKlassen.pdf>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*. <https://doi.org/10.1201/9781420010749>
- LeCun, Y., Haffner, P., Bottou, L., & Bengio, Y. (1998). Object Recognition with Gradient-Based Learning. *AT&T Shannon Lab*, (0).
- LeCun, Y., Haffner, P., Bottou, L., & Bengio, Y. (1999). Object Recognition with Gradient-Based Learning. In D. A. Forsyth, J. L. Mundy, V. di Gesu, & R. Cipolla (Eds.), *Shape, Contour and Grouping in Computer Vision* (pp. 319-345). Berlin: Springer, Berlin, Heidelberg.
- Novet, J. (2014). SkyMind launches with open-source, plug-and-play deep learning features for your app. Retrieved from venturebeat.com website: <https://venturebeat.com/2014/06/02/skyind-launches-with-open-source-plug-and-play-deep-learning-features-for-your-app/>
- Prabhu, P. (2018). *Understanding of Convolutional Neural Network (CNN) — Deep Learning*. 279-291. https://doi.org/10.1142/9789811201233_0

- 014
- Rahaman, M. A., Mahin, M., Ali, M. H., & Hasanuzzaman, M. (2019). BHCDR: Real-Time Bangla Handwritten Characters and Digits Recognition using Adopted Convolutional Neural Network. *1st International Conference on Advances in Science, Engineering and Robotics Technology 2019, ICASERT 2019*, (April).
<https://doi.org/10.1109/ICASERT.2019.8934476>
- Scherer, D., Andreas, M., & Behnke, S. (2010). Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In K. Diamantaras, W. Duch, & L. S. Iliadis (Eds.), *Artificial Neural Networks – ICANN 2010* (pp. 92–101). Thessaloniki: Springer, Berlin, Heidelberg. Retrieved from https://link.springer.com/chapter/10.1007/978-3-642-15825-4_10
- Sharma, P. (2020). Keras Dense Layer Explained for Beginners. Retrieved from [machinelearningknowledge.ai/ website: https://machinelearningknowledge.ai/keras-dense-layer-explained-for-beginners/](https://machinelearningknowledge.ai/website:https://machinelearningknowledge.ai/keras-dense-layer-explained-for-beginners/)