

Komparasi Metode Backpropagation Neural Network dan Convolutional Neural Network Pada Pengenalan Pola Tulisan Tangan

(Comparison of *Backpropagation Neural Network and Convolutional Neural Network on Handwriting Pattern Recognition*)

A. A. SG. Mas Karunia Maharani*, Komang Oka Saputra, Ni Made Ary Esta Dewi Wirastuti

Program Studi Magister Teknik Elektro, Fakultas Teknik, Universitas Udayana

Jl. P. B. Sudirman Denpasar, Bali, INDONESIA

Email: karuniamaharani27@gmail.com, okasaputra@unud.ac.id, dewi.wirastuti@unud.ac.id

*Penulis Korespondensi

Abstract Historical manuscripts are one of documents that important to be preserved because they contain a lot of information, one example of them is script as the historical documents . Historical document mostly still use handwriting in so many reserch. Currently, there are many research regarding the preservation of characters. One way of preservation that can be used is the digitization process. Digitalizing's process tanable by recognizing existing information using technology. The technology that can be used is machine learning. Handwriting is a complex case because of the many variations of these characters and the output of the author where variations of the author will produce different writings. The relevant fields for text and documents are Optical Character Recognition (OCR) and handwriting recognition. There are several methods that can be used in the machine learning process, including Artificial Neural Network (ANN) and Convolutional Neural Network (CNN). Both of these methods are methods that can accept complex image input to be processed and recognized, therefore this method is highly recommended for processing handwriting.

Keywords: Handwritten Recognition, Deep Learning, Machine Learning, Artificial Neural Network, Convolutional Neural Network.

I. PENDAHULUAN

Naskah sejarah dan dokumentasi arsip adalah teks tulisan tangan yang merupakan sumber tulang punggung penyelidikan sejarah [1]. Dokumen sejarah adalah dokumen yang sangat berharga, karena dapat dijadikan sebagai sumber untuk menganalisa banyak hal seperti, sejarah, sosial ataupun ekonomi masa lalu. Salah satu cara dalam mempermudah proses analisis dokumen sejarah adalah dengan menggunakan proses digitalisasi dokumen. Dengan adanya digitalisasi, dokumen dan arsip tersebut dapat diakses secara langsung baik oleh para peneliti maupun publik. Dari proses digitalisasi, dapat ditarik kesimpulan dan analisa serta didapatkan informasi baru dari dokumen dan arsip-arsip tersebut [2]. Beberapa teknik analisa yang dapat diterapkan pada dokumen-dokumen ini adalah segmentasi halaman, *spotting* citra pada simbol dan

angka, *Optical Character Recognition* (OCR) [3] dan *Handwritten Text Recognition* (HTR) [4].

OCR adalah salah satu bidang penelitian pengenalan karakter yang dapat memproses data baik secara *offline* maupun *online*. OCR atau pengenalan karakter adalah proses yang mengaitkan makna simbolik dengan objek (huruf, simbol & angka) yang ada pada sebuah gambar [5]. OCR dapat mengenali gambar maupun pola dari suatu karekter. OCR umumnya dikenal sebagai pengenalan karakter karena menggunakan optical dalam pemrosesannya. Pengimplementasian OCR dapat dilakukan diberbagai ranah penelitian seperti, *structural text form recognition* [6], *automatic number plate recognition*[7], alat bantu baca tuna netra, pengenalan tulisan tangan (*handwritten*) dan sebagainya. OCR juga dapat digunakan untuk melestarikan dan mendigitalkan *script* [3] atau sastra kuno sebagai warisan [1] [8]. Identifikasi *script* merupakan langkah penting dalam pemrosesan gambar dokumen terutama ketika lingkungan multiskrip atau multibahasa [9]. *Handwritten recognition* juga saat ini mendapatkan perhatian peneliti karena aplikasi ini dapat mempermudah kehidupan manusia dengan memanfaatkan teknologi. Seperti halnya mengenali *script* kuno untuk penelitian sejarah, identifikasi citra tulisan, entri data otomatis untuk dokumen bisnis dan sebagainya [10]. Penelitian terkait *handwritten recognition* dianggap penting karena tulisan tangan atau *handwritten* merupakan media standar yang paling umum digunakan manusia dalam berkomunikasi serta komunikasi tersebut dapat mengandung suatu informasi di dalamnya [11].

Dalam proses mengenai karakter, tulisan tangan dan script dibutuhkan suatu teknologi yang dapat memproses dan mengenali tulisan tangan tersebut. Metode ini adalah *machine learning* (ML) atau pembelajaran mesin yang canggih diterapkan untuk mengekstrak fitur dan makna yang terkandung di dalam tulisan tangan [4]. *Machine learning* adalah pengembangan dari *Artificiall intelligent* yang menggunakan algoritma-algoritma dalam proses pembelajarannya. Hal ini dilakukan dengan mempelajari

pola-pola pada data yang dimasukkan ke algoritma tertentu dan mempelajari pola-pola tersebut serta menghasilkan keluaran baik berupa pengklasifikasian, pengenalan *feature* atau ciri dan *rule based* [12]. Dalam mengenali karakter ataupun tulisan, ada beberapa fitur ML yang digunakan yaitu *gradient features*, *chain codes* dan sebagainya. Beberapa metode yang dapat digunakan untuk proses pengklasifikasian tulisan dan fitur ekstraksi dalam machine learning yaitu seperti *Support Vector Machine* (SVM) dan *Artificial Neural Network* (ANN)[13]. Penelitian terkait *Urdu Nastalique Ligature Recognition* menggunakan *Backpropagation Neural Network* (BPP) [14], penelitian selanjutnya terkait *Signature Recognition* menggunakan metode ANN sebagai *classifier* [15], pengenalan karakter *urdu script* menggunakan metode BPP menghasilkan akurasi 92% [5] dan penelitian [16] terkait *recognition of offline Tulu Handwritten Scripts* juga menggunakan metode *machine learning*. Selain menggunakan metode *machine learning*, beberapa penelitian menggunakan pemrosesan yang lebih kompleks sehingga membutuhkan metode yang lebih *advance* yaitu *deep learning*.

Deep learning adalah proses pembelajaran yang lebih mendalam sehingga memungkinkan mesin untuk mempelajari informasi yang kompleks dan mengekstrak representasi data dari berbagai tingkatan informasi. *Convolutional Neural Network* (CNN) adalah salah satu metode *deep learning* yang diakui sebagai salah satu jenis ANN yang memiliki tingkat akurasi terbaik dalam proses pengenalan dan deteksi [17]. Jenis-jenis deteksi yang dapat dilakukan dengan menggunakan metode CNN yaitu *face recognition* [18], *target recognition* [19], *image classification* [20] dan *handwritten recognition* [21]. Penggunaan metode CNN pada penelitian terkait pengenalan pola tulisan tangan, *script* dan teks-teks sejarah sangat banyak dilakukan, seperti penelitian [22] pengenalan *arabic script* dengan menggunakan 28 *script arabic* dan 32 *character persian*. Penelitian selanjutnya mengenai *arabic numeral spotting and handwritten digit recognition* [4] menggunakan 70000 *dataset* dan menggunakan CNN sebagai model *classifiernya* terdapat 128 *features* yang diekstraksi dengan menggunakan KNN *classifier* untuk pengecekan data awal sebelum masuk ke CNN *classifier*, akurasi yang dihasilkan yaitu 99.47%. Dan penelitian terkait yaitu pengenalan karakter Tulisan Tangan Urdu dengan menggunakan metode CNN OCR-AlexNet and OCR-GoogleNet menghasilkan akurasi sebesar 96.3% and 94.7% [23].

Berdasarkan pemaparan diatas, jurnal ini akan membahas perbandingan metode ANN dan *deep learning* dalam proses pengenalan *handwritten recognition* pada beberapa jenis *script*. Metode ANN yang digunakan adalah BPP, sementara metode yang banyak digunakan pada *deep learning* adalah CNN. CNN memiliki model tersendiri yang telah dipatenkan, yaitu *pre-trained model* CNN. *Pre-trained* CNN dikembangkan untuk meningkatkan kinerja pembelajaran mesin pada CNN. Pemrosesan *handwriting* harus memerlukan metode yang

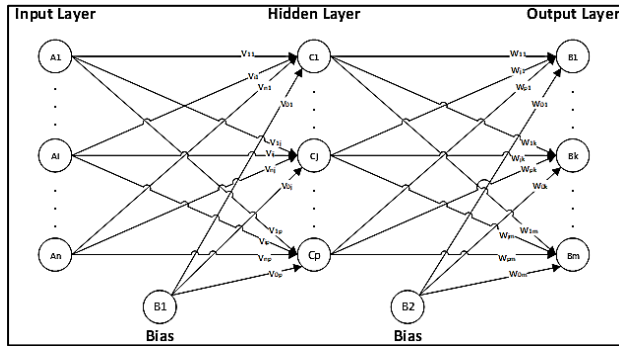
memiliki tingkat pemrosesan tinggi [24] [25], karena *script* dan tulisan tangan memiliki pola yang kompleks sehingga memerlukan metode yang baik dalam menangani hal ini [4][25]. Selain itu tantangan yang ada adalah ketidakseragaman bentuk dan ukuran karakter yang ditulis oleh penulis yang berbeda dan juga kesamaan antara karakter yang termasuk dalam *script* yang berbeda menambah tantangan pengenalan *script* dari gambar dokumen tulisan tangan [9].

II. METODELOGI PENELITIAN

Neural network (NN) adalah teknik pada *machine learning* (ML) yang terinspirasi dan menyerupai sistem saraf manusia dan struktur dari otak manusia itu sendiri. NN tersusun atas beberapa unit *layer* yaitu *input*, *hidden* dan *output layer*. *Node* atau unit yang terdapat pada tiap *layer* akan terhubung ke *node layer* yang berdekatan dan setiap hubungan akan memiliki *weight value* atau nilai bobot. *Input* dikalikan dengan masing-masing *bobot* dan dijumlahkan pada setiap unit. Jumlah tersebut kemudian mengalami transformasi berdasarkan fungsi aktivasi. Fungsi aktivasi yang banyak digunakan adalah *sigmoid*, *hiperbolik*, *tanh* dan *Rectified linear unit* (ReLU). Fungsi-fungsi ini digunakan karena memiliki turunan yang menguntungkan secara matematis, hal ini dapat mempermudah dalam proses perhitungan turunan yang berhubungan dengan bobot pada masing-masing unit. Fungsi *sigmoid* dan *tanh* memiliki rentang *output* yang sempit yaitu 0/1 dan -1/1. ReLU menunjukkan *saturating and non-saturating behavior* dengan menggunakan fungsi $f(x) = \max(0, x)$ [26]. *Output* dari fungsi tersebut akan diinputkan ke *node* berikutnya pada *layer* berikutnya. *Output layer* akan menghasilkan solusi dari permasalahan yang telah diproses pada NN.

A. Backpropagation Neural Network

Computer vision membutuhkan metode klasifikasi yang kuat untuk mencapai tingkat sistem pengenalan yang tinggi dengan waktu dan sumber daya komputasi yang rendah, salah satunya adalah *Backpropagation Neural Network* (BPNN) [27]. Klasifikasi BPNN banyak digunakan untuk melatih NN karena BPNN sederhana, efisien dalam menghitung penurunan gradien, dan mudah diimplementasikan [28]. Menentukan ukuran *neuron*, jumlah sampel dan bobot adalah tantangan dalam proses BPNN, dan hal yang sangat penting adalah menyesuaikan *output neuron*. Pada Gambar 1 [29] BPNN terdiri atas tiga tipe *layer*, yaitu *input layer*, *hidden layer* dan *output layer*. A melambangkan *unit input*, C melambangkan *unit hidden* dan B melambangkan unit output. V melambangkan bobot antara *unit input* (A) dan *unit hidden* (C), sedangkan W melambangkan bobot antara *unit hidden* (C) dan *unit output* (B).



Gambar 1 Arsitektur Backpropagation Neural Network [29].

Fungsi aktivasi pada proses *learning* BPP harus bersifat kontinu, terdiferensial dan fungsi yang tidak turun. Fungsi *sigmoid biner* merupakan salah satu fungsi yang memenuhi ketiga syarat tersebut dan memiliki range (0, 1) [29]. Persamaan (1) dan (2) menunjukkan fungsi aktivasi *sigmoid biner* yaitu sebagai berikut:

$$f(x) = \frac{1}{1+e^{-x}} \quad (1)$$

$$f'(x) = f(x)(1 - f(x)) \quad (2)$$

Algoritma *learning* pada BPP meliputi dua tahapan yaitu *feedforward* dan *backward*. Setelah melewati dua tahapan tersebut dilanjutkan dengan proses perubahan pembobotan atau *weight upgrade*. *Phase I*, *Phase II* dan *Phase III* merupakan tahapan-tahapan yang akan dilakukan pada pemrosesan BPP yaitu sebagai berikut [29]:

Step 1 : Melakukan inisialisasi bobot dimana pemilihan bobot awal dilakukan secara acak dengan nilai yang cukup kecil .

Step 2 : *Step 3* sampai 10 merupakan *step* yang dilakukan ketika kondisi penghentian belum terpenuhi.

Langkah 3 : *Step 4* sampai 9 merupakan *step* yang harus dilalui pada proses *testing* data.

Feedforward (Phase I)

Step 4 : Untuk setiap *node input* ($A_i, i = 1, 2, \dots, n$) yang menerima informasi dari *data input* akan meneruskan informasi *data* tersebut ke *hidden layer*

Step 5 : Melakukan perhitungan untuk output pada hidden layer ($C_j, j = 1, 2, \dots, p$)

$$C_net_j = v_{0j} + \sum_{i=1}^n A_i v_{ij} \quad (3)$$

Untuk menghitung *output signal* dapat digunakan *activation function* pada persamaan (4):

$$C_j = (C_net_j) \quad (4)$$

Output signal pada *hidden layer* dikirim kepada *node-node layer* selanjutnya atau *node output*. *Step* ini dilakukan berulang kali sebanyak jumlah *hidden layer*nya.

Step 6 : Melakukan proses perhitungan *data output* dari *hidden layer* pada *output layer* ($B_k, k = 1, 2, \dots, m$)

$$B_net_k = w_{0k} + \sum_{j=1}^p B_j w_{jk} \quad (5)$$

Untuk menghitung *output signal* dapat digunakan *activation function* pada persamaan (6):

$$B_k = f(B_net_k) \quad (6)$$

Backpropagation (Phase II)

Step 7 : Proses perhitungan faktor δ dari *node output* berdasarkan *error* di setiap *node output* ($b_k, k = 1, 2, \dots, m$)

$$\delta_k = (t_k - b_k) f'(b_net_k) \quad (7)$$

δ adalah *error unit* yang akan digunakan pada proses pembobotan pada *layer* di bawahnya (langkah 8). $f'(b_net_k)$ adalah turunan dari *sigmoid biner activation function*.

Langkah selanjutnya yaitu *weight correction*, dimana hasilnya akan digunakan untuk mengupgrade nilai w_{jk} dengan laju percepatan α

$$\Delta w_{jk} = \alpha \cdot \delta \cdot c_j \quad (8)$$

Selanjutnya dilakukan perhitungan *bias correction* dimana hasilnya akan digunakan untuk mengupgrade nilai w_{0k}

$$\Delta w_{0k} = \alpha \cdot \delta_k \quad (9)$$

Step 8 : Melakukan proses perhitungan faktor δ pada *hidden layer* berdasarkan *error* di tiap *hidden node* ($c_j, j = 1, 2, \dots, p$)

$$\delta_net_j = \sum_{k=1}^m \delta_k \cdot w_{jk} \quad (10)$$

Faktor δ *hidden node* :

$$\delta_j = \delta_net_j f'(c_net_j) \quad (11)$$

Langkah selanjutnya yaitu *weight correction*, dimana hasilnya akan digunakan untuk mengupgrade nilai v_{ij}

$$\Delta v_{ij} = \alpha \cdot \delta_j \cdot a_i \quad (12)$$

Selanjutnya dilakukan perhitungan *bias correction* dimana hasilnya akan digunakan untuk mengupgrade nilai v_{0j}

$$\Delta v_{0j} = \alpha \cdot \delta_j \quad (13)$$

Weight Upgrade (Phase III)

Step 9 : masing-masing *output node* ($B_k, k = 1, 2, \dots, m$) melakukan *upgrade weight* ($j = 0, 1, 2, \dots, p$)

$$w_{jk} (new) = w_{jk} (old) + \Delta w_{jk} \quad (14)$$

Masing-masing *node* pada *hidden layer* ($C_j, j = 1, 2, 3, \dots, p$) *upgrade weight* ($j = 0, 1, 2, 3, \dots, n$)

$$v_{ij} (new) = v_{ij} (old) + \Delta v_{ij} \quad (15)$$

Step 10 : kondisi pada *training proses* berhenti.

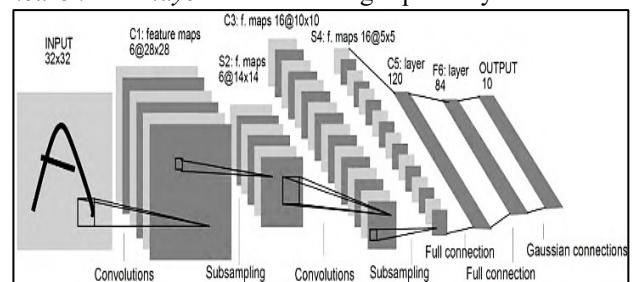
Dilakukan proses perulangan pada *Phase I*, *Phase II* dan *Phase III* secara terus menerus hingga kondisi penghentian dipenuhi. Parameter kondisi penghentian yang banyak digunakan pada penelitian-penelitian sejenis adalah jumlah iterasi atau *error*. Jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, maka proses perhitungan iterasi akan berhenti, atau jika *error* yang ditetapkan sudah lebih kecil dari batas toleransi, maka proses *learning* akan selesai [28].

B. Convolution Neural Network

Deep neural network (DNN) terdiri atas beberapa *layer node*. Arsitektur DNN berbeda telah dikembangkan untuk memecahkan masalah yang berbeda-beda tergantung pada *case* dan penggunaannya [26]. *Convolutional Neural Network* (CNN) paling sering digunakan pada pemrosesan *computer vision* dan *image recognition*. CNN adalah salah satu metode pada jaringan saraf tiruan yang dapat melakukan proses ekstraksi fitur sekaligus melakukan proses klasifikasi dari citra *input* [31] [32]. *Hidden Markov Model* (HMM) adalah salah satu metode *image recognition* yang memerlukan proses segmentasi dalam pemrosesan citranya. *Character image segmentation* adalah salah satu metode yang paling rumit dalam *pattern recognition*, dikarenakan perbedaan fitur *pattern* pada masing-masing citra yang akan disegmentasikan [33] [34]. Beda halnya dengan metode CNN, CNN dapat memproses citra tanpa bantuan segmentasi [35][36]. CNN tidak memerlukan proses segmentasi citra karena CNN dapat melakukan *recognition image* pada arsitektur *Lenet* dan sekaligus melakukan ekstraksi citra dengan bantuan *kernel* [31].

Pemilihan metode klasifikasi yang tepat sangat diperlukan sesuai dengan faktor-faktor yang mendukung tingkat pemrosesan untuk menyelesaikan masalah. CNN terdiri atas serangkaian *layer convolution* yang diikuti oleh *layer* yang saling terhubung (*fully connected layer*) dan *normalizing layer*. Berdasarkan Gambar 2, terdapat arsitektur CNN 7-layer pada pemrosesan *character recognition*. Rangkaian dari beberapa *convolution layers* menghasilkan fitur-fitur terbaik dari setiap *layernya* dimana proses ini dimulai dari *input layer* hingga *output layer*. Proses klasifikasi akan terus berlanjut pada tiap *convolution layer* karena CNN merupakan *fully connected layer*. *Pooling layer* akan terus disisipkan diantara *convolution layer*. CNN mengambil gambar 2D $n \times n$ piksel sebagai *input*. Setiap lapisan terdiri dari kelompok neuron 2D yang disebut *filter* atau *kernel*. Tidak seperti jaringan saraf lainnya, *neuron* di setiap lapisan ekstraksi fitur CNN tidak terhubung ke semua *neuron* di lapisan yang berdekatan. Sebaliknya, *neuron* hanya terhubung ke *neuron* berukuran tetap yang telah dipetakan secara spasial dan sebagian *overlapping* gambar *input* atau *feature map* dari lapisan sebelumnya. Area *input* ini disebut *local receptive field*. Jumlah koneksi yang lebih sedikit dapat mengurangi waktu pelatihan dan kemungkinan *overfitting*.

Semua *neuron* dalam *filter* terhubung ke jumlah *neuron* yang sama di lapisan *input* sebelumnya (atau peta fitur) dan dibatasi bobot dan biasanya. Faktor-faktor ini dapat mempercepat proses pembelajaran dan mengurangi kebutuhan memori. Setiap *neuron* dalam *filter* mencari pola yang sama, namun pola tersebut akan berbeda dengan pola gambar inputnya. Lapisan *subsampling* akan mengurangi ukuran jaringan. Selain itu, *local receptive field* dan *bobot* (dalam filter yang sama), dapat mengurangi kerentanan jaringan terhadap pergeseran, skala dan distorsi gambar. Filter *max/mean pooling* atau *local averaging* digunakan untuk mencapai *subsampling*. *Output layer* CNN melakukan proses klasifikasi, dimana *neuron* antar *layer* akan terhubung sepenuhnya.



Gambar 2. Arsitektur CNN 7 layer pada character recognition [26].

Deep CNN dapat diimplementasikan dengan beberapa seri pembagian *convolution layer* bobot dan *subsampling layer*. CNN dapat menghasilkan representasi berkualitas tinggi dengan mempertahankan lokalitas, mengurangi parameter, dan invarian terhadap variasi kecil pada gambar input. Dalam kebanyakan kasus, *backpropagation* digunakan hanya untuk pelatihan semua parameter (bobot dan bias) di CNN. Implementasi CNN yang berbeda telah menunjukkan peningkatan akurasi yang berkelanjutan dalam *computer vision*. Berikut adalah variasi dan implementasi yang terkenal dari arsitektur CNN diantaranya yaitu, AlexNet, Inception, ResNet, VGG dan DCGAN [26]. AlexNet [37] adalah arsitektur CNN yang dikembangkan untuk dapat melakukan pemrosesan data pada platform komputasi paralel Nvidia dan mendukung kinerja GPU. Inception [38] adalah arsitektur CNN mendalam yang dikembangkan oleh Google. ResNet [39] adalah *residual network* yang sangat mendalam dan dikembangkan oleh Microsoft. Arsitektur ini menang dalam kompetisi ILSFRC 2015 pada dataset ImageNet. VGG [40] adalah arsitektur CNN yang sangat dalam dikembangkan untuk pengenalan gambar skala besar. Dan DCGAN [41] adalah *generative adversarial network* yang dapat digunakan dalam pembelajaran hierarki representasi fitur tanpa pengawasan dalam objek *input*.

III. HASIL DAN PEMBAHASAN

Machine learning dan *computer vision* adalah bidang ilmu yang dapat melakukan pengolahan data pada *image processing*. Beberapa penelitian telah menggunakan teknik pembelajaran mesin untuk OCR. Teknik OCR berbasis segmentasi terdiri dari beberapa langkah, dimana setiap langkah melakukan tugas *task* tertentu untuk menghasilkan *output* yang akan diteruskan pada langkah-

langkah pengerjaan berikutnya. Langkah-langkah yang dilakukan dalam pengolahan citra atau *image processing* adalah *preprocessing*, segmentasi, ekstraksi fitur dan pengklasifikasian berbasis *machine learning* [42]. Teknik-teknik *preprocessing* yang sering digunakan dalam mengolah citra adalah *smoothing* [43], *binarization*, *normalization*, *greyscale* [5] dan sebagainya. Hal ini dilakukan untuk meningkatkan fitur gambar dan mengurangi *noise* pada gambar inputan. *Features reduction* adalah proses pengurangan fitur dan hal ini sangat penting dalam proses *recognition*, karena hal ini dapat meningkatkan waktu saat klasifikasi dilakukan dan mengurangi tingkat *error* saat *computing* proses sedang berjalan [27]. Setelah masuk ke tahap *preprocessing*, citra yang telah siap diolah akan masuk ketahapan ekstraksi fitur. Pada ekstraksi fitur, metode yang digunakan beragam dan tujuan utama dari proses ekstraksi fitur adalah mempelajari fitur-fitur atau ciri yang ada pada citra. Dimana nanti fitur-fitur tersebut yang memberikan informasi untuk pembelajaran mesin di tahap klasifikasi. Setelah fitur-fitur dari suatu citra didapatkan, maka dilanjutkan dengan proses klasifikasi citra dengan menggunakan *classifier*. Pemilihan *classifier* tergantung dengan apa dan berapa *output* yang akan diklasifikasikan [13]. Untuk mengetahui seberapa bagus *performance* suatu metode, dapat diukur dengan menggunakan perhitungan akurasi, presisi dan *recall* [44]. Pengukuran *performance* ini dilakukan dengan menggunakan *confusion matrix*. Perhitungan ini dilakukan untuk mengetahui performa sistem apakah sudah sesuai dengan *output* yang ditargetkan serta mengetahui seberapa besar nilai *error* yang masih terjadi saat dilakukan proses *learning*. Perhitungan akurasi, presisi dan *recall* pada *confusion matrix* dapat dilihat pada rumus sebagai berikut [45] :

$$\text{Akurasi} = \frac{\text{jumlah data sesuai target}}{\text{Total keseluruhan data}} \quad (16)$$

$$\text{Presisi} = \frac{\text{jumlah data yang sesuai target di satu kelas}}{\text{jumlah seluruh data yang sesuai target}} \quad (17)$$

$$\text{Recall} = \frac{\text{jumlah data yang sesuai target di satu kelas}}{\text{jumlah data di satu kelas}} \quad (18)$$

Beberapa penelitian menjabarkan proses pengolahan data dengan menggunakan *classifier* yang berbeda, khususnya pada OCR dan *script recognition*. Penelitian [5] mengenai OCR dengan menggunakan metode BPNN sebagai *classifiernya*. Langkah-langkah yang dilakukan pada penelitian ini yang pertama yaitu *image acquisition*. *Image acquisition* adalah proses pengambilan *input data digital*, dalam penelitian ini menggunakan *digital camera* dan *scanner* dengan menggunakan format JPEG sebagai *input image*. Langkah selanjutnya adalah *preprocessing*, hal ini dilakukan untuk menyiapkan data sebelum masuk ke proses berikutnya. *Preprocessing* yang dilakukan adalah *greyscale*, tujuannya itu untuk meringankan beban komputasi karena *value* dari citra berentang 0 hingga 255 *pixel*, dimana 0 merepresentasikan *black pixel* dan 155

merepresentasikan warna putih. *Preprocessing* selanjutnya yang dilakukan adalah *noise reduction*, tujuannya adalah untuk meningkatkan kualitas dari citra. *Noise reduction* yang digunakan adalah *median filtering technique*. Dilanjutkan dengan teknik *preprocessing thresholding*, tujuannya adalah untuk melakukan *setting background* dan *foreground* [46], teknik *thresholding* yang dilakukan adalah *binarization*. *Binarization* adalah merubah citra *input* menjadi *binary image* dan langkah *preprocessing* terakhir yang dilakukan adalah *normalization*. *Normalization* adalah proses menyamakan ukuran semua *bitmap* karakter yang diekstraksi (citra *biner*). Ada banyak variasi ukuran karakter. Untuk membuat karakter yang diekstraksi dalam matriks berdimensi sama, dilakukan normalisasi. Matriks karakter yang dinormalisasi berukuran 12x8. Setelah proses *preprocessing* dilakukan proses *feature extraction* dan *classification* dengan menggunakan BPNN menghasilkan akurasi sebesar 97% dengan arsitektur 1 *input hidden output layer*, dengan jumlah *neuron* 96,79 dan 62 pada masing-masing *layer*nya.

Penelitian [15] selanjutnya terkait BPNN *method* adalah penggunaan algoritma BPNN untuk pengenalan tanda tangan menggunakan citra digital diubah menjadi *grayscale* untuk mengurangi rentang warna 3D menjadi 1D, kemudian citra berukuran 50x50 untuk meminimalkan *input NN* sehingga waktu pelatihan BPNN dengan jumlah *neuron input* 2500, jumlah *neuron* pada HL 1 150 dan jumlah *neuron* pada HL 2 150 dan jumlah *neuron* keluaran 27. *Learning rate* 0,005 dengan target *error* 0,0000005 dan BPNN mencapai 86,3% tingkat pengenalan. Peneliti lain [27] melakukan pengenalan wajah menggunakan LBPH sebagai metode fitur reduksi dan metode KNN serta BPNN sebagai metode klasifikasi. Perlu dilakukan penambahan langkah setelah ekstraksi fitur dan pengurangan dimensi untuk mendapatkan perbedaan yang jelas *T-Dataset*, yang akan digunakan untuk melatih BPNN kami mencapai akurasi 97,7% pada *dataset* Yale dengan hanya dua ketidakcocokan dari 165 gambar dan 98% akurasi dengan empat ketidakcocokan dari 200 gambar pengujian pada set pelatihan 50% dan skenario pengujian 50%.

Penelitian selanjutnya yaitu mengenai pengenalan karakter Urdu. Urdu adalah salah satu bahasa primitif dan penting dari sub daratan India. Di seluruh dunia bahasa Urdu dituturkan di lebih dari 15 negara dengan perkiraan lebih dari 250 juta penutur. Koleksi besar literatur Urdu yang signifikan dari penyelidikan Islam hingga sains dapat diakses, yang belum didigitalkan. Sistem pengenalan *teks Urdu* berdasarkan skrip telah menjadi bidang penelitian penting selama beberapa dekade, yang dapat digunakan sebagai bagian dari metode pembelajaran dan pengajaran kepada pelajar dan bagaimana menyelidiki dan menangani substansi instruktif dari konten bahasa Urdu. Selama beberapa dekade terakhir, tidak banyak penelitian yang berkonsentrasi pada membangun kerangka pengakuan optimal untuk konten bahasa Urdu [25]. Penelitian [22] menggunakan *dataset* yang berisi 800 gambar dari

masing-masing 38 karakter Urdu dan 10 angka. *Dataset* berasal dari 500 penutur asli bahasa Urdu dari berbagai kelompok sosial. Setelah pengumpulan dataset, halaman teks dipindai dengan *pixel* 300 DPI dan disegmentasi secara manual menjadi gambar 28 kali 28 untuk setiap karakter Urdu dan data angka. Seperti disebutkan sebelumnya, dataset terdiri dari $800 \times 10 = 8000$ gambar angka dan 800×38 adalah 30.400 karakter Urdu. Atribut yang digunakan pada arsitektur NN adalah *learning rate* 0,0025 dengan ukuran *batch* 132 rata-rata tingkat akurasi 98,03%. Fungsi ReLU dapat menangani masalah *gradien vanishing* dengan lebih baik jika dibandingkan dengan fungsi *sigmoid* [22][31][14]. Penelitian mengenai pengenalan script Urdu juga dilakukan pada penelitian [3] yaitu pengenalan karakter bahasa Urdu Berisi dua *dataset Dataset-1* yang berisi 1.446 kelas ligatur dari 3.309.762 *instance ligatur* dan *Dataset-2* yang merupakan *subset* dari *dataset* UPTI yang memiliki 1.600 gambar baris teks yang digunakan. Setiap gambar RASM *ligatur* ditipiskan dan dilintasi untuk menghitung DCT sebagai fitur yang memberikan akurasi pengenalan karakter 98,37%.

Penelitian [22] menggunakan 35 penulis dari kelompok umur yang berbeda, tiga pengklasifikasi yang berbeda yaitu, Jaringan Syaraf Tiruan *Backpropagation* (BPNN), Jaringan Syaraf Probabilistik (PNN), dan *classifier* berbasis korelasi. Pendekatan yang diusulkan diuji pada 85 contoh karakter single-stroke yang diambil dari 35 penulis dari kelompok usia yang berbeda. Hasil penelitian menunjukkan bahwa pengklasifikasi PNN mencapai akurasi yang lebih tinggi yaitu 95% dibandingkan dengan dua pengklasifikasi lainnya. Tidak seperti BNN, pengklasifikasi berbasis PNN tidak memerlukan pelatihan awal. Inilah alasan mengapa pengklasifikasi berbasis PNN mencapai akurasi yang lebih tinggi daripada BNN. Penelitian [9] mengenai pengenalan aksara India berisi 7200 sampel kata tulisan tangan milik 12 aksara India (600 per skrip) yang diklasifikasikan menggunakan pengklasifikasi *Multi Layer Perceptron* (MLP) dengan akurasi maksimum 98,45%. Selain penelitian mengenai pengenalan Urdu dan Indian Script.

Pengenalan tulisan arab juga dilakukan sebuah penelitian. Aksara Arab diadopsi secara luas dalam manuskrip dari berbagai negara dan budaya, misalnya Utsmaniyah, Arab, Urdu, Kurdi, dan Persia. Arab berisi 28 dan Persia berisi 32 karakter [22]. *Script* ini dapat ditulis dengan cara yang berbeda, yang memperumit proses segmentasi halaman, pencarian kata kunci, HTR dan OCR. Properti ini mungkin tidak menyebabkan masalah untuk pengenalan digit karena digit terisolasi, namun ketika pencarian kata kunci dan algoritma pengenalan teks tulisan tangan diterapkan, maka akan menciptakan tantangan baru. Penelitian [4] menggunakan 70000 data dengan pembagian 60000 data training dan 10000 *data testing* dengan menggunakan metode CNN, terdapat 128 fitur yang diekstraksi dengan menggunakan KNN *classier* untuk pengecekan data awal sebelum masuk ke CNN *classifier*, akurasi yang dihasilkan yaitu 99,47%. Penelitian [23] yaitu pengenalan karakter Tulisan Tangan Urdu dengan

menggunakan metode CNN OCR-AlexNet dan OCR-GoogleNet menghasilkan akurasi kinerja sebesar 96,3% dan 94,7%. Pada Tabel I, dapat dilihat akurasi dari beberapa penelitian mengenai *handwritten recognition* dengan menggunakan beberapa metode dengan *dataset* yang beragam.

TABLE I. AKURASI PENELITIAN HANDWRITTEN RECOGNITION MENGGUNAKAN BERAGAM METODE DAN DATASET

Reference	Year	Method	Data Used	Accuracy (%)
[40]	2021	CNN VGGNet-19	Recognition of Arabic Handwritten Scripts	96.6
[40]	2021	CNN MobileNet	Recognition of Arabic Handwritten Scripts	95.72
[25]	2020	CNN + AlexNet	Urdu Handwritten Character Recognition	96.3
[25]	2020	CNN + GoogleNet	Urdu Handwritten Character Recognition	94.7
[33]	2017	EDMS features + BPP-ANN	Arabic Words Recognition using 22 Attributes of feature	97.77
[31]	2020	CNN + Lenet5	Java Script Recognition using 20 classes 6000	97.6
[23]	2018	CNN + LeNet	Arabic handwritten recognition using HODA dataset	97.38
[47]	2019	CNN + CapsNet	Arabic handwritten recognition using HODA dataset	99.87
[48]	2017	CNN + AlexNet	Arabic handwritten recognition using HODA dataset	99.44
[4]	2020	CNN	Arabic handwritten recognition using CmaterDb3.3.1 dataset	99.76
[49]	2019	CNN	Arabic handwritten recognition using CmaterDb3.3.1 dataset	99.4
[50]	2020	CNN	Arabic handwritten recognition using HODA dataset use 70000 data	99.47

IV. KESIMPULAN

Pemilihan metode klasifikasi yang tepat sangat diperlukan sesuai dengan faktor-faktor yang mendukung tingkat pemrosesan untuk menyelesaikan masalah. CNN dapat memproses *image recognition* dan *video recognition*. Dengan menentukan ukuran dari *neural network*, jumlah *sample*, *weight* dan arsitektur NN dapat disesuaikan dengan berapa banyak jumlah *dataset* dan disesuaikan dengan *output* yang diharapkan. Karena perbedaan masalah dapat diselesaikan dengan arsitektur dan *settingan* atribut yang berbeda. Untuk masalah pengenalan pola yang kompleks seperti pengenalan karakter (OCR), *handwritten recognition*, *image processing* sangat sulit untuk menentukan pengklasifikasi mana yang akan menjadi pilihan terbaik. Menggabungkan *classifier* adalah langkah yang tepat, dapat bermanfaat dibandingkan dengan menggunakan pengklasifikasi tunggal. Selain itu pemilihan

arsitektur dan atribut terkait dengan metode yang akan digunakan merupakan hal yang penting. Baik *input* dan *output* harus ditetapkan dengan jelas alur dan prosesnya. Dan yang tidak kalah penting juga adalah *data*, karena *data* sangat berpengaruh terhadap hasil kalsifikasi *classifier*. Data mentah tidak dapat diolah secara langsung, melainkan melalui tahapan-tahapan atau *preprocessing*. Berdasarkan penelitian-penelitian terkait dengan OCR dan *handwritten recognition*, ANN dan CNN memiliki performa yang baik. Tak luput dari itu kedua *classifier* tetap memiliki beberapa kekurangan diantaranya yaitu beban komputasi yang tinggi dan waktu pemrosesan yang cukup lama. Maka dari itu harus dipersiapkan dengan matang saat menggunakan salah satu *classifier* baik dari sisi *cost*, *timing* dan *memory*.

REFERENCES

- [1] S. Leonelli and N. Tempini, *Data Journeys in the Sciences*. 2020.
- [2] Y. S. Can and M. Erdem Kabadayı, "CNN-based page segmentation and object classification for counting population in ottoman archival documentation," *J. Imaging*, vol. 6, no. 32, 2020.
- [3] Q. U. A. Akram and S. Hussain, "Improving Urdu Recognition Using Character-Based Artistic Features of Nastalique Calligraphy," *IEEE Access*, vol. 7, no. c, pp. 8495–8507, 2019.
- [4] Y. S. Can and M. E. Kabadayı, "Automatic CNN-based arabic numeral spotting and handwritten digit recognition by using deep transfer learning in ottoman population registers," *Appl. Sci.*, vol. 10, no. 16, 2020.
- [5] S. Afroge, B. Ahmed, and F. Mahmud, "Optical character recognition using back propagation neural network," *ICECTE 2016 - 2nd Int. Conf. Electr. Comput. Telecommun. Eng.*, no. November 2017, 2017.
- [6] A. Entezami, H. Sarmadi, and B. Behkamal, "Big Data Analytics and Structural Health Monitoring :," 2020.
- [7] A. S. Agbemenu and E. O. Addo, "An Automatic Number Plate Recognition System using OpenCV and Tesseract An Automatic Number Plate Recognition System using OpenCV and Tesseract OCR Engine," no. May 2018, 2019.
- [8] T. Nasir, M. K. Malik, and K. Shahzad, "MMU-OCR-21: Towards End-to-End Urdu Text Recognition Using Deep Learning," *IEEE Access*, vol. 9, pp. 124945–124962, 2021.
- [9] A. Mukhopadhyay, P. K. Singh, R. Sarkar, and M. Nasipuri, "A study of different classifier combination approaches for handwritten indic script recognition," *J. Imaging*, vol. 4, no. 2, 2018.
- [10] M. Jangid and S. Srivastava, "Handwritten Devanagari character recognition using *layer*-wise training of deep convolutional neural networks and adaptive gradient methods," *J. Imaging*, vol. 4, no. 2, 2018.
- [11] D. Keyzers, T. Deselaers, H. A. Rowley, L. L. Wang, and V. Carbune, "Multi-Language Online Handwriting Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1180–1194, 2017.
- [12] N. S. Sani, M. A. Rahman, A. A. Bakar, S. Sahran, and H. M. Sarim, "Machine learning approach for Bottom 40 Percent Households (B40) poverty classification," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 8, no. 4–2, pp. 1698–1705, 2018.
- [13] Y. Lai, X. Jia, and Y. Chi, "Digital conversion model for hand-filled forms using optical character recognition (OCR) Digital conversion model for hand-filled forms using optical character recognition (OCR), *IEEE Access*, vol. 9, pp. 12, 2020."
- [14] K. U. U. Rehman and Y. D. Khan, "A Scale and Rotation Invariant Urdu Nastalique Ligature Recognition Using Cascade Forward Backpropagation Neural Network," *IEEE Access*, vol. 7, pp. 120648–120669, 2019.
- [15] Y. Inan, "Signature Recognition Using Backpropagation Neural Network Signature Recognition Using Backpropagation Neural Network Biometric systems are security systems that enable people to access the system by," vol. 1, no. January, pp. 262–269, 2019.
- [16] C. K. Savitha and P. J. Antony, "Machine Learning Approaches for recognition of offline Tulu Handwritten Scripts," *J. Phys. Conf. Ser.*, vol. 1142, no. 1, 2018.
- [17] H. Abdellatef, M. Khalil-Hani, N. Shaikh-Husin, and S. O. Ayat, "Stochastic computing correlation utilization in Convolutional Neural Network basic functions," *Telkomnika (Telecommunication Comput. Electron. Control)*, vol. 16, no. 6, pp. 2835–2843, 2018.
- [18] I. M. Erwin, R. Risnandar, E. Prakarsa, and B. Sugiarto, "Kayu7net: Identifikasi dan Evaluasi F-Measure Citra Kayu berbasis Deep Convolutional Neural Network (DCNN)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 6, p. 1089, 2020.
- [19] M. S. Islam, M. S. Sultana, M. U. Kumar, J. Al Mahmud, and S. J. Islam, "HARC-New Hybrid Method with Hierarchical Attention Based Bidirectional Recurrent Neural Network with Dilated Convolutional Neural Network to Recognize Multilabel Emotions from Text," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 7, no. 1, p. 142, 2021.
- [20] N. N. A. A. Hamid, R. A. Razali, and Z. Ibrahim, "Comparing bags of features, conventional convolutional neural network and alexnet for fruit recognition," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 14, no. 1, pp. 333–339, 2019.
- [21] T. S. Gunawan, A. F. R. M. Noor, and M. Kartiwi, "Development of english handwritten recognition using deep neural network," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 10, no. 2, pp. 562–568, 2018.
- [22] M. Husnain *et al.*, "Recognition of urdu handwritten characters using convolutional neural network," *Appl. Sci.*, vol. 9, no. 13, 2019.
- [23] N. Altwaijry, "Arabic handwriting recognition system using convolutional neural network," *Neural Comput. Appl.*, vol. 33, no. 7, pp. 2249–2261, 2021.
- [24] Y. S. Can and M. E. Kabadayı, "applied sciences Automatic CNN-Based Arabic Numeral Spotting and Handwritten Digit Recognition by Using Deep Transfer Learning in Ottoman Population Registers," no. August, 2020.
- [25] K. O. Mohammed Aarif and S. Poruran, "OCR-Nets: Variants of Pre-trained CNN for Urdu Handwritten Character Recognition via Transfer Learning," *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 2294–2301, 2020.
- [26] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.
- [27] M. A. Abuzneid and A. Mahmood, "Enhanced human face recognition using LBPH descriptor, multi-KNN,

- and back-propagation neural network,” *IEEE Access*, vol. 6, pp. 20641–20651, 2018.
- [28] A. Gao, Y. Zhu, W. Cai, and Y. Zhang, “Pattern Recognition of Partial Discharge Based on VMD-CWD Spectrum and Optimized CNN with Cross-Layer Feature Fusion,” *IEEE Access*, vol. 8, pp. 151296–151306, 2020.
- [29] A. A. S. Mas, K. Maharani, I. G. Pasek, S. Wijaya, and F. Bimantoro, “PENGENALAN POLA TULISAN TANGAN AKSARA SASAK MENGGUNAKAN METODE LINEAR DISCRIMINANT ANALYSIS DAN JARINGAN SYARAF TIRUAN JENIS BACKPROPAGATION (Handwritten Sasak Script Recognition Using Linear Discriminant Analysis And Backpropagation Artificial Neural Network),” pp. 1–12, 2020.
- [30] A. Jumarwanto, R. Hartanto, and D. Prastiyanto, “Aplikasi Jaringan Saraf Tiruan Backpropagation Untuk Memprediksi Penyakit THT Di Rumah Sakit Mardi Rahayu Kudus,” *J. Tek. Elektro*, vol. 1, no. 1, pp. 11–21, 2009.
- [31] E. P. N. A. Wijaya, H. Armanto, and L. Zaman, “Klasifikasi Akasara Jawa dengan CNN,” *J. Tek.*, vol. 13, no. 2, pp. 61–64, 2020.
- [32] N. Kasim and G. S. Nugraha, “Pengenalan Pola Tulisan Tangan Aksara Arab Menggunakan Metode Convolution Neural Network,” *J. Teknol. Informasi, Komputer, dan Apl. (JTika)*, vol. 3, no. 1, pp. 85–95, 2021.
- [33] J. Ramadan, K. Omar, and M. Faizul, “A novel method to detect segmentation points of arabic words using peaks and neural network,” *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 7, no. 2, pp. 625–631, 2017.
- [34] B. R. Kavitha and C. Srimathi, “Benchmarking on offline Handwritten Tamil Character Recognition using convolutional neural networks,” *J. King Saud Univ. - Comput. Inf. Sci.*, no. 876-889, 2019.
- [35] M. F. Mridha, A. Q. Ohi, J. Shin, M. M. Kabir, M. M. Monowar, and M. A. Hamid, “A Thresholded Gabor-CNN Based Writer Identification System for Indic Scripts,” *IEEE Access*, vol. 9, pp. 132329–132341, 2021.
- [36] A. A. Chandio, A. A. Chandio, M. Asikuzzaman, and M. R. Pickering, “Cursive Character Recognition in Natural Scene Images Using a Multilevel Convolutional Neural Network Fusion,” *IEEE Access*, vol. 8, pp. 109054–109070, 2020.
- [37] S. Y. Arafat and M. J. Iqbal, “Two Stream Deep Neural Network for Sequence-Based Urdu Ligature Recognition,” *IEEE Access*, vol. 7, pp. 159090–159099, 2019.
- [38] M. Aqib, R. Mehmood, A. Alzahrani, I. Katib, A. Albeshri, and S. M. Altowaijri, *Smarter traffic prediction using big data, in-memory computing, deep learning and gpus*, vol. 19, no. 9. 2019.
- [39] S. Y. Arafat and M. J. Iqbal, “Urdu-Text Detection and Recognition in Natural Scene Images Using Deep Learning,” *IEEE Access*, vol. 8, pp. 96787–96803, 2020.
- [40] R. Ahmed *et al.*, “Deep neural network-based contextual recognition of arabic handwritten scripts,” *Entropy*, vol. 23, no. 3, pp. 4–6, 2021.
- [41] N. Sabri, Z. A. Aziz, Z. Ibrahim, M. A. R. B. M. Rosni, and A. H. bin Abd Ghapul, “Comparing Convolution Neural Network models for leaf recognition,” *Int. J. Eng. Technol.*, vol. 7, no. 3, pp. 141–144, 2018.
- [42] D. Firmani, P. Merialdo, E. Nieddu, and S. Scardapane, “In Codice Ratio: OCR of Handwritten Latin Documents using Deep Convolutional Networks,” *IEEE Access*, vol. 7, pp. 130647–130669, 2020.
- [43] A. A. Hidayat, K. Purwandari, T. W. Cenggoro, and B. Pardamean, “A Convolutional Neural Network-based Ancient Sundanese Character Classifier with Data Augmentation,” *Procedia Comput. Sci.*, vol. 179, no. 2020, pp. 195–201, 2021.
- [44] N. Shaffi and F. Hajamohideen, “UTHCD: A New Benchmarking for Tamil Handwritten OCR,” *IEEE Access*, vol. 9, pp. 101469–101493, 2021.
- [45] A. Trivedi, S. Srivastava, A. Mishra, A. Shukla, and R. Tiwari, “Hybrid evolutionary approach for Devanagari handwritten numeral recognition using Convolutional Neural Network,” *Procedia Comput. Sci.*, vol. 125, pp. 525–532, 2018.
- [46] M. K. Hayat, A. L. I. Daud, and A. A. Alshdadi, “Towards Deep Learning Prospects : Insights for Social Media Analytics,” *IEEE Access*, vol. 7, pp. 36958–36979, 2019.
- [47] S. H. Abdulhussain, B. M. Mahmmod, M. A. Naser, and M. Q. Abdulhasan, “A Robust Handwritten Numeral Recognition Using Hybrid Orthogonal Polynomials and Moments,” no. March, 2021.
- [48] H. Zheng, “applied sciences A Deep Convolutional Neural Network-Based Multi-Class Image Classification for Automatic Wafer Map Failure Recognition in Semiconductor Manufacturing,” 2021.
- [49] S. M. Shamim, M. Badrul, A. Miah, A. Sarker, M. Rana, and A. Al Jobair, “Indonesian Journal of Science & Technology Handwritten Digit Recognition Using Machine Learning Algorithms,” vol. 3, pp. 29–39, 2018.
- [50] D. Learning, “Automatic Estimation of Age Distributions from the First Ottoman Empire Population Register Series by Using,” 2021.