

# Pencarian Informasi Berbasis Teks dalam Komik Digital Menggunakan OCR

Anil Kumar Siliwangi<sup>1)</sup>, Yulius Denny Prabowo<sup>2)</sup>

Informatika, Fakultas Ilmu Komputer dan Desain, Institut Teknologi dan Bisnis Kalbis  
Jalan Pulomas Selatan Kav. 22, Jakarta 13210

<sup>1)</sup>Email: anil.kumar.s0202@gmail.com

<sup>2)</sup>Email: yulius.prabowo@kalbis.ac.id

**Abstract:** This research is aimed to research and develop a web-based application which will enable user to find texts in a digital comic pages using OCR. The methods that will be used to achieve this is a neural network learning approach using YOLOv3 algorithm that will enable a program to detect text in a comic page and a text recognition engine developed by google called Tesseract that will enable the program to read texts in a comic page to a functional format such as string text. The software development cycle that is used is the incremental model and the user interface will be made using a python based web framework called Flask using Hyper Text Markup Language, Cascading Style Sheet and Javascript for searching functionality

**Keywords:** OCR, YOLOv3, Tesseract, website, incremental

**Abstrak:** Penelitian ini dibuat dengan tujuan untuk meneliti dan mengembangkan aplikasi berbasis web yang akan membuat pengguna untuk dapat mencari teks dalam sebuah halaman komik digital menggunakan OCR. Metode yang diteliti untuk mencapai penelitian adalah pendekatan neural network menggunakan algoritma YOLOv3 sebagai cara untuk mencari lokasi teks dalam sebuah gambar dan engine Tesseract yang dikembangkan oleh google untuk membuat teks dalam format yang fungsional seperti seting. Alur pengembangan web yang dipakai adalah model inkremental dan antarmuka pengguna akan dibuat menggunakan framework web berbasis python bernama Flask menggunakan Hyper Text Markup Language, Cascading Style Shee dan Javascript untuk fungsi pencarian.

**Kata kunci:** OCR, YOLOv3, Tesseract, website, inkremental

## I. PENDAHULUAN

. Komik didefinisikan sebagai salah satu bentuk literatur yang memakai paduan gambar dan kalimat untuk menyampaikan suatu cerita. Pemakaian komik sendiri sebagai pembawa cerita telah berkembang selama beberapa era dekade terkini, komik yang awalnya dianggap sebagai pembawa cerita yang lebih ditujukan untuk anak-anak sekarang dapat dianggap sebagai bentuk literatur yang serius, dengan pemakaian gambar dan dialog sebuah cerita dapat memakai simbolisme, penyusunan perspektif, dan adegan untuk menyusun

sebuah cerita yang dalam dan bermakna dimana orang dapat membuat esai dan analisa terhadap cerita-cerita tersebut. Berdasarkan pengalaman dan observasi penulis dalam komunitas penggemar komik, salah satu masalah umum yang dimiliki adalah kesulitannya dalam mencari referensi komik digital yang memiliki banyak bab seperti berjumlah 500 sampai 1000 bab. Pencarian teks tidak bisa dilakukan dalam sebuah citra digital, maka penelitian ini ditujukan untuk menggunakan teknologi OCR untuk membuat pembacaan dan penyusunan teks dalam halaman komik untuk dibuat sebuah sistem pencarian informasi. Sebelumnya telah terdapat penelitian terdahulu terkait bidang

aplikasi OCR terhadap ekstraksi teks dan segmentasi terhadap dokumen juga terdapat penelitian yang terkait spesifik terhadap komik seperti ekstraksi *speech bubble* menggunakan deep learning. Dalam penelitian yang dilakukan oleh Dr. M. Sundaresan et al. dalam jurnal mereka yang berjudul “*Text Extraction from Digital English Comic Image Using Two Blobs Extraction Method*”[1] . Dalam jurnal tersebut pengambilan teks dalam komik dilakukan melalui proses dua langkah yaitu *balloon detection* dimana dialog teks dideteksi dan *text extraction*. Pengambilan dialog teks dilakukan dengan algoritma *Connected Component-Labeling* untuk menentukan area yang terhubung untuk tiap daerah *pixel*. Area yang telah di deteksi diaplikasikan *text recognition* melalui proses segmentasi, *feature extraction* dan *classification*. Dalam jurnal ini proses pendeteksian diimplementasi menggunakan MATLAB versi 10[1].

Penelitian pendeteksian teks dalam kasus komik digital selanjutnya dilakukan dalam jurnal *CBCS – Comic Book Cover Synopsis: Generating Synopsis of a Comic Book with Unsupervised Abstractive Dialogue* oleh MS Karthika Devi et al[2]. Jurnal tersebut memiliki tujuan penelitian utama dalam pembuatan sinopsis atau ringkasan otomatis dari sebuah buku komik. Dalam penelitian tersebut penelitian komponen komik dilakukan menggunakan *Hough transform*, *recursive X-Y cut*, *density gradient* dan *connected component-labeling* dan pengenalan teks digunakan melalui engine *OCR Tesseract*[2]

Penelitian tentang deteksi teks dalam kasus komik juga pernah dilakukan dalam jurnal yang berjudul *Comic Text Detection Using Neural Network Approach* oleh Frederic Rayar dan Seiichi Uchida[3]. Dalam penelitian tersebut *text detection* dilakukan menggunakan model *EAST (Efficient*

*and Accurate Scene Text Detector)* dan dataset halaman komik yang telah dianotasi bernama dataset *eBDtheque* berisi 100 halaman komik Amerika, Jepang dan Prancis, halaman komik Jepang dihilangkan karena berdasarkan eksperimen sebelumnya model *EAST* tidak dapat mendeteksinya dengan baik[3].

Berdasarkan penelitian terdahulu tersebut dan permasalahannya peneliti mengusulkan untuk mengembangkan model untuk melakukan OCR terhadap gambar halaman komik digital dengan menggabungkan dua model yakni *YOLOv3* untuk *text detection* yang berbasis dengan *Convolutional Neural Network* dan *library python Tesseract* yang merupakan engine OCR *open-source* yang menggunakan model *LSTM (Long short-term memory)* untuk *text recognition*, sebuah arsitektur yang berbasis dengan *artificial recurrent neural network* dan menerapkan salah satu modelnya terhadap sebuah aplikasi berbasis web yang nantinya akan bisa melakukan pencarian terhadap halaman-halaman komik yang sudah diproses

## **B. Pengumpulan Dataset**

*Dataset* yang penulis kumpulkan berasal dari database komik *public domain* dari *digitalcomicmuseum.com*. Penulis memilih dua judul komik untuk dijadikan *dataset* bahan training yaitu komik berjudul *Forbidden World seri 49* dan *Mickey Finn seri 7*. Dalam data yang telah diambil terdapat beberapa halaman tidak berupa komik seperti halaman iklan, halaman-halaman tersebut akan dihapus karena tidak mengandung data yang terpakai untuk training, yakni dalam berupa *text bubble dialog*.

## 1. Anotasi Gambar

Sebelum *dataset* dapat digunakan tiap gambar yang akan dibuat untuk *training* perlu dilakukan anotasi untuk memberitahu model objek yang ingin dideteksi. Hal ini dicapai dengan dilakukan penggambaran *bounding box* terhadap tiap badan teks dalam halaman komik.



Gambar 1 Anotasi Gambar

## B. Tesseract

*Tesseract* adalah sebuah engine untuk *OCR* yang dikembangkan di perusahaan *Hewlett-Packard (HP)* antara tahun 1984 dan 1995. Proses *Tesseract* mengikuti alur yang tradisional. Langkah pertama adalah menyimpan analisa komponen yang terhubung, dan garis besar tersebut dikumpulkan menjadi *blob*. *Blob* tersebut diatur menjadi garis kalimat dan area di garis tersebut di analisa untuk kemiringan. Garis kata tersebut dipecah lagi menjadi kata yang terpisah tergantung jarak spasi. Proses pengenalan katanya lalu dilalui dua langkah, dalam pass pertama dicoba untuk mengenali tiap kata secara bergiliran [4]. Semenjak tanggal 29 Oktober 2018 *Tesseract* versi 4 menggunakan arsitektur *recurrent neural network LSTM* yang berarti *long-term short memory*. *Reccurent neural network (RNN)* adalah salah satu model *neural network* berbeda

dari tradisional dimana terdapat koneksi antara *node* dalam sebuah sekuens sementara, pada dasarnya *reccurent neural network* dapat membawa memori secara perulangan [5]

## C. Image Thresholding

*Thresholding* adalah teknik yang paling sering digunakan untuk segmentasi gambar. Tujuan dari *thresholding* adalah untuk menentukan nilai *threshold* bagian dari gambar menjadi area yang berarti[6]. *Thresholding* adalah langkah yang penting dalam pemrosesan citra digital seperti pengenalan gambar, pengenalan teks dan peningkatan kualitas gambar. Proses *thresholding* bisa diterapkan pada nilai yang melambangkan warna biner grayscale seperti 0 dan 1 atau properti seperti rata-rata dan tekstur[7].

## D. YOLOv3

*YOLO* adalah pendekatan baru terhadap algoritma pendeteksian objek, pada penelitian terdahulu dalam pendeteksian objek biasanya menggunakan sebuah *classifier* untuk melakukan deteksi, dengan *YOLO*, pendeteksian tersebut dilakukan dengan melakukan pemisahan antara *bounding box* dan probabilitas kelas. Satu jaringan *neural network* dapat memprediksi fitur tersebut dari satu input gambar dalam satu evaluasi[8].

## E. Darknet

*Darknet* adalah *framework neural network open source* yang ditulis dalam bahasa pemrograman *C* dan menggunakan *platform CUDA* yang dikembangkan *NVIDIA* untuk dapat mengakses set instruksi dalam *GPU* yang mendukung fitur *CUDA*[9]. *Darknet* juga telah

terintegrasi dengan *library python OpenCV* untuk membaca *file weight* hasil training untuk digunakan dalam *python* secara langsung. Opsi *CUDA* juga merupakan fitur opsional jika pengguna tidak memiliki *GPU* yang didukung[10]

#### **F. Optical Character Recognition (OCR)**

*Optical Character Recognition*, pengenalan karakter atau yang bisa disebut OCR adalah salah satu bidang dalam *computer vision* dan hubungan komputer manusia. OCR bekerja dengan cara memisahkan kalimat menjadi kata dan simbol huruf sebelum dilakukan pengenalan. Karena kualitas pengenalan sangat bergantung terhadap kualitas gambar dokumen tersebut maka dilakukan beberapa *pre-processing* untuk membuat algoritma dapat menangkap deteksi lebih baik seperti[11]

#### **G. Python**

*Python* adalah sebuah bahasa pemrograman yang *high-level*, *interpreter* dan *general purpose*. *Interpreter* yang berarti python dapat mengeksekusi sebuah instruksi secara langsung tanpa di *compile* atau diubah kedalam bahasa mesin terlebih dahulu, *high-level* yang berarti bahasa yang digunakan memiliki abstraksi yang tinggi dari bahasa mesin dasar, *general purpose* yang berarti python dibuat untuk dapat diaplikasikan ke serangkaian aplikasi yang tidak tertentu. Python memiliki penekanan terhadap kemudahan pembacaan kode dengan adanya penggunaan indentasi, model *object-oriented* dari python juga dibuat dengan tujuan untuk penulisan kode yang jelas untuk projek besar dan kecil[12]

#### **H. OpenCV**

OpenCV adalah *library* pemrograman yang memiliki fungsi utama dalam aplikasi *computer vision* atau pemrosesan gambar digital. OpenCV memiliki banyak fungsi dan alat untuk membantu masalah *computer vision* dari pemrosesan gambar sederhana seperti transformasi dan *filtering* gambar sampai fungsi level tinggi seperti deteksi wajah, pelacakan dan *feature tracking*[13]

#### **I. Flask**

*Flask* adalah sebuah *framework* web yang digunakan untuk membuat sebuah aplikasi web berbasis bahasa *python*. *Flask* dikategorikan sebagai *microframework*[14] karena ia tidak memiliki fungsionalitas bawaan seperti abstraksi database dan *form-validation* tetapi pengguna dapat menambahkan ekstensinya sendiri ke dalam *Flask* untuk diintegrasikan. Spesifikasi web *Flask* menggunakan *WSGI (Web Server Gateway Interface)* yaitu konvensi pemanggilan perintah sebuah aplikasi web terhadap server yang ditulis dalam bahasa pemrograman *python*. [15]

#### **J. HTML**

HTML atau *Hypertext Markup Language* adalah sebuah bahasa markup yang ditujukan untuk ditampilkan pada sebuah halaman web. Bahasa markup adalah bahasa dalam pemrosesan teks komputer untuk membuat anotasi tiap bagian teks maka jika saat sebuah dokumen di keluarkan atau ditampilkan anotasi tersebut tidak terlihat. HTML memiliki banyak elemen yang menjadi basis untuk sebuah halaman web seperti paragraf, gambar dan

objek seperti formulir[16]. HTML dibuat untuk membuat dokumen yang terstruktur untuk membuat anotasi untuk bagian dokumen seperti judul, paragraf, daftar dan bagian lainnya. Tampilan dan fungsionalitas dari HTML dapat dibantu oleh bahasa Cascading Style Sheet dan Javascript[17]

### K. Cascading Style Sheet

Cascading Style Sheet atau CSS adalah bahasa style sheet yang digunakan untuk memerintahkan dan mengatur format penampilan dalam HTML. Bahasa ini digunakan untuk menambahkan penampilan dan juga mengatur penampilan yang berbeda untuk tiap format alat yang berbeda seperti dalam monitor dan mobile. Perintah CSS terbaca setelah pemuatan halaman HTML[18]

### L. Bootstrap

Bootstrap adalah *framework* CSS yang didesain untuk pengembangan front-end website dengan pemikiran utama untuk format mobile. *Bootstrap* pertama dikembangkan untuk *Twitter* sebagai cara menjaga konsistensi dari penampilan di tiap halaman. Sekarang *bootstrap* memiliki banyak fungsionalitas dari *library* CSS dan *Javascript* untuk membuat penampilan website yang konsisten dengan perintah yang minim[19]

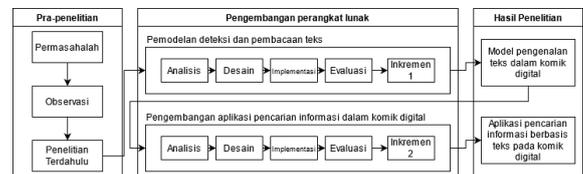
### M. Model Inkremental

Metode inkremental adalah salah satu model pengembangan perangkat lunak atau yang bisa disebut SDLC. Metode ini didasarkan dengan keperluan aplikasi yang dibutuhkan dan dipecah menjadi beberapa fungsi sehingga pengembangan menjadi bertahap. Dengan menggunakan

model ini dapat membantu mengurangi biaya[20].

### N. Proses Penelitian

Hasil dari penelitian ini adalah sebuah aplikasi yang akan bisa membaca teks sebuah gambar dimana teks tersebut bisa dilakukan pencarian terhadap banyak halaman. Pada tahap Inkremental I penulis akan melakukan penelitian terhadap dua langkan OCR yakni *text detection* dan *text recognition*, pada tahap *text detection* penulis akan membuat model learning menggunakan algoritma *YOLOv3*. *Dataset* yang diambil sebagai training menggunakan aset komik *public domain* yang didapat dari *digitalcomicmuseum.com*. Dalam penelitian *text recognition* penulis akan menggunakan *engine OCR tesseract* dan mencoba untuk melakukan proses pre-processing gambar untuk membuat hasil pembacaan yang paling baik



Gambar 2 Proses Penelitian

### U. Iterasi Pertama

Pada tahap inkremental I akan ada dua analisa yang dilakukan. *Text detection* akan dibuat dengan menggunakan model training yang dapat mendeteksi teks dialog dalam sebuah halaman komik, hasil deteksi tersebut lalu akan dimasukkan kedalam proses *text recognition* menggunakan *tesseract*. Sebelum dimasukkan *tesseract* tiap gambar akan diuji beberapa tipe pre-processing yang akan membuat pembacaan lebih baik. Untuk tahap analisis *text detection* akan dibuat pelatihan model menggunakan algoritma *YOLO* dengan framework neural network *darknet*. Data yang digunakan berupa 90% data training dan

10% data testing. Sesuai dengan tahap anotasi gambar, dataset berupa gambar halaman komik dengan metadata berupa koordinat bounding box text dialog yang telah ditentukan secara manual. Pada tahap text detection akan dilakukan analisa beberapa metode pre-processing yang berguna untuk melakukan pembacaan OCR yang paling baik oleh tesseract. Hasil akhir yang diharapkan adalah penggabungan kedua tahap tersebut dan sebuah aplikasi yang dapat membaca, membuat output string dan fungsi pencarian akan hasil output tersebut.

### V. Iterasi Kedua

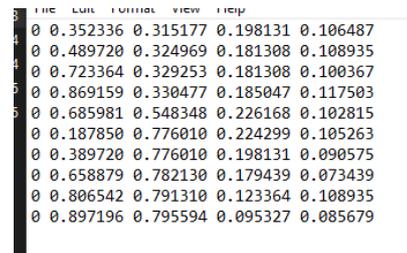
Pada tahap ini difokuskan terhadap pembuatan antarmuka berbasis web untuk bisa mengaplikasikan sistem alur OCR secara praktikal terhadap pengguna berupa GUI (*Graphical User Interface*). Aplikasi berbasis web akan menggunakan *framework* bernama *Flask* yang bisa mengintegrasikan bahasa pemrograman *python* terhadap fungsionalitas aplikasi berbasis web seperti *HTML*, *CSS* dan *Javascript*



Gambar 3 Mockup Halaman Aplikasi

## III. HASIL DAN PEMBAHASAN

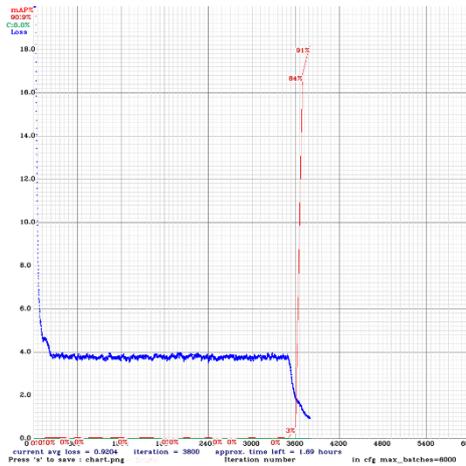
Pada iterasi pertama dilakukan pengambilan koordinat lokasi *bounding box* dari proses anotasi. Koordinat tersebut didapat dalam bentuk file *.txt* yang menunjukkan lokasi koordinat kartesian dalam bentuk angka yang akan diproses dalam training dengan model *algoritma YOLOv3*



Gambar 4 Contoh File Koordinat

Data akhir yang akan digunakan berupa 90% data *training* dan 10% data *testing*. *File* yang dimasukkan kedalam algoritma maka berupa gambar halaman komik dan file koordinat tersebut. Lalu dilakukan penyesuaian parameter *size = 1*, *stride = 1*, *filters = 18*, *activation = linear*, dan *classes 1*

Model dilakukan training dengan algoritma *YOLOv3* menggunakan *framework neural network darknet* dengan *dataset* berupa halaman komik yang telah dianotasi berjumlah total 68 gambar. *Dataset* akhir yang terbuat adalah berupa data train dan testing dengan rasio 9 : 1, 90% data training dan 10% data testing. Hasil dari *training* tersebut dapat terlihat dari output berupa angka *mAP* atau *Mean Average Precision* yang dikeluarkan oleh *framework darknet*. Penulis pertama melakukan training menggunakan parameter default untuk melihat jika ada fungsi optimisasi yang perlu disesuaikan, hasil grafik terhadap *training* pertama dapat dilihat pada Gambar 5 dan hasil deteksi langsung dapat dilihat pada Gambar 6



Gambar 5 Hasil Training



Gambar 6 Hasil Deteksi Teks  
Hasil akhir yang didapatkan adalah nilai *mAP* maksimal 91% dan rata-rata *loss function* yang bernilai 0.9204

Pada tahap kedua akan dilakukan analisa untuk menentukan *pre-processing* yang paling baik untuk *tesseract* dapat melakukan *text recognition*. proses binarisasi gambar akan membuat segmentasi dalam tiap kalimat teks dan membantu *tesseract* membaca kata lebih baik dengan menghitamkan tiap kata dengan jelas. Hal ini dapat dicapai dengan *library OpenCV* yang menyediakan beberapa fungsi binarisasi gambar, penulis akan membandingkan 6 proses binarisasi berbeda yang dapat dibandingkan dengan nilai *confidence score* oleh *tesseract*. Sebelum gambar dibinarisasi, bentuk *input* gambar harus berupa *grayscale* atau abu-abu, hal ini

dikarenakan segmentasi dilakukan dengan menentukan nilai tiap *pixel* antara 1 dan 0 saja atau hitam putih. Hasil *confidence score* pada dilihat pada Gambar 7 dan Gambar 8

```

Hasil Rata-rata Confidence Score Dengan Threshold Binary
block_num
0 -1.000000
1 58.737705
2 54.000000
3 36.000000
4 47.272727
5 55.285714
6 23.000000
Name: conf, dtype: float64
Hasil Rata-rata Confidence Score Dengan Threshold Binary Inverted
block_num
0 -1.000000
1 29.491228
2 44.428571
3 35.571429
4 41.000000
5 23.666667
6 23.000000
Name: conf, dtype: float64
Hasil Rata-rata Confidence Score Dengan Threshold Trunc
block_num
0 -1.000000
1 60.400000
2 38.153846
3 49.571429
4 59.350000
5 46.555556
6 58.000000
7 33.142857
8 37.600000
9 46.541667
10 23.000000

```

Gambar 7 Hasil Nilai Confidence Score

```

Hasil Rata-rata Confidence Score Dengan Threshold To Zero
block_num
0 -1.000000
1 59.483333
2 55.200000
3 38.857143
4 45.363636
5 58.000000
6 23.000000
Name: conf, dtype: float64
Hasil Rata-rata Confidence Score Dengan Threshold To Zero Inverted
block_num
0 -1.000000
1 27.277778
2 30.190476
3 24.000000
4 43.909091
5 5.250000
6 23.000000
Name: conf, dtype: float64
Hasil Rata-rata Confidence Score Dengan Threshold Otsu
block_num
0 -1.000000
1 59.416667
2 50.900000
3 22.200000
4 50.250000
5 30.250000
6 45.727273
7 51.857143
8 23.000000
Name: conf, dtype: float64

```

Gambar 8 Hasil Nilai Confidence Score

Dalam iterasi kedua dilakukan penerapan model deteksi kedalam aplikasi berbasis web menggunakan *framework python Flask* dan *Javascript* sebagai fungsi pencarian. Gambar 9 akan menunjukkan halaman depan web yang menampilkan contoh halaman komik yang sudah diproses terdahulu sebagai percobaan oleh user.

## Searching Text Dengan OCR



Gambar 9 Halaman Depan Aplikasi

Pengguna akan bisa mencoba fitur pencarian dengan membuat input pada kotak pencarian dan halaman dimana kata itu ditemukan akan ditampilkan secara langsung seperti pada gambar 10



Gambar 10 Fitur Pencarian

Pengguna akan bisa mengunggah halaman komiknya sendiri dalam halaman upload seperti pada gambar 11 dan setelah diproses akan ditunjukkan ke dalam halaman pencarian user dimana gambar yang telah diproses akan ditampilkan dan dapat dilakukan pencarian seperti pada Gambar 12



Gambar 11 Halaman Upload



Gambar 12 Halaman Tampil User

## IV. SIMPULAN

Berdasarkan penelitian yang telah dilakukan, penulis dapat menarik beberapa kesimpulan sebagai berikut

1. Penulis berhasil menghasilkan model untuk melakukan deteksi teks dalam halaman komik tetapi pendeteksiannya belum sempurna karena dataset yang dikumpulkan termasuk sedikit dan diambil dari dua sumber komik dengan gaya seni yang relatif sama dengan *font* penulisan yang sama, maka untuk pengaplikasian deteksi yang lebih luas perlu dibuat dataset berisi sumber komik yang lebih beragam
2. Pada training model hasil mAP maksimal yang bisa didapat adalah 91% dengan loss function 0.92
3. Penerapan segmentasi teks dalam gambar menggunakan fungsi thresholding pada halaman komik menghasilkan nilai mean confidence score maksimal 59, nilai tersebut terdapat dari tahap klasifikasi abjad menunjukkan kemungkinan bahwa teks yang telah terdeteksi merupakan teks yang benar
4. Penerapan pembacaan teks pada aplikasi masih tidak sempurna, teks yang bisa terbaca sangat tergantung oleh kualitas gambar. Maka kasus dimana sebuah komik memiliki tulisan tangan dimana abjadnya tergabung akan dinilai lemah oleh tesseract dan menjadi tidak terbaca

## DAFTAR RUJUKAN

- [1] M. Sundaresan and S. Ranjini, "Text extraction from digital English comic image using two blobs extraction method," *Int. Conf. Pattern Recognition, Informatics Med. Eng. PRIME 2012*, pp. 449–452, 2012, doi: 10.1109/ICPRIME.2012.6208388.
- [2] M. S. Karthika Devi, S. Fathima, and R. Baskaran, "CBCS - Comic book cover Synopsis: Generating synopsis of a comic book with unsupervised abstractive dialogue," *Procedia Comput. Sci.*, vol. 172, pp. 701–708, 2020, doi: 10.1016/j.procs.2020.05.100.
- [3] F. Rayar and S. Uchida, "Comic text detection using neural network approach," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11296 LNCS, pp. 672–683, 2019, doi: 10.1007/978-3-030-05716-9\_60.
- [4] R. Smith, "An overview of the tesseract OCR engine," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 2007*, vol. 2, pp. 629–633, doi: 10.1109/ICDAR.2007.4376991.
- [5] "Illustrated Guide to LSTM's and GRU's: A step by step explanation | by Michael Phi | Towards Data Science." <https://towardsdatascience.com/illustrate-d-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21> (accessed Jun. 20, 2021).
- [6] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Comput. Vision, Graph. Image Process.*, vol. 41, no. 2, pp. 233–260, Feb. 1988, doi: 10.1016/0734-189X(88)90022-9.
- [7] L. Caponetti and G. Castellano, "Image Thresholding," 2017, pp. 121–132.
- [8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection." Accessed: Jun. 18, 2021. [Online]. Available: <http://pjreddie.com/yolo/>.
- [9] "Darknet: Open Source Neural Networks in C." <https://pjreddie.com/darknet/> (accessed Jun. 20, 2021).
- [10] S. Carata, R. Mihaescu, E. Barnoviciu, M. Chindea, M. Ghenescu, and V. Ghenescu, "Complete Visualisation, Network Modeling and Training, Web Based Tool, for the Yolo Deep Neural Network Model in the Darknet Framework," *Proc. - 2019 IEEE 15th Int. Conf. Intell. Comput. Commun. Process. ICCP 2019*, pp. 517–523, Sep. 2019, doi: 10.1109/ICCP48234.2019.8959758.
- [11] G. Nagy, T. A. Nartker, and S. V. Rice, "<title>Optical character recognition: an illustrated guide to the frontier</title>," *Doc. Recognit. Retr. VII*, vol. 3967, no. December 1999, pp. 58–69, 1999, doi: 10.1117/12.373511.
- [12] C. R. Severance, "Python for Everybody | Trinket," p. 245, 2018, [Online]. Available: <https://books.trinket.io/pfe/index.html>.
- [13] M. Naveenkumar and V. Ayyasamy, "OpenCV for Computer Vision Applications," *Proc. Natl. Conf. Big Data Cloud Comput.*, no. March 2015, pp. 52–56, 2016, [Online]. Available: [https://www.researchgate.net/publication/301590571\\_OpenCV\\_for\\_Computer\\_Vision\\_Applications](https://www.researchgate.net/publication/301590571_OpenCV_for_Computer_Vision_Applications).
- [14] F. Armash Aslam, H. Nabeel Mohammed Jummal Musab Mohd Munir Murade Aaraf Gulamgaus, and P. S. Lokhande Assistant Professor, "Efficient Way Of Web Development Using Python And Flask," *Int. J. Adv. Res. Comput. Sci.*, vol. 6, no. 2, Accessed: Aug. 14, 2021. [Online]. Available: [www.ijarcs.info](http://www.ijarcs.info).
- [15] Miguel Grinberg, *Flask web development*, vol. 53, no. 9. 2013.
- [16] S. Gupta, G. Kaiser, D. Neistadt, and P. Grimm, "DOM-based content extraction of HTML documents," *Proc. 12th Int. Conf. World Wide Web, WWW 2003*, pp. 207–214, 2003, doi: 10.1145/775152.775182.
- [17] "What Is HTML? Hypertext Markup Language Basics for Beginners." <https://www.hostinger.com/tutorials/what-is-html> (accessed Jun. 20, 2021).
- [18] P. Genevès, N. Layaida, and V. Quint, "On the analysis of cascading style sheets," *WWW'12 - Proc. 21st Annu. Conf. World Wide Web*, pp. 809–818, 2012, doi: 10.1145/2187836.2187946.
- [19] "Bootstrap · The most popular HTML, CSS, and JS library in the world." <https://getbootstrap.com/> (accessed Jun. 21, 2021).
- [20] K. Petersen and C. Wohlin, "The effect of moving from a plan-driven to an incremental software development approach with agile practices," *Empir. Softw. Eng. 2010 156*, vol. 15, no. 6, pp. 654–693, Jul. 2010, doi: 10.1007/S10664-010-9136-6.