

# Pengenalan Simbol Matematika dengan Metode Convolutional Neural Network (CNN)

Alvin Kusuma Putra<sup>1</sup>, Hendra Bunyamin<sup>2</sup>

Program Studi Teknik Informatika, Universitas Kristen Maranatha  
Jl. Prof. drg. Surya Sumantri No. 65 Bandung

<sup>1</sup>alvinkusuma98@gmail.com

<sup>2</sup>hendra.bunyamin@it.maranatha.edu

**Abstract** — Handwriting recognition of mathematical symbol has been a problem in the field of pattern recognition that makes more difficult to detect than the one consisting only handwritings. Specifically, the complex structure of writing and diverse mathematical symbols exacerbates the segmentation, symbol recognition, and structure analysis of handwriting of mathematical expressions. Therefore, the process of converting handwritten mathematical expressions into digital text format such as LaTeX or MathML is desperately needed. This article focuses on the process of handwriting mathematical symbol recognition. Particularly, we compare two state-of-the-art Convolutional Neural Networks (CNN) that are ResNet34 and DenseNet121. This paper also explains how the deep learning approaches, such as CNN that can distinguish an object in an image; additionally, we learn that DenseNet121 architecture delivers better accuracies than the ResNet34 architecture.

**Keywords**— Machine Learning, Deep Learning, Convolutional Neural Network, FastAI.

## I. PENDAHULUAN

Pertumbuhan pengguna Internet dalam tahun-tahun terakhir ini begitu pesat dan meningkatnya pengguna internet maka semakin banyak juga sumber ilmu pengetahuan yang dibuat dalam bentuk teks digital yang dapat diakses oleh umum seperti *academic paper* atau jurnal ilmiah, dan pada teks digital tersebut seringkali terdapat ekspresi matematika didalamnya dan mmebutuhkan notasi khusus berbentuk *format markup language* seperti LaTeX dan MathML, pemahaman mengenai notasi tersebut sangat dibutuhkan, oleh sebab itu untuk mempermudah penulisannya dibutuhkan sistem yang dapat merubah tulisan tangan ekspresi matematika ke dalam bentuk *format markup language*. Penelitian ini hendak membantu pengguna mengenali simbol matematika yang merupakan tulisan tangan secara otomatis. Lebih spesifik lagi, penelitian ini hendak mengimplementasikan metode *deep learning* untuk dapat mengenal simbol matematika. Adapun metode yang digunakan adalah *Convolutional Neural Network (CNN)* [1].

Tujuan dari penelitian ini adalah membandingkan performa dari model *Convolutional Neural Network* sehingga didapatkan hasil dalam bentuk akurasi dan dapat mengenali simbol matematika yang kompleks.

Adapun manfaat yang didapat seperti memperkaya pustaka mengenai penerapan metode *Convolutional Neural Network* untuk pengenalan tulisan tangan simbol matematika, serta hasil penelitian ini diharapkan menjadi pemicu peneliti lainnya untuk dapat mengembangkan dan bisa jadi pembanding dengan metode lainnya.

## II. KAJIAN TEORI

Teori yang dijelaskan meliputi: *Artificial Intelligence (AI)*, *Deep Learning*, *Convolutional Neural Network (CNN)*, *Convolutional Layer*, *Pooling Layer*, Fungsi Aktivasi, *Fully-Connected Layer*, *Residual Neural Network (Resnet)*, dan *Densely Neural Network (Densenet)*.

### A. Artificial Intelligence (AI)

AI atau *Artificial Intelligence* adalah sebuah bidang ilmu yang digunakan untuk membuat hidup manusia lebih baik dari masa ke masa. Upaya ini dilakukan dengan memberikan kecerdasan pada mesin supaya dapat berpikir seolah-olah seperti manusia. Menurut *McKinsey Global Institute*, nilai ekonomi yang akan dihasilkan AI pada tahun 2030 adalah sekitar \$13 trillion atau setara  $18 \times 10^{15}$  rupiah. Pada Gambar 1 menjelaskan bidang-bidang apa saja yang memberikan nilai ekonomis tersebut [2].



Gambar 1. Nilai uang yang akan dihasilkan AI pada tahun 2030 [2]

Secara umum AI dapat dibagi menjadi 2 (dua) pemahaman yaitu *Artificial Narrow Intelligence (ANI)* dan *Artificial General Intelligence (AGI)*.

1) *Artificial Narrow Intelligence (ANI)*: *Artificial Narrow Intelligence* adalah salah satu konsep dari *Artificial Intelligence (AI)*. Jenis kecerdasan buatan ini adalah berfokus pada satu tugas yang spesifik, dengan jangkauan yang terbatas juga, contohnya seperti *smart speaker*, *self-driving car*, *web search*, dan *AI Application* dalam pertanian atau pabrik. Jenis AI ini merupakan suatu trik sederhana tetapi ketika menemukan trik yang sesuai, sehingga trik ini dapat sangat berharga dan merupakan konsep dasar dari AI.

2) *Artificial General Intelligence (AGI)*: *Artificial General Intelligence* adalah salah satu konsep dari *Artificial Intelligence (AI)*. Konsep dari jenis AI ini adalah dapat melakukan sesuatu yang dapat manusia lakukan atau bahkan lebih pintar dan dapat melakukan lebih banyak hal. Hal ini masih terlampaui sulit untuk direalisasikan pada saat ini.

### B. Deep Learning

*Deep Learning* adalah salah satu metode pembelajaran mesin yang memungkinkan komputer untuk mempelajari tugas-tugas yang disesuaikan dengan sifat manusia. Ini adalah teknologi yang mendukung perkembangan kecerdasan buatan atau *Artificial Intelligence (AI)* dan kemajuannya telah memberikan kemajuan diberbagai bidang. *Deep Learning* juga merupakan kunci dalam kendaraan yang memiliki sistem kemudi otomatis dimana teknologi tersebut sedang dikembangkan dalam beberapa tahun terakhir ini. Teknologi *Deep Learning* ini mampu mengenali tanda-tanda berhenti serta mampu membedakan antara tiang listrik dan manusia. Selain itu juga, teknik lainnya adalah memainkan peran penting dalam pengenalan suara pada ponsel, tablet, televisi, dan sebagainya. Ada alasan mengapa *Deep Learning* begitu menarik perhatian dalam beberapa tahun terakhir ini, karena teknologi ini telah mampu mencapai tingkat kinerja yang tidak mungkin dicapai dengan teknologi konvensional.

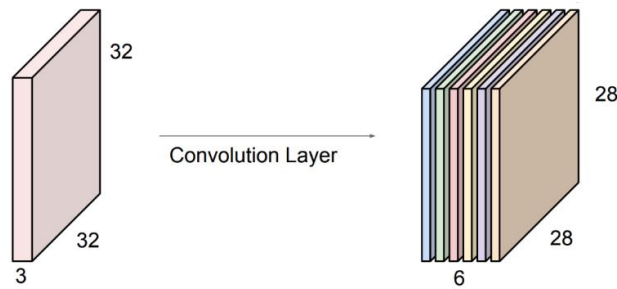
### C. Convolutional Neural Network (CNN)

*Convolutional Neural Network* atau sering disebut CNN adalah salah satu metode *machine learning* dari pengembangan *Multi-Layer Perceptron (MLP)* untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena memiliki banyak tingkat jaringan dan banyak diimplementasikan dalam data citra. Dalam kasus klasifikasi citra, MLP kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik.

CNN pertama kali dikembangkan dengan nama *NeoCognitron* oleh Kunihiko Fukushima [3], seorang peneliti dari NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Jepang. Konsep tersebut kemudian dimatangkan oleh Laboratories di Holmdel, New Jersey, USA [4]. Model CNN dengan nama LeNet berhasil diterapkan oleh LeCun pada penelitiannya mengenai pengenalan angka dan tulisan tangan. Pada tahun 2012, Alex Krizhevsky dengan penerapan CNN miliknya berhasil menjuarai kompetisi *ImageNet Large Scale Visual Recognition Challenge 2012* [5]. Prestasi tersebut menjadi momen pembuktian bahwa metode *Deep Learning*, khususnya CNN. Metode CNN terbukti berhasil mengungguli metode *Machine Learning* lainnya seperti SVM pada kasus klasifikasi objek pada citra.

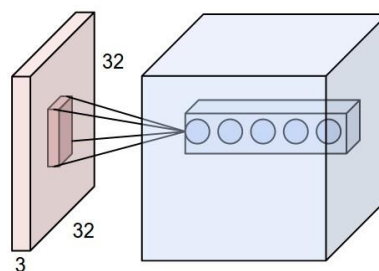
### D. Convolutional Layer

*Convolutional Layer* merupakan blok bangunan inti CNN tempat sebagian besar komputasi dilakukan pada lapis ini. Misalkan, sebuah *convolutional layer* dibangun dengan satu lembaran neuron yang berisi 28x28. Masing-masing terhubung dengan suatu area kecil dalam (citra) masukan, misalnya 5x5 (pixel) yang merupakan bidang reseptif untuk setiap *neuron* dan menyatakan bahwa filter yang digunakan berukuran 5x5, seperti pada Gambar 2. Seluruh bidang reseptif akan ditelusuri secara tumpang tindih parsial, maka semua neuron tersebut pasti berbagai bobot koneksi [6].



Gambar 2. Convolutional Layer dengan satu buah filter 5x5 [6]

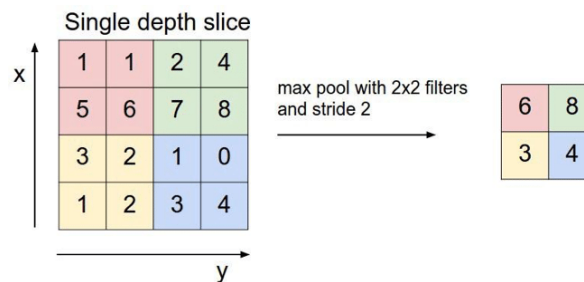
Convolutional layer dalam arsitektur CNN umumnya menggunakan lebih dari satu filter. Jika menggunakan empat filter, maka convolutional layer akan berisi sejumlah neuron yang tersusun dalam kisi berukuran 28x28x4. Selanjutnya, akan ada empat neurons yang melihat area yang sama pada (citra) masukan tersebut, seperti pada Gambar 3. Dengan cara ini, overfitting dapat dikontrol secara mudah dan setiap filter akan berusaha mencocokkan sebuah fitur tunggal pada semua posisi spasial.



Gambar 3. Convolutional Layer dengan empat filters 5x5 [6]

### E. Pooling Layer

Layer ini ada setelah convolutional layer. Layer ini terdiri dari sebuah filter dengan ukuran dan stride tertentu yang akan bergeser pada seluruh activation map. Pooling yang digunakan adalah max dan average pooling. Contohnya, dengan menggunakan max pooling 2x2 dan stride 2, di setiap pergeseran filter, nilai maksimum pada area 2x2 pixel tersebut yang akan dipilih. Sedangkan average pooling akan memilih nilai rata-ratanya [6]. Pooling layer digunakan untuk mempercepat komputasi karena parameter yang harus di-update semakin sedikit dan mengatasi overfitting.



Gambar 4. Pooling Layer

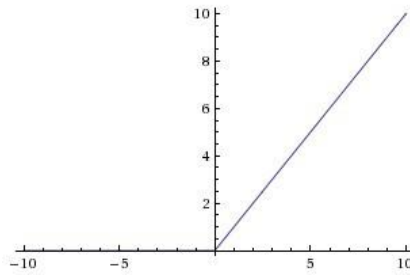
### F. Fungsi Aktivasi

Fungsi aktivasi merupakan fungsi yang digunakan pada jaringan saraf untuk mengaktifkan atau tidak mengaktifkan neuron. Karakteristik yang harus dimiliki oleh fungsi aktivasi jaringan perambatan balik antara lain harus kontinyu, terdiferensialkan, dan tidak menurun secara monotonis (monotonically non-decreasing) [7].

Pada penelitian ini fungsi aktivasi yang digunakan adalah Rectified Linear Unit (ReLU), Rectified Linear Unit (ReLU) adalah sebuah fungsi aktivasi yang memiliki perhitungan sederhana. Proses forward dan backward melalui ReLU hanya menggunakan kondisi if. Jika elemen bernilai negatif maka nilainya akan diset menjadi 0, tidak ada operasi eksponensial, perkalian atau pembagian. Dengan karakteristik ini, ReLU memiliki kelebihan yang akan muncul saat berhadapan dengan

jaringan yang memiliki neuron yang banyak sehingga dapat mengurangi waktu training dan testing dengan signifikan. Adapun persamaan dari fungsi aktivasi ReLU ini adalah sebagai berikut:

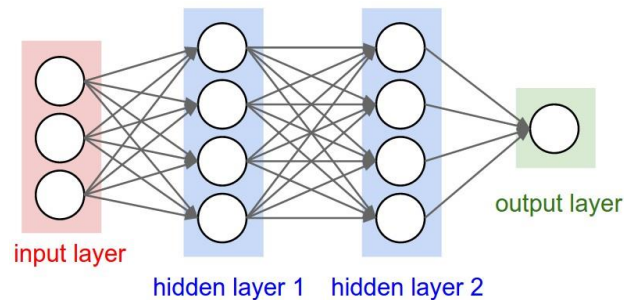
$$F(x) = \max(0, x)$$



Gambar 5. Grafik *Rectified Linear Unit* [7]

### G. *Fully Connected Layer*

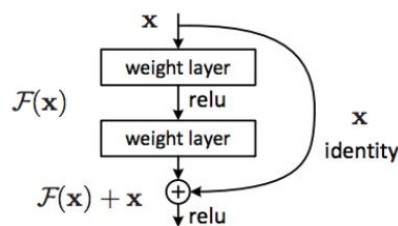
Pada lapisan yang terhubung secara penuh (*fully connected layer*), setiap *neurons* memiliki koneksi penuh ke semua aktivasi dalam lapisan sebelumnya. Hal ini sama persis dengan yang ada pada MLP. Model aktivasinya pun sama persis dengan MLP, yaitu komputasi menggunakan suatu perkalian matriks yang diikuti dengan bias *offset*. [9]



Gambar 6. *Multi Layer Perception* dengan 2 *hidden layer* [9]

### H. *Residual Neural Network (Resnet)*

*Residual Neural Network* atau yang biasa disebut sebagai Resnet adalah salah satu jenis arsitektur yang cukup populer; arsitektur ini dibuat oleh Kaiming He et al [10]. Arsitektur CNN yang memiliki kedalaman tinggi adalah salah satu hal penting dalam membangun model CNN yang memiliki performa yang baik, tetapi model CNN yang memiliki kedalaman yang tinggi juga memiliki masalah, yaitu *vanishing gradient problem*, yaitu suatu keadaan dengan hasil gradien yang dipelajari oleh model, tidak dapat mencapai layer pertama karena gradien mengalami perkalian berkali-kali sehingga layer pertama tidak menerima gradien apapun, atau secara singkatnya, hal ini menyebabkan suatu CNN tidak dapat belajar dari *error* yang telah dikalkulasi. Resnet memiliki berbagai macam jenis arsitektur, mulai dari 18, 34, 50, 101, sampai 152 layer. Pada penelitian ini akan digunakan arsitektur Resnet34, yaitu arsitektur resnet yang memiliki 34 layer, arsitektur ini dipilih karena arsitektur ini memiliki performa yang baik pada kompetisi ILSVRC [10].

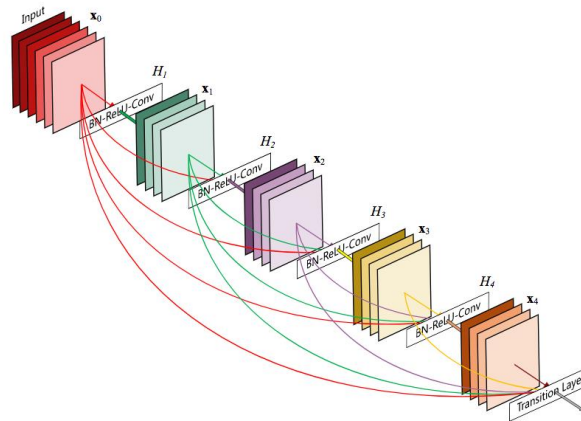


Gambar 7. Blok residual yang menjadi fondasi dari arsitektur ResNet [10]

### I. Densely Neural Network (Densenet)

Densely Neural Network atau yang biasa disebut dengan DenseNet merupakan arsitektur yang menghubungkan dari setiap layer ke layer lainnya dengan cara *feed-forward*. Penelitian sudah menunjukkan bahwa tingkat akurasi yang tinggi juga dipengaruhi oleh kedalaman suatu layer, dan training yang efisien akan tercapai jika CNN memiliki koneksi yang lebih dekat antara layer dengan input dan output. DenseNet memiliki beberapa kelebihan, yaitu: mengurangi *vanishing-gradient problem*, memperkuat *feature propagation*, dan mengurangi jumlah parameter (Huang, Liu, Weinberger, & van der Maaten, 2016) [11].

DenseNet terdiri dari blok Dense seperti pada Gambar 8. Di dalam blok-blok itu terdapat layer yang saling terhubung satu sama lain dan setiap layer mendapat input dari layer sebelumnya. Pada penelitian ini akan digunakan arsitektur DenseNet121 yaitu arsitektur dengan 121 layer.



Gambar 8. Blok diagram DenseNet

## III. METODOLOGI PENELITIAN

### A. Tahapan Penelitian

Pada penelitian ini akan dilakukan dengan beberapa tahap. Tahapan penelitian ini antara lain adalah pengkondisian dataset, studi literatur, rancangan metode, pengujian, memprediksi pada *dataset test*, melakukan analisis dan pembahasan, dan terakhir penarikan kesimpulan.

1) *Pengkondisian Dataset*: Pengkondisian dataset dilakukan untuk mempersiapkan semua data yang dibutuhkan pada sistem pengenalan simbol matematika. Dataset digunakan sebagai masukan yang kemudian akan diproses pada tahap selanjutnya. Pada penelitian ini dataset yang digunakan adalah dataset dari kaggle. Dataset tersebut berisikan alfabet Greek seperti: alpha, beta, gamma, mu, sigma, phi, dan theta, termasuk juga *alphanumeric*, semua operator matematika, *basic math function* seperti: log, lim, cos, sin, tan, dan juga simbol matematika seperti: \int, \sum, \sqrt, \Delta dan sebagainya. Selanjutnya semua simbol ini akan disebut kelas. Pada dataset ini terdapat gambar simbol matematika dengan rata-rata awal 810 gambar untuk setiap kelasnya. Namun ada perbedaan yang signifikan pada jumlah gambar tersebut (*imbalanced dataset*). Untuk mengatasi permasalahan *imbalanced dataset* ini jumlah gambar akan dikenakan proses *oversampling* dan *undersampling*. Setelah dilakukan balancing data rata-rata menjadi 486 gambar. Rencananya untuk 80% dari setiap kelas akan menjadi dataset train, 20% menjadi dataset validation dan 10% dari dataset validation menjadi dataset test.

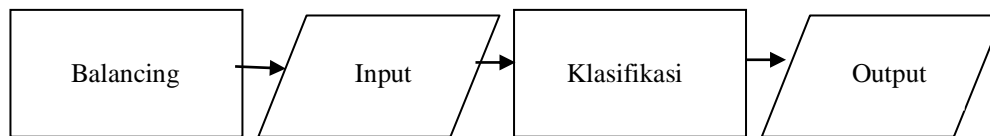
2) *Oversampling*: Dalam proses *oversampling* ini, akan dilakukan duplikasi data dengan melakukan transformasi gambar. Misalnya, menambahkan transformasi gambar dari gambar asli dengan merubah *brightness*, *contrast*, *tilt*, *jitter*, *skew*, *squish* dan sebagainya.

3) *Undersampling*: Dalam proses *undersampling* ini, akan dilakukan teknik *random under-sampling*. Teknik pengambilan sampel yang sederhana dengan mengambil sampel dalam kelas utama secara manual dengan cara membuang sebagian data agar seimbang dengan data lainnya.

4) *Studi Literatur*: Studi literatur dilakukan dengan cara mencari referensi yang bersumber dari buku, jurnal, dan artikel yang memiliki keterkaitan dengan permasalahan yang telah direncanakan untuk diteliti untuk mencari solusinya. Hal yang berhubungan dengan penelitian ini adalah, tentang tulisan tangan dan juga suatu metode *deep learning* yaitu *CNN* (*Convolutional Neural Networks*) dengan model ResNet34 dan model DenseNet121. Setelah dilakukannya studi literatur ini, maka akan didapatkan suatu rumusan langkah yang harus dikerjakan.

5) *Rancangan Metode*: Pada tahapan ini akan dilakukan perancangan arsitektur metode CNN yang akan digunakan. Perancangan ini meliputi penentuan model arsitektur yang akan digunakan.

6) *Pengujian*: Pada tahapan ini pengujian akan dilakukan dengan alur seperti Gambar 9.



Gambar 9. Alur Pengujian

Tahapan pertama, sebelum input diproses dilakukan proses oversampling dan undersampling agar data seimbang, lalu input diproses dalam klasifikasi dataset dengan pembagian dataset 80% untuk dataset train, 20% untuk dataset valid, dan 10% dari dataset valid sebagai dataset test. Kemudian dilakukan proses training dan validation dataset, dan akan menghasilkan output berupa akurasi, *training loss*, dan *validation loss*.

7) *Predict Dataset Test*: Pada tahap ini kedua model akan diuji coba untuk memprediksi gambar pada dataset test setelah kedua model tersebut selesai melakukan proses *training*.

8) *Analisis dan Pembahasan*: Pada tahap ini akan dilakukan analisis dan evaluasi untuk menguji performa dari model ResNet34 dan DenseNet121. Untuk mengetahui performa dari model CNN dapat dievaluasi dengan menghitung akurasinya dan tingkat lossnya.

9) *Penarikan Kesimpulan*: Pada bagian ini akan dibuat suatu kesimpulan yang berasal dari hasil analisis dan pembahasan dari data yang sudah diuji menurut rumusan masalah. Dengan cara ini bisa ditarik kesimpulan dalam bentuk akurasi.

#### B. Alat Penelitian

Platform yang digunakan adalah platform dari Google, yaitu *Google Colaboratory* dengan pengaturan *runtime GPU* aktif. Selama menggunakan *Google Colaboratory*, dibutuhkan koneksi yang stabil agar saat menjalankan proses *training* dapat berjalan dengan lancar. Media penyimpanan dataset yang digunakan adalah *Google Drive*, karena untuk mengolah data di *Google Colaboratory* diperlukan data yang diimport dari *Google Drive*, dan bahasa pemrograman yang digunakan pada penelitian ini adalah bahasa pemrograman Python dengan *library fastai*.

## IV. IMPLEMENTASI

### A. Proses Training dan Validating

Proses train dan validasi dilakukan pada kedua model yang akan diuji, yaitu ResNet34 dan DenseNet121. Kedua model tersebut akan dilakukan proses train dan validasi terlebih dahulu sebelum proses testing dilakukan, dan kedua model tersebut juga akan dihitung keakuratannya. Kumpulan gambar yang akan digunakan yaitu 82 simbol matematika dengan total gambar untuk training adalah 31893 gambar dan 7162 gambar untuk proses validasi. Tabel 1 menunjukkan hasil proses *training* dan *validating*, dan terlihat bahwa keakuratan model saat train dan validasi, DenseNet121 lebih baik dibandingkan dengan ResNet34.

TABEL I  
HASIL TINGKAT AKURASI TRAIN DAN VALIDATION MODEL

Training Epochs		ResNet34	DenseNet121
25	Training Loss	28.70%	15.28%
	Validation Loss	25.06%	23.82%
	Accuracy	93.35%	94.44%
50	Training Loss	12.17%	5.92%
	Validation Loss	19.48%	21.51%
	Accuracy	95.33%	96.07%
75	Training Loss	5.83%	1.58%
	Validation Loss	19.34%	22.30%
	Accuracy	95.90%	96.73%
100	Training Loss	4.73%	1.10%
	Validation Loss	19.28%	24.40%
	Accuracy	95.99%	96.64%

B. Proses Testing

Proses testing atau pengujian dilakukan pada kedua model yang sudah melakukan proses train dan validasi. Kedua model tersebut akan diuji terhadap dataset test yang berisikan 82 simbol matematika dengan total 833 gambar, dan akan dihitung keakuratan prediksinya. Tabel 2 menunjukkan hasil dari pengujian tingkat akurasi dari tiap model, dan terlihat bahwa keakuratan prediksi dari DenseNet121 lebih baik dibandingkan dengan ResNet34.

TABEL II  
HASIL TINGKAT AKURASI PREDIKSI MODEL

Model	Correct Prediction	Accuracy
ResNet34	800/833 Gambar	96.03%
DenseNet121	803/833 Gambar	96.39%

C. Error Analysis

Proses ini merupakan proses untuk menyelidiki kembali hasil dari prediksi salah yang dilakukan oleh model ResNet34 dan DenseNet121 untuk mengetahui hal-hal yang dapat memungkinkan terjadinya *error*.

TABEL III  
BEBERAPA SAMPEL SIMBOL ANALISIS

Simbol	ResNet34	DenseNet121
12 Simbol Tambah (+)	['+', 'sqrt', '+', '+', '+', '+', '+', 'T', '+', '+', '+', '+']	['+', 'sqrt', '+', '+', '+', '+', '+', '+', '+', '+', '+', '+']
<b>Analisis</b>	Pada tulisan tangan simbol kedua pada dataset memang terlihat seperti simbol akar dan symbol T yang diprediksi oleh ResNet34 tidak terlihat mirip simbol T	
11 Simbol Koma (,)	[',', 'forward_slash', ',', ',', ',', ',', ',', 'forward_slash', 'forward_slash', 'forward_slash', ',']	[',', 'forward_slash', ',', ',', ',', ',', ',', 'forward_slash', ',']
<b>Analisis</b>	Pada tulisan tangan simbol koma ini pada model ResNet34 banyak memprediksi bahwa itu forward slash atau garis miring, terlihat juga simbolnya mirip dengan garis miring.	

Pada Tabel III menunjukan beberapa dari sampel 82 simbol matematika yang diuji dan menunjukan bahwa hasil dari prediksi pada model DenseNet121 lebih baik dibandingkan dengan model ResNet34.

## V. SIMPULAN DAN SARAN

Berdasarkan hasil yang didapatkan penelitian ini berhasil untuk mendeteksi simbol matematika dari input tulisan tangan, dan dilihat hasilnya dari kedua model FastAI yang diujikan (ResNet34 dan DenseNet121), model DenseNet121 memiliki tingkat akurasi *saat training*, *validation* dan *test* lebih baik dibandingkan dengan model ResNet34.

Adapun saran yang dapat dikembangkan pada penelitian selanjutnya, seperti menambah model yang akan diuji, menggunakan GPU dengan kualitas baik agar saat proses training berjalan cepat dan lancar, dan menggunakan dataset dengan jumlah data yang lebih banyak lagi.

## DAFTAR PUSTAKA

- [1] D.-Y. Y. Kam-Fai Chan, "Mathematical Expression Recognition a Survey," *International Journal on Document Analysis and Recognition*, pp. 3-15, 2000.
- [2] A. Ng, "AI For Everyone," 2019. [Online]. Available: <https://www.coursera.org/learn/ai-for-everyone/home/welcome>. [Diakses 28 11 2019].
- [3] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, pp. 193-202, April 1980.
- [4] Y. LeCun, "Handwritten Digit Recognition with a BackPropagation Network," 1990.
- [5] H. Alex Krizhevsky, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, pp. 1097-1105, 2012.
- [6] Suyanto, "*Machine Learning Tingkat Dasar dan Lanjut*", Bandung: Informatika, 2008.
- [7] Z. Julpan, "Analisis Fungsi Aktivasi Sigmoid Biner dan Sigmoid," *Jurnal Teknovasi*, Vol. %1 dari %2Volume 02, Nomor 1,, p. 103 – 116, 2015.
- [8] A. Zisserman dan Karen Simonyan, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Computer Vision and Pattern Recognition*, 2015.
- [9] Wayan Suartika E. P, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101," *JURNAL TEKNIK ITS*, Vol. %1 dari %2Vol. 5, No. 1, pp. 65-69, 2016.
- [10] K. He, X. Zhang, S. Ren dan J. Sun, "Deep Residual Learning For Image Recognition," *arXiv:1512.03385 [cs]*, p. arXiv: 1512.03385, Dec 2015.
- [11] W. Gao Huang, "Densely Connected Convolutional Networks," pp. 4700-4708, 2009.