

Identifikasi Citra Tulisan Tangan Digital Menggunakan *Convolutional Neural Network* (CNN)

Nahila Khunafa Qudsi¹, Rosa Andrie Asmara², Arie Rachmad Syulistyo³

^{1,2,3} Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang

¹ nahilaqudsi@gmail.com, ² rosa.andrie@polinema.ac.id, ³ arie.rachmad.s@polinema.ac.id

Abstrak— *Handwriting recognition* adalah sebuah proses untuk mengenali tulisan tangan seseorang yang didapatkan dari sebuah citra digital kemudian sistem akan mengenali tulisan tersebut menggunakan metode klasifikasi. Adapun pemanfaatan dari pengenalan tulisan tangan ini adalah mengidentifikasi dokumen – dokumen penting, seperti : identifikasi tanda tangan dalam sebuah dokumen perjanjian / kontrak, pengenalan angka dan huruf dalam sebuah dokumen yang diharuskan dalam bentuk tulisan tangan. Namun dengan adanya dokumen yang ditulis tangan menyebabkan permasalahan dalam pemrosesan data tersebut. Seseorang harus menuliskan kembali data tersebut dalam bentuk digital dengan kata lain data tersebut diketikkan kedalam komputer. Hal ini tidak efisien dan membutuhkan waktu yang lama karena mengerjakan hal yang sama dua kali. Untuk mengatasi hal tersebut dapat digunakan sebuah sistem cerdas dimana komputer akan diatur sedemikian rupa agar dapat memiliki kemampuan seperti manusia. Metode atau algoritma yang digunakan dalam penelitian ini adalah *Convolutional Neural Network* (CNN). CNN memiliki 2 tahap yaitu *feature extraction layer* dan *fully connected layer*. Tahap pertama merupakan tahap untuk mengekstraksi citra menjadi beberapa *feature* berupa angka atau nilai yang merepresentasikan citra tersebut. Tahap kedua merupakan tahap pembelajaran data dan klasifikasi menggunakan *multilayer perceptron*. Sebelum proses *training*, dilakukan *resize image* yaitu mengubah ukuran citra dari 28 x 28 menjadi 32 x 32 *pixel*. Setelah itu, dilakukan proses ekstraksi citra dengan menggunakan konvolusi *layer* dan *average pooling layer* sebanyak dua kali. Selanjutnya dilakukan proses pembelajaran menggunakan *feed forward* dan *backpropagation* sehingga menghasilkan bobot dan bias dari masing – masing kelompok citra. Dataset yang digunakan dalam penelitian ini adalah database MNIST baik untuk proses *training* maupun *testing*. Selain menggunakan database MNIST, *testing* dilakukan terhadap tulisan tangan 20 orang koresponden yang menuliskan angka 0 hingga 9. Adapun hasil pengujian dari proses klasifikasi menggunakan CNN menunjukkan akurasi sebesar 98,6 % untuk dataset MNIST dan sebesar 88% untuk data koresponden. Hal ini dapat disimpulkan bahwa metode CNN mampu mengenali tulisan angka dengan baik.

Kata kunci—Convolutional Neural Network; citra, filter, tulisan tangan

I. PENDAHULUAN

Perkembangan teknologi yang pesat menyebabkan berkembangnya ilmu pengetahuan. Salah satu contoh keterkaitan antara teknologi dan ilmu pengetahuan adalah

pengenalan pola (*pattern recognition*). *Pattern* adalah entitas yang terdefinisi dan dapat diidentifikasi melalui ciri-cirinya dimana ciri – ciri tersebut digunakan untuk membedakan satu *pattern* dengan *pattern* lainnya. *Pattern recognition* adalah suatu teori, algoritma, sistem yang bertujuan menentukan kelompok atau kategori *pattern* berdasarkan ciri-ciri yang dimiliki oleh *pattern* tersebut. Dengan kata lain, *pattern recognition* membedakan suatu objek dengan objek lain [1].

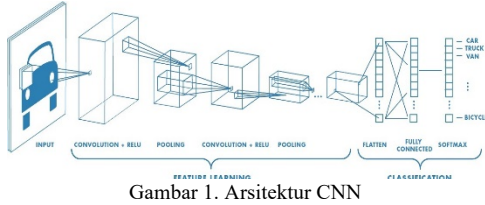
Salah satu implementasi dari *pattern recognition* adalah pengenalan tulisan tangan (*handwriting recognition*). *Handwriting recognition* adalah sebuah proses untuk mengenali tulisan tangan seseorang yang didapatkan dari sebuah citra digital kemudian sistem akan mengenali tulisan tersebut menggunakan metode klasifikasi [2]. Adapun pemanfaatan dari pengenalan tulisan tangan ini adalah mengidentifikasi dokumen – dokumen penting, seperti : identifikasi tanda tangan dalam sebuah dokumen perjanjian / kontrak, pengenalan angka dan huruf dalam sebuah dokumen yang diharuskan dalam bentuk tulisan tangan. Namun dengan adanya dokumen yang ditulis tangan menyebabkan permasalahan dalam pemrosesan data tersebut. Seseorang harus menuliskan kembali data tersebut dalam bentuk digital dengan kata lain data tersebut diketikkan kedalam komputer. Hal ini tidak efisien dan membutuhkan waktu yang lama karena mengerjakan hal yang sama dua kali [3]. Untuk mengatasi hal tersebut dapat digunakan sebuah sistem cerdas dimana komputer akan diatur sedemikian rupa agar dapat memiliki kemampuan seperti manusia.

Penelitian ini bertujuan untuk mengidentifikasi tulisan tangan berupa angka menggunakan metode CNN serta menguji kinerja CNN itu sendiri dalam mengidentifikasi citra tulisan tangan digital. CNN digunakan untuk mengenali dan mengklasifikasikan pola yang terdapat pada tulisan tangan. Dengan menggunakan metode CNN diharapkan menghasilkan hasil akurasi yang baik dengan tingkat *error* yang rendah. Dataset yang digunakan dalam penelitian ini adalah *database* MNIST baik untuk proses *training* maupun *testing*. Selain menggunakan *database* MNIST, pengujian dilakukan terhadap tulisan tangan 20 orang koresponden yang menuliskan angka 0 hingga 9.

II. TINJAUAN PUSTAKA

A. Convolutional Neural Network

Convolutional Neural Network merupakan salah satu metode yang ada pada deep learning. Konsepnya hampir sama dengan *Multi Layer Perceptron (MLP)* yaitu menggunakan algoritma backpropagation dalam klasifikasi datanya. Perbedaan CNN dan MLP adalah arsitektur dari CNN didahului dengan proses mengenali pola secara langsung dari piksel suatu gambar sehingga dapat meminimalkan *preprocessing*. Proses dari CNN secara garis besar terbagi menjadi 2 proses yaitu : *Feature Extraction Layer* dan *Fully Connected Layer*.



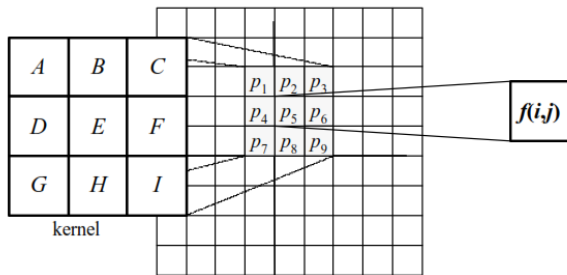
Gambar 1. Arsitektur CNN

1) Feature Extraction Layer

Feature Extraction Layer digunakan untuk mengekstraksi fitur yang ada pada sebuah gambar berupa angka-angka yang merepresentasikan gambar tersebut. Pada layer ini terdapat dua proses yaitu :

a. Konvolusi

Konvolusi citra adalah teknik untuk menghaluskan suatu citra atau memperjelas citra dengan menggantikan nilai piksel dengan sejumlah nilai piksel yang sesuai atau berdekatan dengan piksel aslinya [4]. Cara perhitungan konvolusi adalah mengalikan piksel gambar dengan filter yang telah ditentukan sehingga menghasilkan *output* berupa *feature map*.



Gambar 2. Ilustrasi konvolusi

$$f(i, j) = Ap_1 + Bp_2 + Cp_3 + Dp_4 + Ep_5 + Fp_6 + Gp_7 + Hp_8 + Ip_9 \quad (1)$$

Untuk menghitung ukuran *feature map* dilakukan dengan rumus sebagai berikut :

$$Output = \frac{W - N + 2P}{S} + 1 \quad (2)$$

Keterangan :

W = Panjang / Tinggi Input

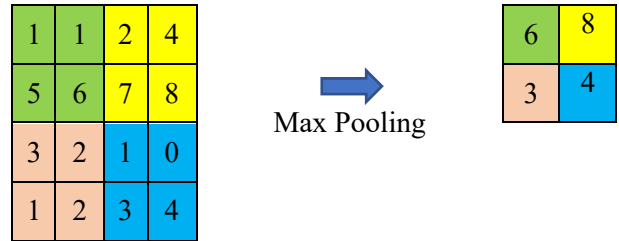
N = Panjang / Tinggi Filter

P = *Padding*

S = *Stride*

b. Pooling

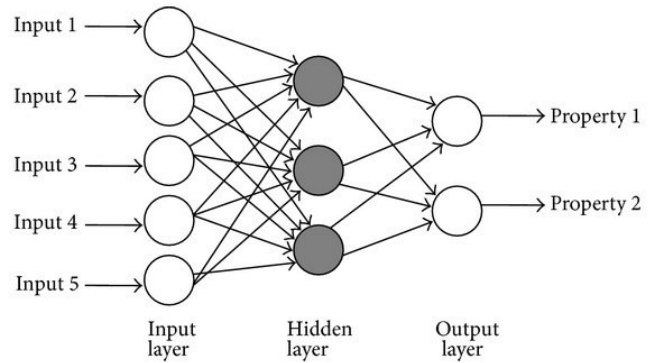
Pooling adalah mengurangi dimensi dari *feature map* sehingga mengurangi *overfitting* dan dapat mempercepat komputasi. Hal tersebut dikarenakan parameter yang harus diupdate jumlahnya semakin sedikit. *Pooling layer* yang biasa digunakan ada 2 macam yaitu *Max Pooling Layer* dan *Average Pooling Layer*. Sebagai contoh jika menggunakan *Max Pooling Layer* dengan ukuran 2x2 dengan stride 2, artinya setiap pergeseran nilai *feature map* diambil nilai maksimum dari area 2x2 piksel yang terpilih. Berbeda dengan *Max Pooling*, *Average Pooling* mengambil nilai rata-rata dari area piksel yang terpilih



Gambar 3. Max Pooling Layer

2) Fully Connected Layer

Feature map yang dihasilkan dari dari *feature extraction layer* masih berbentuk *multidimensional array*. Sehingga perlu dilakukan *flatten* atau *reshape feature map* menjadi sebuah vektor agar dapat digunakan sebagai input dari *fully connected layer* [5]. Pada *fully connected layer* memiliki cara kerja yang sama dengan MLP dan memiliki bagian yang sama yaitu : beberapa *input layer*, *hidden layer*, *activation function*, dan *output layer*. Layer ini bertujuan agar data dapat diklasifikasikan.

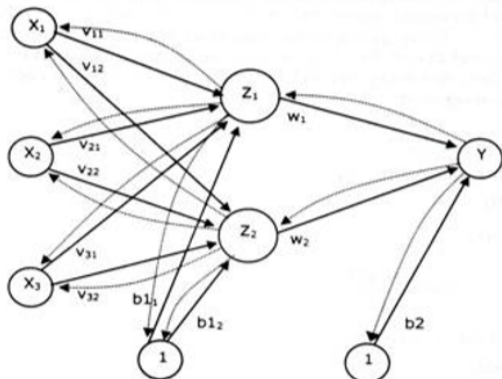


Gambar 4. Multi Layer Perceptron

Algoritma yang digunakan pada layer ini yaitu *feed-forward* dan *backpropagation*. Tahapan yang dilakukan adalah mengerjakan mengerjakan *feed-forward* terlebih dahulu, kemudian mencari nilai error, dilanjutkan mengubah bobot menggunakan *backpropagation*. Pada saat melakukan feed forward masing-masing neuron diaktifkan menggunakan fungsi aktivasi yang dapat diturunkan [6].

Gambar 5 menunjukkan *neural network* yang terdiri dari tiga unit *neuron* pada lapisan input(x1, x2, dan x3), dua *neuron* pada lapisan tersembunyi(Z1 dan Z2), dan 1 unit *neuron* pada lapisan output (Y). Bobot yang menghubungkan

x_1 , x_2 , dan x_3 dengan neuron pertama pada lapisan tersembunyi adalah V_{11} , V_{21} , dan V_{31} . b_{11} dan b_{12} adalah bobot bias yang menuju neuron pertama dan kedua pada lapisan tersembunyi. Bobot yang menghubungkan Z_1 dan Z_2 dengan neuron pada lapisan output adalah w_1 dan w_2 . Bobot bias b_2 menghubungkan lapisan tersembunyi dengan lapisan output.



Gambar 5. Ilustrasi backpropagatiion

B. Stride

Stride adalah parameter yang menentukan berapa jumlah pergeseran filter. Jika nilai *stride* adalah 1, maka *convolution filter* akan bergeser sebanyak 1 *pixels* secara *horizontal* lalu *vertical*. Semakin kecil *stride* maka akan semakin detail informasi yang didapatkan dari sebuah input, namun membutuhkan komputasi yang lebih jika dibandingkan dengan *stride* yang besar

C. Padding

Padding atau *Zero Padding* adalah parameter yang menentukan jumlah *pixels* (berisi nilai 0) yang akan ditambahkan di setiap sisi dari input. Hal ini digunakan dengan tujuan untuk memanipulasi dimensi *output* dari *convolution layer* (Feature Map).

D. Fungsi Aktivasi RELU

Fungsi aktivasi merupakan fungsi yang menggambarkan hubungan antara tingkat aktivitas internal (summation function) yang mungkin berbentuk linear ataupun non-linear. Fungsi ini bertujuan untuk menentukan apakah neuron diaktifkan atau tidak [7]. ReLu atau *Rectified Linear Unit* merupakan salah satu fungsi aktivasi yang biasanya digunakan pada *deep learning*. Cara kerja dari ReLu adalah mengubah nilai negative menjadi 0. ReLu hanya membuat pembatas pada bilangan nol [8], artinya apabila $x \leq 0$ maka $x = 0$ dan apabila $x > 0$ maka $x = x$. Penulisan rumus matematikanya adalah sebagai berikut :

$$f(x) = \max(0, x) \tag{3}$$

E. Softmax Classifier

Fungsi *softmax classifier* adalah menghitung semua probabilitas masing-masing kelas target terhadap kemungkinan seluruh kelas. Biasanya *softmax* digunakan untuk klasifikasi yang memiliki banyak kelas. Sehingga dengan fungsi ini

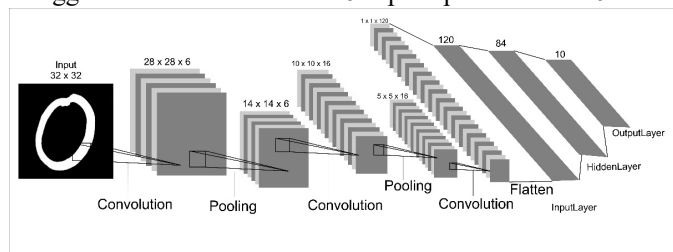
dapat menghitung probabilitas dari masing-masing kelas dan menentukan kelas target data yang diinputkan berdasarkan nilai probabilitas yang tertinggi. Rentang nilai probabilitas dari *softmax classifier* sendiri adalah 0 hingga 1 dengan jumlah keseluruhan nilai probabilitas = 1. Adapun rumus dari *softmax classifier* adalah sebagai berikut :

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \tag{4}$$

III. PERANCANGAN SISTEM

A. Arsitektur Sistem

Metode yang digunakan untuk mengembangkan aplikasi ini adalah *Convolutional Neural Network* dengan menggunakan arsitektur *LeNet-5* seperti pada Gambar 6.



Gambar 6.arsitektur sistem

Arsitektur *Lenet 5* memiliki 2 proses *convolutional* – *pooling* diikuti *convolutional layer* dan *flattening* kemudian *fully connected layer* dan menggunakan *softmax* sebagai *classifier*. Berikut ini adalah penjelasan dari Gambar 4 :

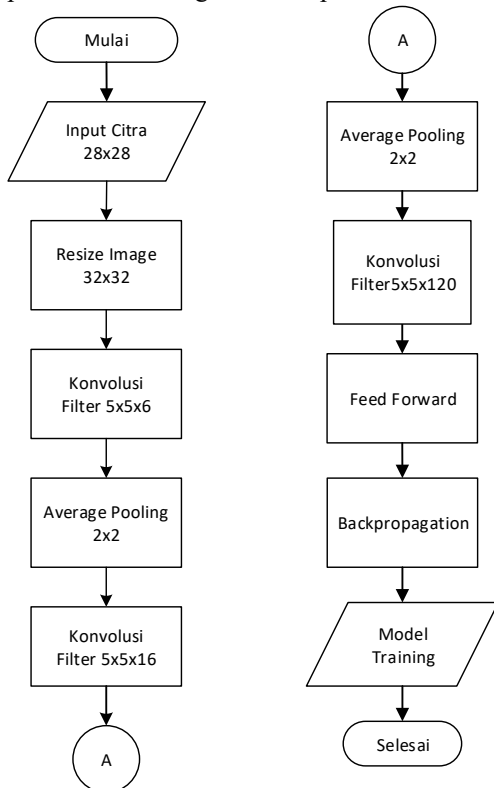
- Input merupakan citra *grayscale* ukuran 32 x 32 piksel
- Layer pertama adalah *Convolutional layer*. *Convolutional* terdiri dari 6 macam filter berukuran 5 x 5 sehingga dimensi citra yang awalnya 32 x 32 x 1 berubah menjadi 28 x 28 x 6. Filter yang digunakan adalah *gaussian filter* dimana nilai filternya adalah -1;0;1 yang didapatkan secara random oleh aplikasi. Selanjutnya diterapkan fungsi aktivasi *ReLU* terhadap hasil konvolusi
- Layer kedua adalah *sub-sampling layer* menggunakan *average pooling*. Ukuran filter yang digunakan adalah 2 x 2 dengan *stride* = 2. Dimensi citra yang dihasilkan yaitu 14 x 14 x 6
- Layer ketiga adalah *convolutional layer* dengan 16 macam filter berukuran 5 x 5. Sehingga didapatkan dimensi citra 10 x 10 x 16. Filter yang digunakan sama dengan layer pertama.
- Layer keempat merupakan *average pooling* kedua dengan filter 2x2 dan *stride* = 2. Dimensi citra dari layer ini adalah 5 x 5 x 16
- Layer kelima adalah *convolutional layer* dengan 120 macam filter berukuran 5 x 5. Sehingga didapatkan dimensi citra 1 x 1 x 120
- Flattening* merupakan konversi dari hasil matriks pada layer kelima menjadi matriks 1 dimensi
- Layer keenam adalah *fully connected layer* dengan 120 *nodes* sebagai *input layer*. Masing – masing *nodes*

tersebut terhubung dengan *hidden layer* sebanyak 84 *nodes*

- i. *Output layer* adalah hasil klasifikasi menggunakan *softmax* dengan 10 hasil kemungkinan yaitu 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

B. Proses Training

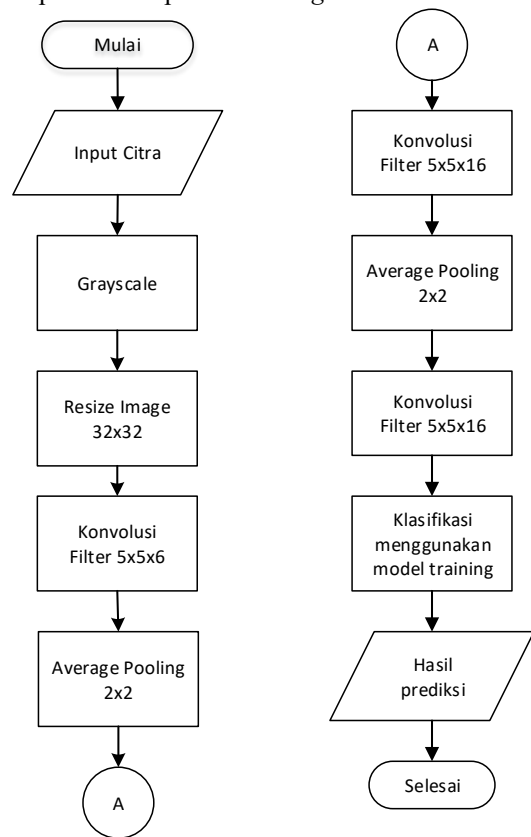
Proses *training* dilakukan untuk pengenalan bentuk – bentuk angka menggunakan data *training*. Pada proses *training*, pertama – tama citra inputan yang berukuran 28 x 28 piksel akan *resize* ke dalam ukuran 32 x 32 piksel. Pada layer pertama citra diolah dengan proses konvolusi dimana matriks citra akan dikalikan dengan matriks filter yang ukurannya sudah ditentukan terlebih dahulu. Adapun nilai dari matriks filter merupakan nilai acak. Kemudian pada layer kedua dilakukan proses *average pooling*. *Average pooling* didapatkan dari rata – rata dari suatu blok matriks. Proses konvolusi dan *average pooling* diulangi 2x sehingga mendapatkan 4 layer. Setelah itu dilanjutkan proses konvolusi dengan filter baru. Hasil konvolusi tersebut melalui proses *flattening* yaitu menjadikan matriks menjadi array 1 dimensi. Kemudian dilakukan proses *feed forward* dan *backpropagation*. Untuk mendapatkan nilai *weight* dan *bias* dari masing – masing angka 0 hingga 9 digunakan fungsi aktivasi *softmax* sejumlah output yaitu 10. Alur proses *training* pada sistem ini digambarkan pada Gambar 7.



Gambar 7. Proses training

C. Proses Testing

Proses *testing* merupakan proses klasifikasi menggunakan bobot dan bias dari hasil proses *training*. Perbedaan dari proses *training* dan *testing* terletak pada proses *backpropagation*. Pada proses *testing* tidak terdapat proses *backpropagation*. Untuk proses preprocessing nya sendiri, sama dengan *training* yaitu citra inputan berukuran 28 x 28 piksel *resize* menjadi 32 x 32 piksel. Setelah itu, citra hasil *resize* diubah kedalam *grayscale*. Pada layer pertama, citra *grayscale* dikonvolusi dengan *gaussian filter* yang didapatkan dari *random* berukuran 5 x 5. Selanjutnya dilakukan proses *pooling* menggunakan *average pooling* dengan ukuran 2 x 2. Hasil dari *average pooling* dijadikan inputan pada konvolusi kedua. Konvolusi kedua menggunakan *filter random* dengan ukuran 5 x 5 sebanyak 16 filter. Setelah itu dilakukan *average pooling* dengan ukuran 2 x 2. Kemudian dilakukan konvolusi lagi hingga menghasilkan *output* dengan ukuran 1x1x120. Proses selanjutnya yaitu *flattening* dilanjutkan dengan *fully connected layer* dimana pada proses ini dilakukan klasifikasi dengan menggunakan bobot dan bias yang didapatkan dari proses *training*.



Gambar 8. Proses testing

IV. HASIL PENGUJIAN DAN PEMBAHASAN

Dari proses perancangan yang telah dilakukan, maka sistem ini diimplementasikan kedalam sebuah aplikasi. Selanjutnya dilakukan pengujian akurasi. Pengujian akurasi ini dilakukan untuk mengetahui apakah sistem dapat mengidentifikasi citra tulisan tangan digital berdasarkan model training yang dilakukan sebelumnya. Adapun data yang

digunakan dalam pengujian ini menggunakan data MNIST sebanyak 10.000 citra testing ditambah dengan 200 citra yang diambil dari koresponden.

A. Pengaruh Jumlah Layer

Layer pada *feature extraction layer* terdapat 2 macam yaitu konvolusi dan pooling. Konvolusi bertujuan untuk mendapatkan fitur dari masing – masing citra berdasarkan filter yang telah ditentukan. Sedangkan *pooling* digunakan untuk mengurangi ukuran citra hasil konvolusi. Untuk mengetahui seberapa besar pengaruh layer terhadap hasil akurasi CNN maka dilakukan pengujian terhadap jumlah layer yang ditunjukkan pada Tabel 1.

TABEL 1. ARSITEKTUR PENGUJIAN JUMLAH LAYER

No.	Percobaan	Asitektur
1.	Percobaan – 1	Citra 32x32 → Konvolusi Filter 5x5x6 → Average Pooling 2x2 → Flattening → Fully Connected Layer
2.	Percobaan – 2	Citra 32x32 → Konvolusi Filter 5x5x6 → Average Pooling 2x2 → Konvolusi Filter 5x5x16 → Average Pooling 2x2 → Flattening → Fully Connected Layer
3.	Percobaan - 3	Citra 32x32 → Konvolusi Filter 5x5x6 → Average Pooling 2x2 → Konvolusi Filter 5x5x16 → Average Pooling 2x2 → Konvolusi Filter 5x5x120 → Flattening → Fully Connected Layer

Dari hasil percobaan menggunakan arsitektur pada Tabel 1 diketahui bahwa semakin banyak layer pada feature extraction layer maka semakin baik pula hasil yang didapatkan. Hasil percobaan pengujian terhadap jumlah layer ditunjukkan pada Tabel 2.

TABEL 2. HASIL PENGUJIAN TERHADAP JUMLAH LAYER

No.	Percobaan	MNIST	Koresponden	Rata - Rata
1.	Percobaan - 1	97.41%	79 %	88.2 %
2	Percobaan – 2	98 %	85 %	91.5 %
3	Percobaan - 3	98.6%	88 %	93.3 %

B. Pengujian Per Epoch

Epoch merupakan jumlah perulangan yang dilakukan pada *Neural Network*. Dalam arti lain, *epoch* adalah suatu kondisi saat seluruh dataset *training* melalui proses yang terdiri dari *feed forward* dan *backpropagation* sampai dikembalikan ke proses *feed forward* dalam satu kali putaran. Untuk mempercepat proses *training* dataset biasanya terbagi menjadi beberapa *batch*. Adapun jumlah *batch* yang digunakan dalam penelitian adalah 128, dimana seluruh

jumlah dataset akan dibagi menjadi 469 sehingga masing-masing batch terdiri dari 128 data. Hasil pengujian dengan mengubah parameter epoch akan dijelaskan pada Tabel 3.

TABEL 3. HASIL PENGUJIAN TERHADAP JUMLAH EPOCH

No.	Jumlah Epoch	Akurasi MNIST	Akurasi Koresponden	Rata-Rata
1.	10	98.05%	84.5%	91.27 %
2	30	98.6%	88%	93 %
3	50	98%	85%	91.8 %
Rata-rata		98.21 %	85.83 %	

Dari hasil percobaan pada Tabel 2 dan Tabel 3 maka percobaan selanjutnya menggunakan arsitektur pada percobaan-3 dan jumlah epoch sebanyak 30

C. Pengujian Terhadap Citra Original

Citra original terdapat 2 macam yaitu : database MNIST dan dataset koresponden. Dengan menggunakan model *training* sebelumnya didapatkan hasil pengujian sebagai berikut :



TABEL 4 HASIL PENGUJIAN CITRA ORIGINAL

No.	Dataset	Akurasi
1.	MNIST	98.6%
2.	Koresponden	88%

D. Pengujian Terhadap Citra Noise

Untuk mengetahui seberapa besar CNN dapat mengidentifikasi tulisan tangan digital, maka dari itu CNN diujikan kepada citra yang memiliki *noise*. Adapun *noise* yang digunakan adalah *noise gaussian* dan *noise salt-paper*. Berikut ini adalah contoh citra yang ditambahkan noise.

TABEL 5. CONTOH CITRA BER-NOISE

No	Noise	Citra
1.	Gaussian	
2.	Salt-Papper	

Citra yang digunakan dalam pengujian ini berjumlah 200 yang didapatkan dari citra koresponden. Adapun hasil pengujian disajikan dalam Tabel 6.

TABEL 6. HASIL PENGUJIAN CITRA BER-NOISE





No.	Citra	Akurasi
1.	Gaussian	76.5 %
2.	Salt-Paper	72.5 %
Rata-Rata		74.5 %

E. Pengujian Terhadap Citra Rotation

Pengujian selanjutnya dilakukan rotasi terhadap 200 citra koresponden. Rotasi yang digunakan adalah 15°, 30°, 45°, dan 90°

TABEL 7. CONTOH CITRA ROTATION

No	Rotasi	Citra
----	--------	-------

1.	Rotasi 15°	
2.	Rotasi 30°	
3.	Rotasi 45°	
4.	Rotasi 90°	

TABEL 8. HASIL PENGUJIAN CITRA BER-NOISE

No.	Citra	Akurasi
1.	Rotation 15°	82.5 %
2.	Rotation 30°	63.5 %
3.	Rotation 45°	36,5 %
4.	Rotation 90°	25 %

V. KESIMPULAN

Sistem identifikasi citra tulisan tangan digital menggunakan *convolutional neural network* (CNN) telah diimplementasikan menggunakan bahasa pemrograman python. Berdasarkan pengujian yang telah dilakukan dapat ditarik kesimpulan sebagai berikut :

- Hasil prediksi dari sistem identifikasi citra tulisan tangan digital menggunakan CNN dipengaruhi oleh beberapa factor yaitu : letak tulisan, besar kecilnya tulisan, tebal tipisnya tulisan, dan dataset yang digunakan untuk training.
- Hasil akurasi citra MNIST lebih baik daripada citra koresponden yaitu sebesar 98.21 % untuk citra MNIST dan 85.83% untuk citra koresponden.
- Model CNN masih bisa mengenali tulisan tangan digital angka meskipun memiliki noise dengan tingkat akurasi sebesar 74.5%
- CNN mampu mengenali citra yang mengalami rotasi sebesar 15° dengan tingkat akurasi sebesar 82.5% dan rotasi sebesar 30° dengan tingkat akurasi sebesar 63.5%. Namun kurang baik dalam mengenali citra yang mengalami rotasi 45° hingga 90° dengan akurasi dibawah 50%.

daftar pustaka

- [1] R. Munir, *Pengolahan Citra Digital*, Bandung: Penerbit Informatika, 2006.
- [2] I. S. S. Made Edwin Wira Putra, "Structural off-line handwriting character recognition using approximate subgraph matching and levenshtein distance," *Procedia Computer Science* 59, p. 340 – 349, 2015.
- [3] Morwati, "Pengenalan Citra Huruf Alphabet Tulisan Tangan Menggunakan Naive Bayes Classifier," Universitas Maulana Malik Ibrahim, 2014.
- [4] A. Y. W. R. S. I Wayan Suartika E. P, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101," *Jurnal Teknik ITS*, 2016.
- [5] S. Sena, "Medium," 13 November 2017. [Online]. Available: [https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-](https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94)

b003b477dc94. [Accessed 19 November 2018].

- [6] Q. Lina, "mc.ai," 2 Januari 2019. [Online]. Available: <https://mc.ai/apa-itu-convolutional-neural-network/>. [Accessed 31 Juli 2019].
- [7] T. Nurhikmat, "Implementasi Deep Learning Untuk Image Classification Menggunakan Algoritma Convolutional Neural Network(CNN) Pada Citra Wayang Golek," Yogyakarta, 2018.
- [8] P. H. K., "Medium," 16 Mei 2017. [Online]. Available: <https://medium.com/@opam22/menilik-activation-functions-7710177a54c9>. [Accessed 15 April 2019].