Pengenalan Pola Karakter Bahasa Jepang Hiragana Menggunakan 2D Convolutional Neural Network

Mellysa Margarita Susilo[1], Daniel Martomanggolo Wonohadidjojo[2], Nehemia Sugianto[3]

Abstrak— **Komik disukai oleh banyak orang di seluruh dunia. Komik Jepang atau yang biasa disebut *manga* dapat ditemukan di internet, tapi tidak semua orang dapat membacanya karena berbahasa Jepang. Terdapat tim non-profit yang bersedia menerjemahkan *manga* namun karena waktu dan tenaga yang terbatas maka tidak semua *manga* dapat diterjemahkan. Oleh sebab itu, walaupun terdapat banyak manga di internet, tidak semua orang dapat membacanya. Sehingga menimbulkan kekecewaan pada pembaca yang menantikannya. Masalah ini dapat diselesaikan dengan menggunakan pattern recognition dan algoritma convolutional neural network. Pada penelitian ini akan menggunakan media gambar komik. Langkah pertama yang dilakukan adalah mengambil tulisan Jepang dari balon teks pada manga. Setelah tulisan Jepang didapatkan, dilakukan feature extraction untuk dianalisa dan disimpan polanya untuk dibandingkan dengan bank data tulisan yang telah dipersiapkan sehingga dapat ditentukan karakter yang terdapat pada sumber gambar dikenali sebagai apa. Untuk melatih agar pattern recognition menghasilkan hasil yang maksimal, dikembangkanlah dengan algoritma convolutional neural network agar aplikasi dapat berjalan secara mandiri dan semakin pintar dalam mengenali pola. Akan terdapat dua aplikasi, aplikasi pertama adalah Element Extractor from Manga (Japanese Comic) yang dikembangkan dengan bahasa pemograman C# dan library AForge. Aplikasi kedua adalah program pattern recognition yang dikembangkan dengan MATLAB. Tes dilakukan dengan 10 input yang berbeda untuk tiap tahapnya. Element Extractor from Manga berhasil mengekstrak 88% frame komik, 91% balon teks, dan 46% karakter. Pelatihan convolutional neural network mencapai akurasi 96.2% dan tes cross validation mencapai akurasi 86%.**

***Kata Kunci*: Pencitraan Komputer, Aplikasi Dekstop, Pengenalan Pola, Jaringan Saraf Tiruan *Convolutional*, Hiragana, Komik Jepang**

Abstract— **Comics are loved by people all over the world. Japanese comic or manga can be found in internet, but not everyone can read it because it is written in Japanese. There are non-profit teams who are willing to translate manga, but because of limited time and effort not all manga can be translated. Therefore, even though there are many manga in internet, not everyone can read it. So cause disappointment for the readers whom anticipate it. It can be solved by using pattern recognition, and convolutional neural network algorithm. In this research, manga will be used as the input sample. The first step is to retrieve Japanese text from text balloon in manga. After that, perform feature extraction, then analyze and save the pattern to be compared with prepared bank of data so it can be decided what is the character of the source's image. To train the pattern recognition for maximum output, Convolutional Neural Network algorithm is used to make the application smarter in pattern recognition and can run independently. There will be two applications, the first application is Element Extractor from Manga (Japanese Comic), developed using C# programming language and AForge library. And pattern recognition program developed using MATLAB. Based on testing result using 10 different inputs, the Element Extractor from Manga program able to extract 88% frame, 91% text balloon, and 46% character. Character extraction showing a low result and needs more pre-processing steps. The convolutional neural network training reached 96.2% accuracy and 86% accuracy from cross validation test.**

Keywords: Computer Vision, Dekstop Application, Pattern Recognition, Convolutional Neural Network, Hiragana, Japanese Comics

I. PENDAHULUAN

Komik digemari oleh banyak orang di seluruh dunia saat ini. Komik dapat dibedakan dari lokasi asalnya, seperti komik dari Jepang disebut *manga*, komik dari Korea disebut *manhwa*, dan komik dari China disebut *manhua*. Selain dari nama, perbedaan lain juga terdapat pada gaya penggambarannya. Pada era digital saat ini komik telah banyak dipindai dan diunggah ke internet sehingga dapat dinikmati oleh masyarakat dari seluruh dunia. *Manga* adalah komik terbanyak yang beredar di internet, karena berasal dari Jepang maka *manga* berisi tulisan Jepang. Karena minat pembaca *manga* sangatlah banyak, terbentuklah tim non-profit untuk menerjemahkan *manga* dari bahasa Jepang menjadi bahasa Inggris. Namun *manga* yang beredar sangatlah banyak dan tim penerjemah sangat terbatas, oleh sebab itu banyak *manga* yang belum diterjemahkan [1, 2].

[1] *Mahasiswa, Program Studi Teknik Informatika Fakultas Industri Kreatif Universitas Ciputra, Jln. UC Town, Citraland, Surabaya 60219 INDONESIA (tlp: 031-7451699; fax: 031-7451698; e-mail: mmargarita@student.ciputra.ac.id)*

[2] *Dosen, Program Studi Teknik Informatika Fakultas Industri Kreatif Universitas Ciputra, Jln. UC Town, Citralamd, Surabaya 60219 INDONESIA (tlp: 031-7451699; fax: 031-7451698; e-mail: daniel.m.w@ciputra.ac.id)*

[3] *Dosen, Program Studi Teknik Informatika Fakultas Industri Kreatif Universitas Ciputra, Jln. UC Town Citraland, Surabaya 60219 INDONESIA (tlp: 031-7451699; fax: 031-7451698; e-mail: nsugianto@ciputra.ac.id)*

Komputer dapat digunakan untuk mengetik karakter Bahasa Jepang, namun untuk mengenali karakter yang terdapat pada media gambar dibutuhkan komputasi yang besar dimana dipengaruhi oleh akurasinya [3]. Bahasa Jepang memiliki susunan kalimat yang terdiri dari huruf *Hiragana, Katakana*, dan *Kanji*, dan dituliskan dengan kombinasi karakter suku kata (*hiragana* dan *katakana*) dan yang bersifatkan gambar (*kanji*). Selain itu terdapat tambahan diakritik yang jika dipakai akan menghasilkan arti yang berbeda, hal ini membuat Bahasa Jepang memiliki banyak kemungkinan kombinasi [4]. Huruf Jepang tidak memiliki pemisah seperti spasi, dan beberapa karakter Jepang memiliki kemiripan yang menambah kompleksitas dalam pengenalan. Oleh sebab itu OCR untuk Bahasa Jepang adalah penelitian yang menantang dan memerlukan banyak usaha untuk dilaksanakan [4].

Optical Character Recognition dapat digunakan dengan berbagai metode yang ada, salah satunya adalah *neural network* [5]. Cara kerja *neural network* seperti otak manusia yang dapat dilatih untuk menambah pengetahuannya untuk mendapatkan akurasi tinggi. Dengan menganalisa tiap *pixel* gambar dan mencocokkannya dengan data yang telah ada, metode ini cocok untuk dokumen dan teks yang rusak. *Neural network* ideal untuk masalah spesifik seperti data pasar saham atau menemukan trend pola gambar, sejauh ini *neural network* adalah metode yang paling efisien dibandingkan dengan metode lain [5].

Convolutional Neural Network (CNN) adalah salah satu algoritma lanjutan yang dimiliki oleh neural network dan memiliki kelas model yang bagus untuk mengenali *handwritten text*, terutama bilangan digit dan karakter Cina. [6]

Pada penelitian ini akan diterapkan penggunaan *neural network* dengan algoritma 2D *Convolutional Neural Network*. Telah banyak penelitian menggunakan CNN untuk mengenali tulisan handrwitten dalam bahasa Jepang [6], namun belum ada penelitian untuk tulisan cetak.

Tujuan penelitian ini adalah melakukan pengenalan pola huruf Hiragana dengan menggunakan metode 2D Convolutional Neural Network dengan tingkat akurasi minimal 60%.

Untuk mencapai tujuan tersebut penelitian dilakukan dengan melakukan studi pendahuluan terlebih dahulu dengan sumber jurnal, buku, artikel, dan publikasi di internet. Studi yang dilakukan adalah mengenai teknik *computer vision*, *text extraction from image*, pengenalan pola karakter Jepang dan *convolutional neural network.* Setelah studi pendahuluan dilanjutkan dengan pembuatan desain aplikasi seperti desain arsitektur, desain tampilan *user interface* dan desain arsitektur *convolutional neural network.* Implementasi dilakukan dengan membuat aplikasi berdasarkan teori yang didapatkan pada studi pendahuluan dan desain yang telah dirancang. Setelah aplikasi selesai dibuat dilakukan tes untuk menguji peforma dan akurasi aplikasi serta mengetahui kesalahan atau *error* yang terjadi selama aplikasi digunakan. Tes dilakukan dengan menggunakan 10 input yang berbeda untuk tiap tahapnya. Langkah terakhir adalah menyusun laporan dengan menyertakan studi pendahuluan yang digunakan, rancang desain, implementasi aplikasi berserta langkah-langkahnya dan hasil tes yang telah dilakukan.

II. LANDASAN TEORI

a. Komik

Komik adalah seni yang menggunakan gambar yang disusun sedemikian rupa sehingga membentuk jalinan cerita. Komik dicetak di atas kertas dan dilengkapi dengan teks. Komik dapat diterbitkan dalam berbagai bentuk, mulai dari strip dalam koran, dimuat dalam majalah, hingga berbentuk buku tersendiri [1].

Komik yang dibuat di Jepang disebut dengan *Manga*. *Manga* memiliki gaya gambar yang khas hingga seringkali digunakan sebagai referensi menggambar oleh komikus internasional. Di Jepang industri ini dapat mencapai angka 40,6 miliar yen tiap tahunnya. Industri penerbit mengelompokkan *manga* pada usia dan jenis kelamin target pembaca. Untuk laki-laki biasa disebut dengan *shōnen* dan *shōjo* untuk pembaca perempuan [1].

Majalah *manga* biasanya terdiri dari beberapa judul komik yang masing-masing memiliki 30-40 halaman untuk tiap bab-nya, sehingga total jumlah halaman majalah *manga* berkisar 200 hingga 850 halaman. Alur membaca *manga* dari kanan ke kiri karena kebiasaan menulis Bahasa Jepang, berbeda dengan alur membaca Indonesia yang dari kiri ke kanan, karena itu penerbit Indonesia umumnya melakukan *flip* untuk halaman komik sehingga dapat dibaca dari kiri ke kanan, namun hal ini membuat ambiguitas terutama dengan *manga* kategori detektif dimana sering memberikan informasi yang tidak sesuai dengan gambar.

b. Bahasa Jepang

Bahasa Jepang memiliki tiga macam karakter, yaitu Hiragana, Katakana, dan Kanji. Kanji digunakan untuk menyatakan arti dasar dari kata (baik berupa kata benda, kata kerja, kata sifat, atau kata sandang).

Karakter Hiragana dan Katakana memiliki 46 set huruf masing-masing. Hiragana dan Katakana tidak memiliki arti apapun, seperti layaknya abjad dalam Bahasa Indonesia, hanya melambangkan suatu bunyi tertentu, meskipun ada juga kata-kata dalam bahasa Jepang yang terdiri dari satu suku kata, seperti *me* (mata), *ki* (pohon), *ni* (dua). Berbeda dengan kanji yang tiap hurufnya melambangkan suatu arti tertentu.

Dalam kalimat bahasa Jepang tidak ada spasi yang memisahkan antara kata dan tidak ada spasi yang memisahkan antara kalimat. Terdapat dua tanda baca yang dikenal dalam bahasa Jepang yaitu *kuten* (。) yang berfungsi sebagai tanda baca titik, dan *toten* (、) yang berfungsi sebagai tanda baca koma.

Dalam penulisan kalimat Bahasa Jepang dimulai dari atas ke bawah, berbeda dengan alfabet yang dimulai dari kiri ke kanan. Penulisan kalimat selanjutnya diletakkan di sebelah kiri kalimat sebelumnya, sehingga arah baca penulisannya dimulai dari kanan ke kiri [7].

c. Pattern Recognition

Pattern recognition merupakan salah satu cabang pembelajaran dari *machine learning* yang berfokus pada pengenalan pola dan regularitas data. Sistem pengenalan pola dilatih dari data *training* yang telah dilabeli (*supervised learning*), namun saat ada data yang belum di labeli maka algoritma akan mengenali pola tersebut (*unsupervised learning*).

Secara umum, algoritma *pattern recognition* bertujuan untuk menyediakan jawaban yang paling mendekati input. Hal ini bertentangan dengan algoritma *pattern matching* dimana mencari jawaban yang benar-benar sesuai dengan input oleh pola yang telah ada.

Pattern recognition memiliki beberapa proses yaitu, *image retrieval, pre-processing* (untuk menghilangkan *noise* ataupun normalisasi gambar), *feature extraction,* dan *classification. Pattern recognition* bergerak pada proses *classification* [8].

d. Neural Network

Jaringan saraf tiruan atau yang sering disebut dengan *neural network* adalah jaringan dari sekelompok unit pemroses yang menggunakan model jaringan saraf manusia. *Neural network* sangat adaptif dan dapat mengubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir melalui jaringan tersebut.

Secara sederhana, *neural network* adalah alat pemodelan data statistik non linear yang dapat digunakan untuk membuat model hubungan yang kompleks antara input dan output untuk menemukan pola-pola pada data. *Neural network* dibangun dari banyak node/unit/neuron yang dihubungkan oleh link/sinaps secara langsung. Link dari unit yang satu ke unit yang lainnya digunakan untuk melakukan propagasi aktivasi dari unit pertama ke unit selanjutnya. Setiap link memiliki bobot numerik (weight). Bobot ini menentukan kekuatan serta penanda dari sebuah konektivitas [9].

Neural network terdiri dari banyak neuron yang dikelompokkan ke dalam beberapa layer. Neuron pada tiap layer terhubung dengan neuron pada layer lainnya. Informasi yang diterima di layer input dilanjutkan ke layer-layer dalam *neural network* secara satu persatu hingga mencapai layer terakhir/output. Layer yang terletak di antara input dan output disebut *hidden* layer. Banyaknya *hidden* layer tidak tetap, bergantung pada kebutuhan. Metode dasar dari *neural network* adalah *multilayer perceptron.*

e. Convolutional Neural Network

Convolutional Neural Network (CNN) adalah pengembangan dari Multi*layer* Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra.

Cara kerja CNN memiliki kesamaan pada MLP, namun pada CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi. Bobot pada CNN berbentuk empat dimensi yang merupakan kumpulan kernel konvolusi. Dimensi bobot pada CNN adalah:

neuron input x neuron output x tinggi x lebar.

Karena sifat proses konvolusi, CNN hanya dapat digunakan pada data yang memiliki struktur dua dimensi seperti gambar dan suara. CNN umumnya menggunakan implementasi dari LeCun, yaitu LeNet. LeNet terdiri dari beberapa *layer*, jenis *layer* yaitu *convolution* dan *subsampling.*

Layer convolution digunakan untuk mengaplikasikan 'filter' pada gambar. Filter terdiri dari *layer* hubungan bobot, dengan input ukuran gambar patch kecil 2D, dan outputnya unit tunggal. Filter ini diaplikasikan berulang kali sehingga menghasilkan hubungan seperti serangkaian bidang *receptive fields*.

Layer Subsampling mengacu pada pengurangan rata-rata besar sinyal. Metode spesifik *subsampling* yang digunakan dikenal dengan 'max pooling', yang meliputi pemisahan matriks filter output menjadi *grid* kecil yang *non-overlapping* (semakin besar *grid*, semakin besar sinyal reduksi), dan mengambil nilai maksimal pada tiap *grid* sebagai nilai matriks yang tereduksi. Singkatnya, *layer* ini merespon perubahan yang terjadi pada *layer convolutional*. Dengan mengaplikasikan *layer* ini diantara *layer convolutional*, maka dapat meningkatkan *spatial abstractness* seiring meningkatnya *feature abstractness*.

f. Inisialisasi Awal Xavier

Inisialisasi awal merupakan langkah yang sangat penting pada pembelajaran *neural network*. Tujuan dari inisialisasi awal adalah memberikan nilai bobot awal sebelum pelatihan dilakukan. Dimana saat pelatihan dilakukan bobot akan disesuaikan kembali untuk menghasilkan klasifikasi yang akurat.

Untuk menginisialisasi bobot awal terdapat beberapa cara, cara yang paling sederhana adalah memberi nilai acak antara 0 atau 1. Namun setelah diteliti lebih lanjut penentuan bobot awal dengan metode tertentu dapat meningkatkan hasil klasifikasi lebih baik. Salah satu metode tersebut adalah menggunakan metode Xavier, dimana inisialisasi yang umum digunakan adalah [10] :

$$Var(W) = \frac{1}{n_{in}} \quad (1)$$

Dimana W adalah distribusi inisialisasi untuk neuron, sedangkan n_{in} adalah jumlah neuron feeding. Distribusi yang umum digunakan adalah Gaussian atau seragam. Pada metode Xavier digunakan [10] :

$$Var(W) = \frac{2}{n_{in} + n_{out}} \quad (2)$$

Dimana n_{out} adalah jumlah hasil neuron yang telah di feeding.

III. ANALISIS DAN DESAIN

a. Analisis Masalah

Pada studi yang dilakukan oleh Tsai [11] didapati kekurangan jenis data latih. Seluruh data latih yang digunakan adalah *handwritten* dan tidak ada huruf cetak.

Huruf cetak selalu digunakan pada media elektronik, sehingga walaupun dapat membaca tulisan *handwritten* dengan baik namun jika sistem tidak dapat membaca huruf cetak, maka akan berakibat fatal.

Pada studi yang sudah pernah dilakukan [12] dijelaskan bagaimana metode pengambilan karakter dari komik secara sistematis, namun pada studi tersebut tidak dijelaskan selanjutnya akan digunakan untuk apa hasil dari metode tersebut. Sehingga studi tersebut tidak memiliki tujuan konkrit untuk mengekstrasi elemen komik.

Pada studi yang sudah pernah dilakukan [11] mengenali karakter Jepang dengan menggunakan cara perhitungan manual, dengan mencari titik tengah dan titik potong. Kekurangan dari metode ini adalah terbatasnya nilai *feature* dari karakter, dimana pada paper ini *feature* yang dimiliki hanyalah 4, padahal seharusnya diperlukan 46 jenis huruf yang harus diklasifikasikan.

b. Solusi

Berdasarkan permasalahan yang didapatkan dari penelitian terdahulu, terdapat beberapa solusi yang ditawarkan. Adapun pemetaan solusi dapat dilihat dalam tabel berikut.

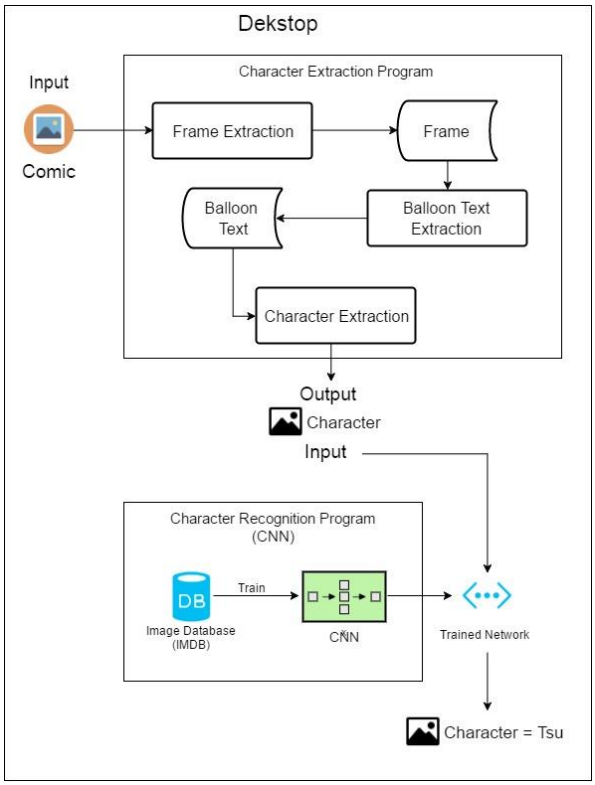
TABEL I
PEMETAAN MASALAH DAN SOLUSI

Masalah	Solusi
Data latih pengenalan karakter Jepang hanya menggunakan jenis tulisan *handwritten*	Menggunakan data latih dengan jenis tulisan cetak
Tidak adanya tujuan konkrit untuk mengekstrasi elemen komik	Memberikan tujuan konkrit setelah proses ekstrasi selesai, yaitu mengenali pola karakter
Pengenalan karakter dengan cara penghitungan manual menghasilkan sedikit *feature*	Pengenalan karakter menggunakan cara Convolutional Neural Network untuk menghasilkan banyak *feature*

c. Desain Sistem

i. Desain Arsitektur

Pada penelitian ini dibangun dua aplikasi, yaitu aplikasi pengambilan karakter dan aplikasi pengenalan pola Hiragana. Input pada aplikasi pengambilan karakter adalah gambar komik, dimana output dari aplikasi tersebut adalah gambar karakter yang di dapatkan dari dalam balon teks pada komik. Pada program pengenalan pola telah disiapkan data latih dalam *image database* (IMDB) untuk dilatih pada jaringan *convolutional neural network*, dimana output dari aplikasi tersebut adalah jaringan yang telah dilatih. Output dari aplikasi pertama digunakan sebagai input untuk jaringan yang telah dilatih mengidentifikasi karakter Hiragana input gambar tersebut, sehingga outputnya adalah hasil klasifikasi pengenalan karakter.

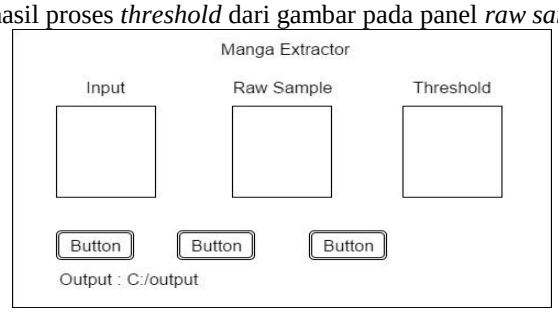


Gambar 1. Desain Arsitektur

ii. Desain User Interface

Gambar 2 adalah desain tampilan dari aplikasi ekstraksi komik. Terdapat tiga tombol untuk memproses tiga jenis input. Tombol pertama untuk input berupa satu halaman komik, tombol kedua untuk input berupa satu *frame* komik, dan tombol terakhir untuk input berupa satu balon teks.

Lalu terdapat 3 gambar yang dipergunakan untuk menampilkan input dari pengguna pada panel input. Pada panel *raw sample* untuk menunjukkan hasil ekstraksi yang belum di proses. Pada panel *threshold* untuk menunjukkan hasil proses *threshold* dari gambar pada panel *raw sample.*

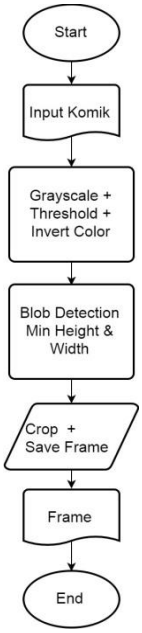


(Gambar 2 Desain *User Interface)*

iii. Desain Algoritma Pengambilan Karakter

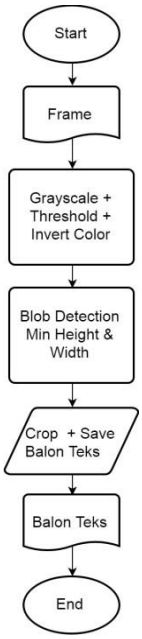
Untuk mengambil karakter dari input komik dibutuhkan 3 proses. Proses pertama ialah pengambilan frame satu-

persatu dari input komik, dimana output dari proses pertama ini adalah *frame-frame* yang terdapat pada komik [12].



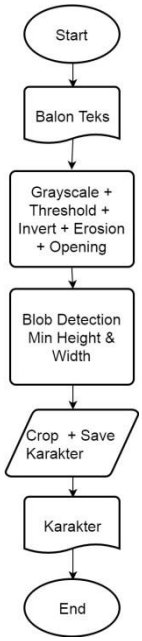
Gambar 3. Algoritma Pengambilan *Frame*

Proses kedua adalah pengambilan balon teks dari frame yang telah diekstraksi dari proses sebelumnya. Output dari proses ini adalah balon teks dari frame.



Gambar 4. Algoritma Pengambilan Balon Teks

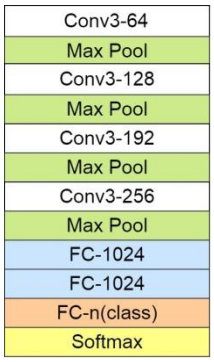
Proses ketiga dan yang terakhir adalah pengambilan karakter dari balon teks yang telah diekstraksi dari proses sebelumnya. Output dari proses ini adalah karakter dari balon teks.



Gambar 5. Algoritma Pengambilan Karakter

iv. Desain Algoritma Convolutional Neural Network

Input dari CNN adalah berupa gambar, berbeda dengan NN dimana inputnya berupa data. Dilanjutkan dengan *layer convolutional*, dimana input gambar di proses untuk mendapatkan fitur-fiturnya. Output dari *layer convolutional* menjadi node input untuk *layer* input bagian NN. Model dari CNN yang digunakan adalah model M7-2 [11] dengan beberapa perubahan untuk menyesuaikan dengan data input.



Gambar 6. Model CNN M7-2 (Tsai, 2016)

Maksud dari Conv3-64 adalah *layer convolution* dengan matrix 3x3 dengan 64 *feature* output, semua *layer convolution* menggunakan 1 *stride* dan 1 *padding*. Untuk semua *layer maxpool* menggunakan matrix 2x2, 2 stride, dan 0 padding. *Layer* FC-1024 adalah *layer fully-connected* dengan 1024 *feature* output menggunakan *layer convolution*, dimana *layer* FC pertama untuk membuat output menjadi 1x1 dimensi, sehingga matrix *layer* FC pertama menyesuaikan dengan sisa matrix dari *layer* sebelumnya. Untuk output *layer* FC terakhir menyesuaikan dengan banyaknya *class* yang ingin diklasifikasikan.

$$f(x_i; W) = W\, x_i \quad (3)$$

Layer terakhir *softmax* digunakan untuk mendapatkan hasil klasifikasi, dimana fungsi mapping menghasilkan nilai yang diintrepetasi sebagai probabilitas yang belum dinormalisasi untuk tiap kelas. Nilai kelas dihitung dengan menggunakan fungsi softmax [3]:

$$y_{ijk} = \frac{e^{x_{ijk}}}{\sum_{t=1}^{D} e^{x_{ijt}}} \quad (4)$$

Dimana x adalah vector yang berisi nilai yang didapatkan dari *layer* fully-connected terakhir dan y_{ijk} adalah vector yang berisi nilai antara 0 dan 1, dan jika dijumlah hasilnya adalah 1. Fungsi loss dihitung dengan [13]:

$$l(x,c) = -log\frac{e^{x_c}}{\sum_{k=1}^{C} e^{x_k}} = -x_c + log\sum_{k=1}^{C} e^{x_k} \quad (5)$$

Dimana $l(x,c)$ untuk membandingkan prediksi (x) dengan label (c). x adalah vector dari probabilitas akhir $p(k) = x_k,\ k = 1$, dan C adalah banyak kelas.

IV. IMPLEMENTASI

a. Implementasi Pengambilan Karakter

Pengambilan karakter diambil dari input yang berupa komik. Terdapat 3 langkah untuk pengambilan karakter dari input komik, yaitu mengambil frame yang terdapat pada satu halaman komik, mengambil balon teks yang terdapat pada frame yang diambil sebelumnya, dan mengambil karakter dari balon teks yang diambil sebelumnya.

Gambar 7. Contoh Input Gambar

i. Pengambilan Frame Komik

Input yang akan digunakan untuk pengambilan *frame* komik adalah satu halaman komik seperti Gambar 7. *Frame* yang dapat dikenali haruslah memenuhi beberapa syarat sebagai berikut:

- Frame berbentuk kotak
- Frame tidak tertumpuk ataupun tertutup apapun
- Frame memiliki *space* putih disekelilingnya
- Frame memiliki border
- Tinggi dan lebar minimal *frame* adalah ¼ input gambar

Langkah-langkah untuk mendeteksi dan memotong per-frame dijelaskan sebagai berikut :

1. Mengubah input gambar menjadi *grayscale*
2. Melakukan *threshold* pada gambar
3. Mengubah warna dengan *invert*
4. Menentukan tinggi dan lebar minimal dari frame sesuai syarat
5. Melakukan *BlobCounter* dengan memasukan kriteria minimal tinggi dan lebar
6. Jika *blob* yang didapatkan sesuai dengan kriteria maka di *crop* dan disimpan

Gambar 8. Hasil Ekstraksi *Frame* Komik

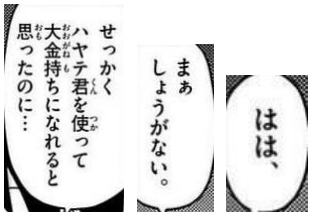
ii. Pengambilan Balon Teks

Pengambilan balon teks dilakukan dengan *frame* yang telah diambil pada proses sebelumnya. Balon teks yang dapat dikenali harus memenuhi beberapa syarat berikut :

- Latar belakang putih
- Tidak bertumpukan dengan tulisan atau balon teks lain
- Lebar balon teks minimal 1/8 dari lebar *frame*
- Tinggi balon teks minimal 1/6 dari tinggi *frame*

Langkah-langkah untuk mendeteksi dan memotong balon teks dijelaskan sebagai berikut :

1. Mengubah input gambar menjadi *grayscale*
2. Melakukan *threshold* pada gambar
3. Mengubah warna dengan *invert*
4. Menentukan tinggi dan lebar minimal dari frame sesuai syarat
5. Melakukan *BlobCounter* dengan memasukan kriteria minimal tinggi dan lebar
6. Jika *blob* yang didapatkan sesuai dengan kriteria maka di *crop* dan disimpan



Gambar 9. Hasil Ekstraksi Balon Teks

Adapun untuk gambar hasil kesalahan ekstraksi adalah sebagai berikut.

Gambar 10. Hasil Kesalahan Ekstraksi Balon Teks

iii. Pengambilan Karakter

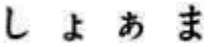
Langkah-langkah untuk mendeteksi dan memotong per-karakter dijelaskan sebagai berikut :

1. Mengubah gambar menjadi *grayscale*
2. Menghilangkan *noise* pada gambar dengan *Median*
3. Melakukan threshold pada gambar dengan value threshold didapatkan dengan menggunakan rumus:

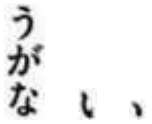
$$\mu = \frac{1}{N}\sum_{i=0}^{N-1} I_i \quad (6)$$

I_i = nilai pixel pada index i

4. Mengimplementasikan *morphology* dengan menggunakan filter *erosion*
5. Menentukan minimal lebar karakter 1/20 dari balon teks dan tinggi karakter 1/25 dari balon teks
6. Mengubah warna dengan *invert*
7. Melakukan *BlobCounter* dengan memasukan kriteria minimal lebar dan tinggi
8. Jika *Blob* yang didapatkan sesuai dengan kriteria maka di *crop* dan disimpan



(Gambar 11 Hasil Ekstraksi Karakter)



(Gambar 12 Hasil Ekstraksi Karakter Kurang Sempurna)

b. Implementasi *Convolutional Neural Network*

Dalam mengimplementasikan CNN terdapat 2 tahap, yaitu pelatihan (*training*) dan pengujian. Tahap pelatihan adalah tahap utama sebelum melakukan tahap pengujian. Pada tahap pelatihan didapatkan hasil *network* yang telah dilatih, dimana *network* tersebut dapat digunakan selanjutnya pada tahap pengujian.

i. Data Latih

Data latihan memiliki 460 data yang terdiri dari 46 jenis huruf Hiragana. Dimana pada tiap jenis hurufnya terdapat 10 gambar dengan jenis tulisan yang berbeda. 80% dari total gambar digunakan sebagai data latih dan 20% sisanya digunakan sebagai data tes.

ii. Parameter Pelatihan

Pelatihan dilakukan menggunakan parameter sebagai berikut: *learning rate* 0,0001, 16 *batch size*, 0,0001 *weight decay* dan 100 *epoch* [11]. Inisialisasi bobot awal menggunakan metode Xavier [10]. Komputasi dilakukan dengan menggunakan mode CPU.

V. HASIL PENGUJIAN

Pengujian untuk ekstraksi elemen komik dilakukan dengan tiga tahap, yakni pengambilan *frame*, pengambilan balon teks, dan pengambilan karakter.

Implementasi pengenalan karakter menghasilkan output berupa akurasi pelatihan, *objective loss* training, dan akurasi dari uji tes *cross validation*.

Pada Tabel II tertera rangkuman hasil pengujian yang telah dilakukan untuk aplikasi ekstraksi elemen komik dan pengenalan karakter.

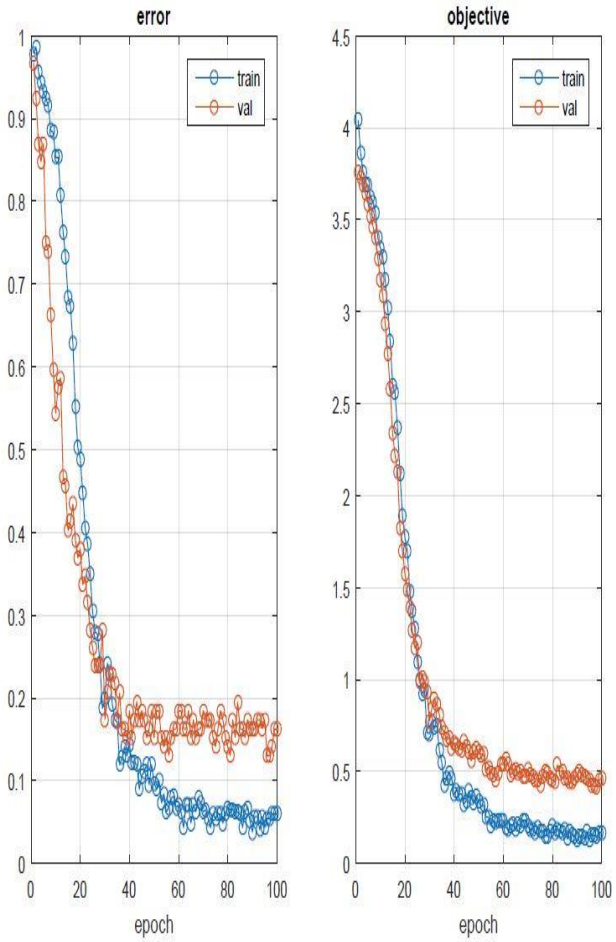
TABEL II. HASIL PENGUJIAN

Ekstraksi Elemen Komik	
Pengambilan Frame	
Benar	88%
Salah	12%
Pengambilan Balon Teks	
Benar	91%
Salah	9%
Pengambilan Karakter	
Benar	46%
Salah	54%
Pengenalan Karakter	
Akurasi Training	96,2 %
Obejctive Loss	0,112
Akurasi Uji *Cross Validation*	86%

Pada pengambilan *frame* perlu diperhatikan bahwa *frame* yang dapat dikenali hanyalah yang berbentuk kotak dan memiliki *space* putih di ke empat sisinya, sehingga untuk *frame* yang memiliki bentuk selain kotak dan *frame* yang berhimpitan dengan batas luar halaman komik tidak dapat dikenali. Untuk *frame* yang bertumpukan satu sama lain tidak dapat dikenali sebagai *frame* yang berbeda sehingga dianggap kesatuan *frame* tersebut, berlaku juga dengan tulisan maupun balon teks yang bertabrakan dengan *frame*.

Pada pengambilan balon teks didapatkan akurasi 91% dimana mendekati sempurna. Metode yang digunakan untuk mengambil balon teks sudah tepat dan hanya dibutuhkan sedikit tambahan untuk mengatasi kesalahan pendeteksian wajah sebagai balon teks.

Pengambilan karakter didapati akurasi 46% sehingga dibutuhkan metode yang lebih tepat untuk memperbaiki hasil tersebut. Kesalahan disebabkan oleh terlalu rapatnya karakter satu sama lain, sehingga dibaca sebagai kesatuan huruf.

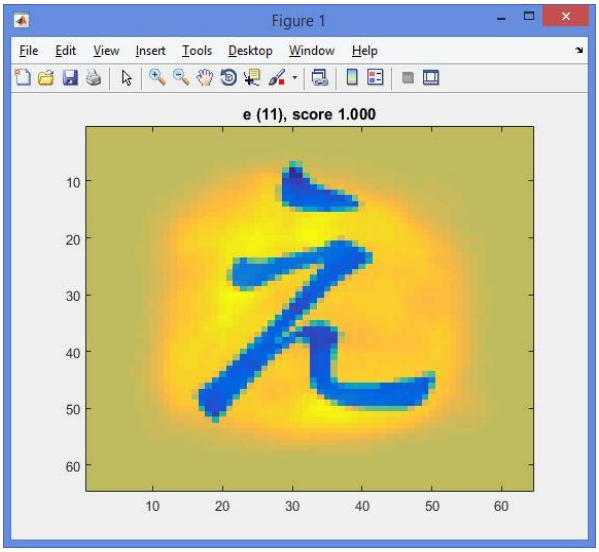


Gambar 13. Hasil *Training*

Pengujian pengenalan karakter dilakukan dua kali, yaitu pengujian akurasi *training* dan akurasi tes. Gambar 13 menunjukan grafik hasil *training* menggunakan CNN. Semakin bertambahnya *epoch* tingkat *error* turun hingga mencapai titik 0,048 pada *epoch* ke-84 untuk training, dan untuk validasi tingkat *error* mencapai titik 0,125 pada *epoch* ke-83.

Begitu pula dengan *objective loss*, semakin bertambahnya *epoch loss*-nya semakin turun hingga mencapai titik 0,112 pada *epoch* ke-90 untuk training, dan untuk validasi *loss*-nya mencapai 0,486 pada *epoch* ke-90.

Dengan demikian hasil akhir yang didapatkan dari hasil latihan ini adalah akurasi mencapai 96,2%. Selain itu juga dilakukan pengujian *cross-validation* dimana hasilnya adalah 79 gambar dinyatakan benar klasifikasinya dari total 92 gambar, sehingga akurasi tes adalah 86%. Salah satu contoh karakter yang diuji menggunakan *cross validation* ditunjukkan pada Gambar 14.



Gambar 14. Uji *Cross Validation*

Pada Gambar 14 nilai yang didapatkan dari klasifikasi gambar tersebut adalah 1 dimana 1 sebanding dengan 100%, sedangkan "e" adalah hasil klasifikasi dari gambar tes tersebut, dan 11 adalah kode dari label "e".

VI. KESIMPULAN DAN SARAN

a. Kesimpulan

Berdasarkan rancangan dan implementasi yang telah dilakukan, maka didapatkan kesimpulan berikut:

1. Penelitian ini berhasil mengimplementasikan desain dan algoritma untuk mengekstraksi elemen komik.
2. Arsitektur jaringan *convolutional neural network* yang digunakan menghasilkan akurasi 96,2%
3. Metode untuk memotong *frame* dan balon teks dapat diimplementasikan dan menghasilkan akurasi >80%
4. Metode untuk memotong karakter masih belum tepat dan menghasilkan akurasi yang rendah.

b. Saran

Berdasarkan kesimpulan yang telah dijabarkan, saran yang diberikan untuk pengembangan lebih lanjut dalam pengambilan karakter dan pengenalan pola disampaikan sebagai berikut:

1. Menyempurnakan metode pengambilan karakter sehingga karakter dapat digunakan secara baik dalam proses pengenalan menggunakan *convolutional neural network.*
2. Menambah data latih untuk *convolutional neural network* dengan berbagai variasi huruf dan melakukan *training* kembali secara berkala.
3. Mengkaji kembali arsitektur jaringan *convolutional neural netwok* untuk mencapai hasil yang lebih optimal.
4. Mencari cara menggunakan algoritma lain untuk menemukan *learning rate* yang optimal bagi arsitektur jaringan *convolutional neural network.*

DAFTAR PUSTAKA

[1] Fusanosuke, N. (2003). Japanese Manga: Its Expression and Popularity. *ABD, 34*(1), 3-5.

[2] Moeran, B. (2014). Japanese "Merchants of Culture": The Publishing Business in Japan. *International Journal of Culture, Tourism and Hospitality Research, 4*(4), 97-125.

[3] Vedaldi, A., Lenc, K., & Gupta, A. (2014). *MatConvNet Convolutional Neural Network for MATLAB.* Oxford: MatConvNet.

[4] Das, S., & Banerjee, S. (2015). An Algorithm for Japanese Character Recognition. *International Journal of Image, Graphics, and Signal Processing , 1*, 9-15.

[5] Rao, N. V., Sastry, A. C., Chakravarthy, A. N., & Kalyanchakravarthi, P. (2016, January). Optical Character Recognition Technique Algorithms. *Journal of Theoretical and Applied Information Technology, 83*(2), 275-282.

[6] Tsai, C. (2016). *Recognizing Handwritten Japanese Characters Using Deep Convolutional Neural Networks.* Stanford University, Department of Chemical Engineering. California: Stanford University.

[7] Sakamoto, K. (2015). *Translation of Japanese Poems into English: Literature in the First Language as a Motive to Communicate in a Second Language.* Tokyo: Palgrave Macmillan UK.

[8] Rigelsford, J. (2002). Pattern Recognition: Concepts, Methods and Applications. *Assembly Automation, 22*(4), 10.

[9] Rughooputh, H. S., & Rughooputh, S. D. (2002). Neural Network Process Vision Systems for Flotation Process. *Kybernetes, 31*(3/4), 529-535.

[10] Goldberg, Y. (2017). *Neural Network Methods in Natural Language Processing.* Toronto: Morgan & Claypool Publisher.

[11] Tsai, C. (2016). *Recognizing Handwritten Japanese Characters Using Deep Convolutional Neural Networks.* Stanford University, Department of Chemical Engineering. California: Stanford University.

[12] Arai, K., & Tolle, H. (2011). Method for Real Time Extraction of Digital Manga Comic. *International Journal of Image Processing (IJIIP), 4*(6), 669-676.