LayoutParser: Toolkit Terpadu untuk Deep Analisis Gambar Dokumen Berbasis Pembelajaran

Ze Jiang Shen[1] ( ), Ruo Chen Zhang [2], Melissa Dell [3], Benyamin Charles Germain Lee[4], Jacob Carlson [3], dan Weining Li [5]

[1] Institut Allen untuk AI
shannons@allenai.org
[2] Universitas Brown
ruochen zhang@brown.edu
[3] Universitas Harvard
{melissadell,jacob carlson }@fas.harvard.edu Universitas
[4] Washington bcgl@cs.washington.edu
Universitas Waterloo
[5]
w422li@uwaterloo.ca

Abstrak. Kemajuan terbaru dalam analisis gambar dokumen (DIA) terutama didorong oleh penerapan jaringan saraf. Idealnya, hasil penelitian dapat dengan mudah digunakan dalam produksi dan diperluas untuk penyelidikan lebih lanjut. Namun, berbagai faktor seperti basis kode yang diatur secara longgar dan konfigurasi model yang canggih mempersulit penggunaan kembali inovasi penting dengan mudah oleh khalayak luas. Meskipun ada upaya berkelanjutan untuk meningkatkan penggunaan kembali dan menyederhanakan pengembangan model pembelajaran mendalam (DL) dalam disiplin ilmu seperti pemrosesan bahasa alami dan visi komputer, tidak satu pun dari mereka yang dioptimalkan untuk tantangan dalam domain DIA. Ini merupakan kesenjangan besar dalam perangkat yang ada, karena DIA merupakan pusat penelitian akademik di berbagai disiplin ilmu sosial dan humaniora. Makalah ini memperkenalkan LayoutParser, perpustakaan sumber terbuka untuk merampingkan penggunaan DL dalam penelitian dan aplikasi DIA. Pustaka LayoutParser inti hadir dengan serangkaian antarmuka sederhana dan intuitif untuk menerapkan dan menyesuaikan model DL untuk deteksi tata letak, pengenalan karakter, dan banyak tugas pemrosesan dokumen lainnya. Untuk mempromosikan ekstensibilitas, LayoutParser juga menggabungkan platform komunitas untuk berbagi model pra-pelatihan dan pipeline digitalisasi dokumen lengkap. Kami mendemonstrasikan bahwa LayoutParser berguna untuk pipeline digitalisasi skala besar dan ringan dalam kasus penggunaan nyata.
Perpustakaan tersedia untuk umum di https://layout-parser.github.io .

Kata Kunci: Analisis Gambar Dokumen · Pembelajaran Mendalam · Analisis Tata Letak · Pengenalan Karakter · Pustaka Open Source · Toolkit.

1. Perkenalan

Pendekatan berbasis Deep Learning (DL) adalah yang paling canggih untuk berbagai tugas analisis gambar dokumen (DIA) termasuk klasifikasi gambar dokumen [11 ,

37], deteksi tata letak [38, 22], deteksi tabel [26], dan deteksi teks adegan [4].
Kerangka kerja berbasis pembelajaran umum secara dramatis mengurangi kebutuhan akan spesifikasi manual dari aturan yang rumit, yang merupakan status quo dengan metode tradisional. DL memiliki potensi untuk mentransformasi jalur pipa DIA dan menguntungkan spektrum yang luas dari proyek digitalisasi dokumen berskala besar.

Namun, ada beberapa kesulitan praktis untuk memanfaatkan kemajuan terbaru dalam metode berbasis DL: 1) Model DL terkenal berbelit-belit untuk digunakan kembali dan diperluas. Model yang ada dikembangkan menggunakan kerangka kerja yang berbeda seperti TensorFlow [1] atau PyTorch [24], dan parameter tingkat tinggi dapat disamarkan dengan detail implementasi [8]. Ini bisa menjadi pengalaman yang memakan waktu dan membuat frustrasi untuk men-debug, mereproduksi, dan mengadaptasi model yang ada untuk DIA, dan banyak peneliti yang paling diuntungkan dari penggunaan metode ini tidak memiliki latar belakang teknis untuk menerapkannya dari awal. 2) Gambar dokumen berisi pola yang beragam dan berbeda di seluruh domain, dan pelatihan yang disesuaikan seringkali diperlukan untuk mencapai akurasi deteksi yang diinginkan. Saat ini tidak ada infrastruktur lengkap untuk dengan mudah mengkurasi kumpulan data gambar dokumen target dan menyempurnakan atau melatih ulang model. 3) DIA biasanya membutuhkan urutan model dan pemrosesan lainnya untuk mendapatkan keluaran akhir. Seringkali tim peneliti menggunakan model DL dan kemudian melakukan analisis dokumen lebih lanjut dalam proses terpisah, dan jalur pipa ini tidak didokumentasikan di lokasi pusat mana pun (dan seringkali tidak didokumentasikan sama sekali). Hal ini mempersulit tim peneliti untuk mempelajari tentang penerapan jalur pipa penuh dan mengarahkan mereka untuk menginvestasikan sumber daya yang signifikan dalam menemukan kembali roda DIA.

LayoutParser menyediakan toolkit terpadu untuk mendukung analisis dan pemrosesan gambar dokumen berbasis DL. Untuk mengatasi tantangan yang disebutkan di atas, LayoutParser dibuat dengan komponen berikut:

1. Toolkit siap pakai untuk menerapkan model DL untuk deteksi tata letak, karakter pengakuan, dan tugas DIA lainnya (Bagian 3)
2. Repositori kaya model jaringan saraf pra-terlatih (Model Zoo) yang mendasari penggunaan siap pakai 3. Alat komprehensif untuk anotasi data gambar dokumen yang efisien dan penyetelan model untuk mendukung berbagai tingkat penyesuaian 4. Hub model DL dan platform komunitas untuk kemudahan berbagi, distribusi, dan diskusi model dan jalur pipa DIA, untuk mempromosikan penggunaan kembali, reproduktifitas, dan ekstensibilitas (Bagian 4)

Pustaka mengimplementasikan API Python yang sederhana dan intuitif tanpa mengorbankan kemampuan generalisasi dan keserbagunaan, dan dapat dipasang dengan mudah melalui pip. Fungsinya yang mudah digunakan untuk menangani data gambar dokumen dapat diintegrasikan secara mulus dengan jalur pipa DIA yang ada. Dengan dokumentasi terperinci dan tutorial yang dikuratori dengan hati-hati, kami berharap alat ini akan bermanfaat bagi berbagai pengguna akhir, dan akan mengarah pada kemajuan aplikasi baik dalam penelitian industri maupun akademik.

LayoutParser selaras dengan upaya terbaru untuk meningkatkan penggunaan kembali model DL dalam disiplin lain seperti pemrosesan bahasa alami [8, 34] dan visi komputer [35], tetapi dengan fokus pada tantangan unik di DIA. Kami menunjukkan LayoutParser dapat diterapkan dalam proyek digitalisasi yang canggih dan berskala besar

yang membutuhkan presisi, efisiensi, dan ketangguhan, serta tugas pemrosesan dokumen yang sederhana dan ringan yang berfokus pada kemanjuran dan fleksibilitas (Bagian 5). LayoutParser sedang dipelihara secara aktif, dan dukungan untuk model pembelajaran yang lebih dalam dan metode baru dalam metode analisis tata letak berbasis teks [37, 34] direncanakan .

Sisa kertas ini disusun sebagai berikut. Bagian 2 memberikan ikhtisar tentang pekerjaan terkait. Pustaka LayoutParser inti , DL Model Zoo, dan pelatihan model yang disesuaikan dijelaskan di Bagian 3, dan hub model DL serta platform komunitas dirinci di Bagian 4. Bagian 5 menunjukkan dua contoh bagaimana LayoutParser dapat digunakan dalam proyek DIA praktis, dan Bagian 6 menyimpulkan.

2 Pekerjaan Terkait

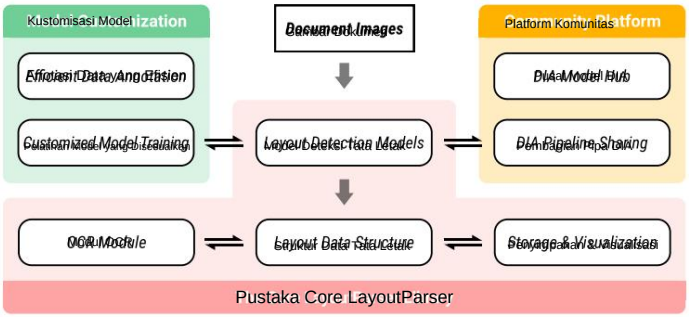
Baru-baru ini, berbagai model dan kumpulan data DL telah dikembangkan untuk tugas analisis tata letak. dhSegment [22] menggunakan jaringan konvolusional penuh [20] untuk tugas segmentasi pada dokumen historis. Metode berbasis deteksi objek seperti Faster R-CNN [28] dan Mask R-CNN [12] digunakan untuk mengidentifikasi elemen dokumen [38] dan mendeteksi tabel [30, 26]. Baru-baru ini, Graph Neural Networks [29] juga telah digunakan dalam deteksi tabel [27]. Namun, model ini biasanya diimplementasikan secara individual dan tidak ada kerangka kerja terpadu untuk memuat dan menggunakan model tersebut.

Ada lonjakan minat dalam membuat alat sumber terbuka untuk pemrosesan gambar dokumen: pencarian analisis gambar dokumen di Github menghasilkan 5 juta potongan kode yang relevan [6] ; namun kebanyakan dari mereka hanya mengandung tugas yang terbatas. Beberapa sebelumnya yang paling dekat dengan pekerjaan kami adalah proyek OCR-D [7] , yang juga mencoba membuat pipeline lengkap untuk DIA, namun tidak dapat digunakan dengan platform yang dokumen sejarah, dan tidak memberikan dukungan untuk model DL terbaru.

Proyek DocumentLayoutAnalysis[8] berfokus pada pemrosesan dokumen PDF digital melalui analisis data PDF yang disimpan . Repositori seperti DeepLayout[9] dan Detectron2-PubLayNet[10] adalah model pembelajaran mendalam individual yang dilatih pada kumpulan data analisis tata letak tanpa dukungan untuk pipeline DIA lengkap. Platform Analisis dan Eksploitasi Dokumen (DAE) [15] dan proyek DeepDIVA [2] bertujuan untuk meningkatkan reproduktifitas metode DIA (atau model DL), namun tidak dipelihara secara aktif. Mesin OCR seperti Tesseract [14], easyOCR[11] dan paddleOCR[12] biasanya tidak hadir dengan fungsionalitas komprehensif untuk tugas DIA lainnya seperti analisis tata letak.

Beberapa tahun terakhir juga terlihat banyak upaya untuk membuat perpustakaan untuk mempromosikan reproduktifitas dan usabilitas di bidang DL. Perpustakaan seperti Dectectron2 [35],

[6] Angka yang ditampilkan diperoleh dengan menentukan jenis pencarian sebagai 'kode'.
[7] https://ocr-d.de/en/about
[8] https://github.com/BobLd/DocumentLayoutAnalysis
[9] https://github.com/leonlulu/DeepLayout
[10] https://github.com/hpanwar08/detectron2
[11] https:// github.com/JaidedAI/EasyOCR
[12] https://github.com/PaddlePaddle/PaddleOCR



Gambar 1: Keseluruhan arsitektur LayoutParser. Untuk gambar dokumen input, pustaka LayoutParser inti menyediakan seperangkat alat siap pakai untuk deteksi tata letak, OCR, visualisasi, dan penyimpanan, yang didukung oleh struktur data tata letak yang dirancang dengan cermat. LayoutParser juga mendukung penyesuaian tingkat tinggi melalui anotasi tata letak yang efisien dan fungsi pelatihan model. Ini meningkatkan akurasi model pada sampel target. Platform komunitas memungkinkan berbagi model DIA dengan mudah dan seluruh saluran digitalisasi untuk mempromosikan penggunaan kembali dan reproduktifitas. Kumpulan dokumentasi terperinci, tutorial, dan proyek contoh membuat LayoutParser mudah dipelajari dan digunakan.

AllenNLP [8] dan transformer [34] telah menyediakan komunitas dengan dukungan berbasis DL lengkap untuk mengembangkan dan menerapkan model untuk visi komputer umum dan masalah pemrosesan bahasa alami. LayoutParser, di sisi lain , berspesialisasi secara khusus dalam tugas DIA. LayoutParser juga dilengkapi dengan platform komunitas yang terinspirasi oleh hub model yang sudah mapan seperti Torch Hub [23] dan TensorFlow Hub [1]. Hal ini memungkinkan pembagian model prapelatihan serta alur pemrosesan dokumen lengkap yang unik untuk tugas DIA.

Ada berbagai koleksi data dokumen untuk memfasilitasi pengembangan model DL. Beberapa contoh termasuk PRImA [3](tata letak majalah), PubLayNet [38](tata letak makalah akademik), Bank Meja [18](tabel dalam makalah akademik), Kumpulan Data Navigator Surat Kabar [16, 17](tata letak gambar surat kabar) dan HJDataset [31](tata letak dokumen sejarah Jepang). Spektrum model yang dilatih pada kumpulan data ini saat ini tersedia di kebun binatang model LayoutParser untuk mendukung kasus penggunaan yang berbeda.

3 Pustaka Core LayoutParser

Inti dari LayoutParser adalah toolkit siap pakai yang merampingkan analisis gambar dokumen berbasis DL. Lima komponen mendukung antarmuka sederhana dengan fungsionalitas komprehensif: 1) Model deteksi tata letak memungkinkan penggunaan model DL pra-terlatih atau mandiri untuk deteksi tata letak hanya dengan empat baris kode. 2) Informasi tata letak yang terdeteksi disimpan dengan hati-hati direkayasa

Tabel 1: Model deteksi tata letak saat ini di kebun binatang model LayoutParser

Himpunan data	Model Dasar[1]	Model Besar	Catatan
PubLayNet [38]	F / M	M	Tata letak dokumen ilmiah modern
PRIMA [3]	M	-	Tata letak majalah modern yang dipindai dan laporan ilmiah
Koran [17]	F	-	Tata letak surat kabar AS yang dipindai dari abad ke-20
Bank Meja [18]	F	F	Wilayah tabel pada dokumen ilmiah dan bisnis modern
HJDataset [31]	F / M	-	Tata letak dokumen sejarah Jepang

[1] Untuk setiap kumpulan data, kami melatih beberapa model dengan ukuran berbeda untuk kebutuhan berbeda (trade-off antara akurasi vs. biaya komputasi). Untuk "model dasar" dan "model besar", kami merujuk masing-masing menggunakan tulang punggung ResNet 50 atau ResNet 101 [13] . Seseorang dapat melatih model arsitektur yang berbeda, seperti Faster R-CNN [28] (F) dan Mask R-CNN [12] (M). Misalnya, F di kolom Model Besar menunjukkan model R-CNN Lebih Cepat yang dilatih menggunakan tulang punggung ResNet 101. Platform dipertahankan dan sejumlah penambahan akan dilakukan pada kebun binatang model dalam beberapa bulan mendatang.

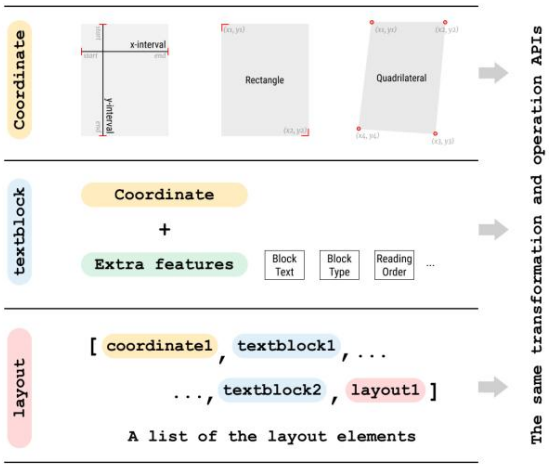
tata letak struktur data, yang dioptimalkan untuk efisiensi dan keserbagunaan. 3) Bila perlu, pengguna dapat menggunakan model OCR yang sudah ada atau disesuaikan melalui API terpadu yang disediakan dalam modul OCR. 4) LayoutParser dilengkapi dengan serangkaian fungsi utilitas untuk visualisasi dan penyimpanan data tata letak. 5) LayoutParser juga sangat dapat disesuaikan, melalui integrasinya dengan fungsi untuk anotasi data tata letak dan pelatihan model. Kami sekarang memberikan deskripsi terperinci untuk setiap komponen.

3.1 Model Deteksi Tata Letak

Di LayoutParser, model tata letak mengambil gambar dokumen sebagai input dan menghasilkan daftar kotak persegi panjang untuk wilayah konten target. Berbeda dari metode tradisional, ini bergantung pada jaringan saraf konvolusional yang dalam daripada aturan yang dikuratori secara manual untuk mengidentifikasi wilayah konten. Ini diformulasikan sebagai masalah deteksi objek dan model canggih seperti Faster R-CNN [28] dan Mask R-CNN [12] digunakan. Ini menghasilkan hasil prediksi dengan akurasi tinggi dan memungkinkan untuk membangun antarmuka umum yang ringkas untuk deteksi tata letak. LayoutParser, dibangun di atas Detectron2 [35], menyediakan API minimal yang dapat melakukan deteksi tata letak hanya dengan empat baris kode di Python:

```
impor layoutparser sebagai lp image =
cv2 . imread (" image_file ") # muat gambar model = lp . Detectron2LayoutModel
( "lp :// PubLayNet / fast_rcnn_R_50_FPN_3x / config ") layout = model .
        deteksi (gambar)
```

LayoutParser menyediakan banyak bobot model terlatih menggunakan berbagai kumpulan data yang mencakup berbagai bahasa, periode waktu, dan jenis dokumen. Karena pergeseran domain [7], kinerja prediksi dapat turun secara signifikan ketika model diterapkan pada sampel target yang berbeda secara signifikan dari dataset pelatihan. Karena struktur dan tata letak dokumen sangat bervariasi di domain yang berbeda, penting untuk memilih model yang dilatih pada kumpulan data yang mirip dengan sampel uji. Sintaks semantik digunakan untuk menginisialisasi bobot model di LayoutParser, menggunakan nama set data dan nama model lp://<namaset-data>/<nama-arsitektur-model>.



Gambar 2: Hubungan antara ketiga jenis struktur data tata letak.
Koordinat mendukung tiga jenis variasi; TextBlock terdiri dari informasi koordinat dan fitur tambahan seperti blok teks, jenis, dan perintah membaca; objek Layout adalah daftar semua kemungkinan elemen layout, termasuk objek Layout lainnya . Semuanya mendukung rangkaian API transformasi dan operasi yang sama untuk fleksibilitas maksimum.

Ditunjukkan pada Tabel 1, LayoutParser saat ini menghosting 9 model pra-pelatihan yang dilatih pada 5 kumpulan data berbeda. Deskripsi set data pelatihan disediakan bersama dengan model yang dilatih sehingga pengguna dapat dengan cepat mengidentifikasi model yang paling cocok untuk tugas mereka. Selain itu, ketika model seperti itu tidak tersedia, LayoutParser juga mendukung pelatihan model tata letak yang disesuaikan dan pembagian model oleh komunitas (dirinci dalam Bagian 3.5).

3.2 Tata Letak Struktur Data

Fitur penting LayoutParser adalah penerapan rangkaian struktur data dan operasi yang dapat digunakan untuk memproses dan memanipulasi elemen tata letak secara efisien. Dalam pipa analisis gambar dokumen, berbagai pasca-pemrosesan pada keluaran model analisis tata letak biasanya diperlukan untuk mendapatkan keluaran akhir. Secara tradisional, ini memerlukan ekspor keluaran model DL dan kemudian memuat hasilnya ke jaringan pipa lainnya. Semua output model dari LayoutParser akan disimpan dalam tipe data yang direkayasa dengan hati-hati yang dioptimalkan untuk pemrosesan lebih lanjut, yang memungkinkan untuk membangun pipeline digitalisasi dokumen ujung ke ujung dalam LayoutParser. Ada tiga komponen utama dalam struktur data, yaitu sistem Koordinat , TextBlock, dan Layout. Mereka memberikan tingkat abstraksi yang berbeda untuk data tata letak, dan satu set API didukung untuk transformasi atau operasi pada kelas-kelas ini.

Koordinat adalah landasan untuk menyimpan informasi tata letak. Saat ini, tiga jenis struktur data Koordinat disediakan di LayoutParser, yang ditunjukkan pada Gambar 2. Interval dan Persegi Panjang adalah jenis data yang paling umum dan mendukung penetapan wilayah 1D atau 2D dalam dokumen. Mereka diparameterisasi dengan 2 dan 4 parameter. Kelas Segiempat juga diimplementasikan untuk mendukung representasi yang lebih umum dari wilayah persegi panjang saat dokumen miring atau terdistorsi, di mana 4 titik sudut dapat ditentukan dan didukung total 8 derajat kebebasan. Kumpulan transformasi yang luas seperti shift, pad, dan scale, dan operasi seperti intersect, union, dan is_in, didukung untuk kelas-kelas ini. Khususnya, adalah umum untuk memisahkan segmen gambar dan menganalisisnya satu per satu. LayoutParser memberikan dukungan penuh untuk skenario ini melalui operasi pemotongan gambar crop_image dan transformasi koordinat seperti relative_to dan condition_on yang mengubah koordinat ke dan dari representasi relatifnya. Kami merujuk pembaca ke Tabel 2 untuk penjelasan lebih rinci tentang operasi ini13 .

Berdasarkan Koordinat, kami mengimplementasikan kelas TextBlock yang menyimpan fitur posisi dan tambahan dari elemen tata letak individual. Ini juga mendukung menentukan perintah membaca melalui pengaturan bidang induk ke indeks objek induk . Kelas Tata Letak dibangun yang mengambil daftar TextBlocks dan mendukung pemrosesan elemen dalam batch. Tata letak juga dapat disarangkan untuk mendukung struktur tata letak hierarkis. Mereka mendukung operasi dan transformasi yang sama seperti kelas Koordinasi, meminimalkan upaya pembelajaran dan penerapan.

3.3 OCR

LayoutParser menyediakan antarmuka terpadu untuk alat OCR yang ada. Meskipun ada banyak alat OCR yang tersedia, mereka biasanya dikonfigurasi secara berbeda dengan API atau protokol yang berbeda untuk menggunakannya. Mungkin tidak efisien untuk menambahkan alat OCR baru ke dalam pipa yang ada, dan sulit untuk membuat perbandingan langsung di antara alat yang tersedia untuk menemukan opsi terbaik untuk proyek tertentu. Untuk tujuan ini, LayoutParser membuat serangkaian pembungkus di antara mesin OCR yang ada, dan menyediakan sintaks yang hampir sama untuk menggunakannya. Ini mendukung gaya plug-and-play menggunakan mesin OCR, membuatnya mudah untuk beralih, mengevaluasi, dan membandingkan berbagai modul OCR:

```
ocr_agent = lp . TesseractAgen ()
# Dapat dengan mudah dialihkan ke perangkat lunak OCR lainnya
token = ocr_agent . deteksi (gambar)
```

Keluaran OCR juga akan disimpan dalam struktur data tata letak yang disebutkan di atas dan dapat digabungkan dengan mulus ke dalam pipa digitalisasi. Saat ini LayoutParser mendukung mesin OCR Tesseract dan Google Cloud Vision .

LayoutParser juga dilengkapi dengan model OCR CNN-RNN berbasis DL [6] yang dilatih dengan hilangnya Connectionist Temporal Classification (CTC) [10]. Ini dapat digunakan seperti modul OCR lainnya, dan dapat dengan mudah dilatih pada kumpulan data yang disesuaikan.

13 Ini juga tersedia di halaman dokumentasi LayoutParser.

Tabel 2: Semua operasi yang didukung oleh elemen tata letak. API yang sama didukung di berbagai kelas elemen tata letak termasuk tipe Koordinat , TextBlock, dan Tata Letak.

Nama Operasi	Keterangan
block.pad(atas, bawah, kanan, kiri)	Perbesar blok saat ini sesuai dengan input
blok.skala(fx, fy)	Skala blok saat ini diberikan rasio dalam arah x dan y
blok.shift(dx, dy)	Pindahkan blok saat ini dengan shift jarak dalam arah x dan y
blok1.ada di(blok2)	Apakah blok1 ada di dalam blok2
blok1.persimpangan(blok2)	Kembalikan wilayah persimpangan blok1 dan blok2. Jenis koordinat akan ditentukan berdasarkan input.
blok1.union(blok2)	Kembalikan wilayah gabungan dari blok1 dan blok2. Jenis koordinat akan ditentukan berdasarkan input.
block1.relatif ke (block2)	Ubah koordinat absolut dari blok1 menjadi koordinat relatif ke blok2
block1.kondisi aktif(block2)	Hitung koordinat absolut dari blok1 yang diberikan koordinat absolut blok2 kanvas
blokir. potong gambar (gambar)	Dapatkan segmen gambar di wilayah blok

3.4 Penyimpanan dan visualisasi

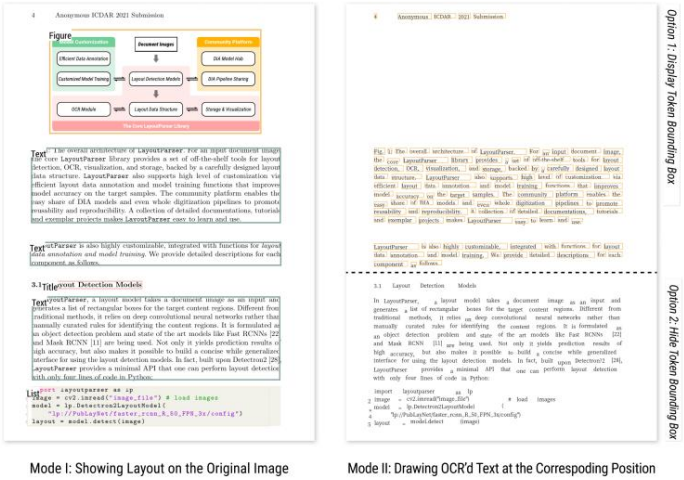
Tujuan akhir DIA adalah mengubah data dokumen berbasis gambar menjadi database terstruktur. LayoutParser mendukung ekspor data tata letak ke format yang berbeda seperti JSON, csv, dan akan menambahkan dukungan untuk format XML METS/ ALTO. Itu juga dapat memuat kumpulan data dari format khusus analisis tata letak seperti COCO [38] dan Format Halaman [25] untuk model tata letak pelatihan (Bagian 3.5).

Visualisasi hasil deteksi tata letak sangat penting untuk presentasi dan debugging. LayoutParser dibangun dengan API terintegrasi untuk menampilkan informasi tata letak bersama dengan gambar dokumen asli. Ditunjukkan pada Gambar 3, ini memungkinkan penyajian data tata letak dengan informasi meta yang kaya dan fitur dalam mode yang berbeda. Informasi lebih rinci dapat ditemukan di halaman dokumentasi LayoutParser online.

3.5 Pelatihan Model yang Disesuaikan

Selain pustaka siap pakai, LayoutParser juga sangat dapat disesuaikan dengan dukungan untuk tugas analisis dokumen yang sangat unik dan menantang. Gambar dokumen target bisa sangat berbeda dari kumpulan data yang ada untuk melatih model tata letak, yang menyebabkan akurasi deteksi tata letak rendah. Data pelatihan

[14] https://altoxml.github.io

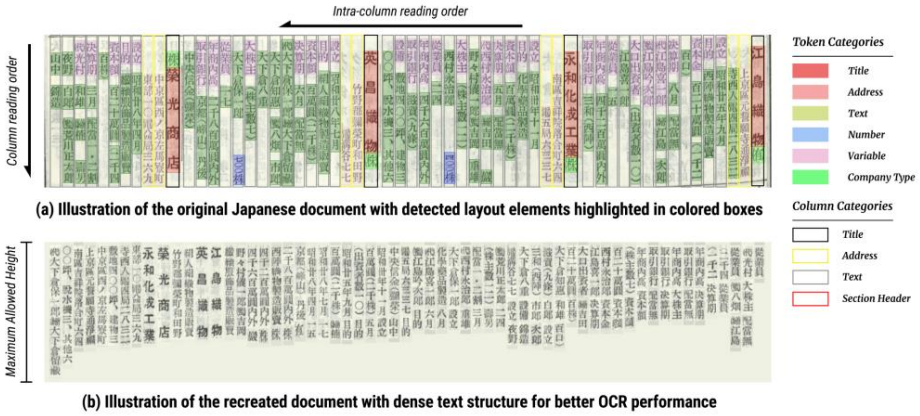


Gambar 3: Deteksi tata letak dan visualisasi hasil OCR yang dihasilkan oleh API LayoutParser . Mode I secara langsung menghamparkan kotak dan kategori pembatas wilayah tata letak di atas gambar asli. Mode II membuat ulang dokumen asli dengan menggambar teks OCR pada posisi yang sesuai di kanvas gambar. Dalam gambar ini, token di wilayah tekstual difilter menggunakan API, lalu ditampilkan.

juga bisa sangat sensitif dan tidak dapat dibagikan secara publik. Untuk mengatasi tantangan ini, LayoutParser dibuat dengan fitur yang kaya untuk anotasi data yang efisien dan pelatihan model yang disesuaikan.

LayoutParser menggabungkan toolkit yang dioptimalkan untuk tata letak dokumen anotasi menggunakan pembelajaran aktif tingkat objek [32]. Dengan bantuan model deteksi tata letak yang dilatih bersama dengan pelabelan, hanya objek tata letak yang paling penting dalam setiap gambar, bukan keseluruhan gambar, yang diperlukan untuk pelabelan. Wilayah lainnya secara otomatis dianotasi dengan prediksi keyakinan tinggi dari model deteksi tata letak. Ini memungkinkan kumpulan data tata letak dibuat lebih efisien dengan hanya sekitar 60% dari anggaran pelabelan.

Setelah set data pelatihan diseleksi, LayoutParser mendukung berbagai mode untuk melatih model tata letak. Penyesuaian halus dapat digunakan untuk melatih model pada set data kecil yang baru diberi label dengan menginisialisasi model dengan bobot yang telah dilatih sebelumnya. Pelatihan dari awal dapat membantu ketika kumpulan data sumber dan target berbeda secara signifikan dan kumpulan pelatihan yang besar tersedia. Namun, seperti yang disarankan dalam karya Studer et al. [33], memuat bobot yang telah dilatih sebelumnya pada kumpulan data skala besar seperti ImageNet [5], bahkan dari domain yang sama sekali berbeda, masih dapat meningkatkan kinerja model. Melalui API terintegrasi yang disediakan oleh LayoutParser, pengguna dapat dengan mudah membandingkan performa model pada kumpulan data tolok ukur.



Gambar 4: Ilustrasi (a) dokumen sejarah asli Jepang dengan hasil deteksi tata letak dan (b) versi gambar dokumen yang dibuat ulang yang menghasilkan pengenalan pengenalan karakter yang jauh lebih baik. Algoritme reorganisasi mengatur ulang token berdasarkan kotak pembatas yang terdeteksi dengan ketinggian maksimum yang diizinkan.

4 Platform Komunitas LayoutParser

Fokus lain dari LayoutParser adalah mempromosikan penggunaan kembali model deteksi tata letak dan pipeline digitalisasi penuh. Mirip dengan banyak pustaka deep learning yang ada, LayoutParser dilengkapi dengan hub model komunitas untuk mendistribusikan model tata letak. Pengguna akhir dapat mengunggah model yang dilatih sendiri ke hub model, dan model ini dapat dimuat ke antarmuka yang serupa dengan model pra-terlatih LayoutParser yang tersedia saat ini. Sebagai contoh, model yang dilatih pada dataset News Navigator [17] telah dimasukkan ke dalam model hub.

Di luar model DL, LayoutParser juga mempromosikan pembagian seluruh pipeline digitalisasi dokumen. Misalnya, terkadang pipeline memerlukan kombinasi beberapa model DL untuk mencapai akurasi yang lebih baik. Saat ini, pipeline sebagian besar dijelaskan dalam makalah akademis dan implementasinya seringkali tidak tersedia untuk umum. Untuk tujuan ini, platform komunitas LayoutParser juga memungkinkan berbagi pipa tata letak untuk mempromosikan diskusi dan penggunaan kembali teknik. Untuk setiap saluran bersama, ini memiliki halaman proyek khusus, dengan tautan ke kode sumber, dokumentasi, dan garis besar pendekatan. Panel diskusi disediakan untuk bertukar ide. Dikombinasikan dengan pustaka LayoutParser inti , pengguna dapat dengan mudah membuat komponen yang dapat digunakan kembali berdasarkan saluran pipa bersama dan menerapkannya untuk memecahkan masalah unik mereka.

5 Kasus Penggunaan

Tujuan inti dari LayoutParser adalah untuk mempermudah pembuatan pipeline digitalisasi dokumen berskala besar dan ringan. Pemrosesan dokumen berskala besar

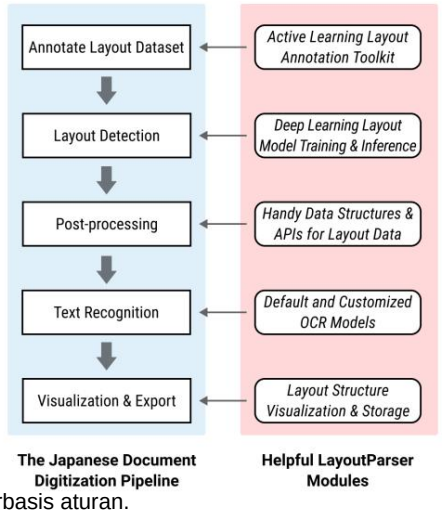
berfokus pada presisi, efisiensi, dan ketahanan. Dokumen target mungkin memiliki struktur yang rumit, dan mungkin memerlukan pelatihan beberapa model deteksi tata letak untuk mencapai akurasi yang optimal. Pipa ringan dibuat untuk dokumen yang relatif sederhana, dengan penekanan pada kemudahan, kecepatan, dan fleksibilitas pengembangan. Idealnya seseorang hanya perlu menggunakan sumber daya yang ada, dan pelatihan model harus dihindari. Melalui dua contoh proyek, kami menunjukkan bagaimana para praktisi di bidang akademisi dan industri dapat dengan mudah membangun pipeline tersebut menggunakan LayoutParser dan mengekstrak data dokumen terstruktur berkualitas tinggi untuk tugas hilir mereka. Kode sumber untuk proyek ini akan tersedia untuk umum di hub komunitas LayoutParser.

5.1 Alur Digitalisasi Dokumen Historis yang Komprehensif

Digitalisasi dokumen sejarah dapat membuka data berharga yang dapat menjelaskan banyak pertanyaan sosial, ekonomi, dan sejarah yang penting. Namun karena kebisingan pemindaian, pemakaian halaman, dan prevalensi struktur tata letak yang rumit, memperoleh representasi terstruktur dari pemindaian dokumen historis seringkali sangat rumit.

Dalam contoh ini, LayoutParser digunakan untuk mengembangkan alur yang komprehensif, ditunjukkan pada Gambar 5, untuk menghasilkan data terstruktur berkualitas tinggi dari tabel keuangan perusahaan Jepang historis dengan tata letak yang rumit. Pipeline menerapkan dua model tata letak untuk mengidentifikasi berbagai tingkat struktur dokumen dan dua mesin OCR yang disesuaikan untuk akurasi pengenalan karakter yang dioptimalkan.

Seperti yang ditunjukkan pada Gambar 4 (a), dokumen berisi kolom teks yang ditulis secara vertikal 15, gaya umum dalam bahasa Jepang. Karena kebisingan pemindaian dan teknologi pencetakan kuno, kolom dapat miring atau memiliki lebar yang bervariasi, dan karenanya tidak dapat dengan mudah diidentifikasi melalui metode berbasis aturan. Dalam setiap kolom, kata dipisahkan oleh spasi putih dengan ukuran variabel, dan posisi vertikal objek dapat menjadi indikator jenis tata letaknya .



Gambar 5: Ilustrasi bagaimana LayoutParser membantu pipa digitalisasi dokumen historis .

15 Halaman dokumen terdiri dari delapan baris seperti ini. Untuk mempermudah, kami melewatkan diskusi segmentasi baris dan mengarahkan pembaca ke kode sumber jika tersedia.

Untuk menguraikan struktur tata
letak yang rumit, dua model deteksi objek telah dilatih untuk mengenali masing-masing kolom dan token. Satu set pelatihan kecil (400 gambar dengan masing-masing sekitar 100 anotasi) dikurasi melalui alat anotasi berbasis pembelajaran aktif [32] di LayoutParser. Model belajar mengidentifikasi kategori dan wilayah untuk setiap token atau kolom melalui fitur visualnya yang berbeda. Struktur data tata letak memungkinkan pengelompokan token yang mudah dalam setiap kolom, dan mengatur ulang kolom untuk mencapai urutan pembacaan yang benar berdasarkan posisi horizontal. Kesalahan diidentifikasi dan diperbaiki melalui pemeriksaan konsistensi prediksi model. Oleh karena itu, meskipun dilatih pada dataset kecil, pipeline mencapai tingkat akurasi deteksi tata letak yang tinggi: mencapai skor 96,97 AP [19] di 5 kategori untuk model deteksi kolom, dan 89,23 AP di 4 kategori untuk model deteksi token .

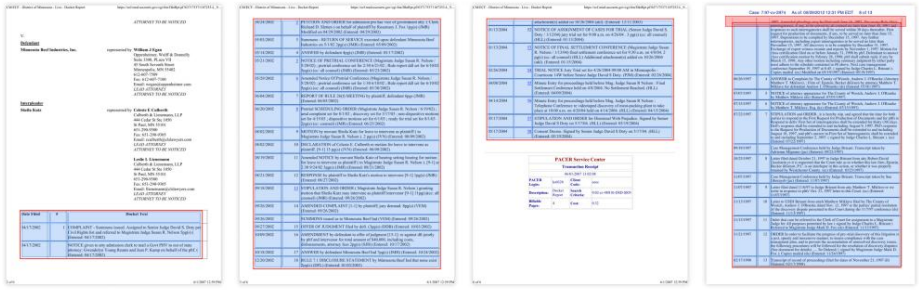
Kombinasi metode pengenalan karakter dikembangkan untuk mengatasi tantangan unik dalam dokumen ini. Dalam percobaan kami, kami menemukan bahwa jarak yang tidak teratur antara token menyebabkan tingkat penarikan pengenalan karakter yang rendah, sedangkan model OCR yang ada cenderung bekerja lebih baik pada teks yang tersusun rapat. Untuk mengatasi tantangan ini, kami membuat algoritme penataan ulang dokumen yang menyusun ulang teks berdasarkan kotak pembatas token yang terdeteksi pada langkah analisis tata letak. Gambar 4 (b) mengilustrasikan gambar teks padat yang dihasilkan, yang dikirim ke API OCR secara keseluruhan untuk mengurangi biaya transaksi. Sistem koordinat fleksibel di LayoutParser digunakan untuk mengubah hasil OCR relatif terhadap posisi aslinya di halaman.

Selain itu, dokumen sejarah biasanya menggunakan font unik dengan mesin terbang berbeda, yang secara signifikan menurunkan akurasi model OCR yang dilatih pada teks modern. Dalam dokumen ini, font datar khusus digunakan untuk mencetak angka dan tidak dapat dideteksi oleh mesin OCR siap pakai. Dengan menggunakan fungsionalitas yang sangat fleksibel dari LayoutParser, pendekatan jalur pipa dibangun yang mencapai akurasi pengenalan tinggi dengan sedikit usaha. Karena karakter memiliki struktur visual yang unik dan biasanya dikelompokkan bersama, kami melatih model tata letak untuk mengidentifikasi area angka dengan kategori khusus. Selanjutnya, LayoutParser memotong gambar dalam wilayah ini, dan mengidentifikasi karakter di dalamnya menggunakan model OCR yang dilatih sendiri berdasarkan CNN-RNN [6]. Model ini mendeteksi total 15 kemungkinan kategori, dan mencapai skor Jaccard 0,9816 dan jarak Levinstein rata-rata 0,1717 untuk prediksi token pada set pengujian.

Secara keseluruhan, dimungkinkan untuk membuat pipa digitalisasi yang rumit dan sangat akurat untuk digitalisasi skala besar menggunakan LayoutParser. Pipeline menghindari penetapan aturan rumit yang digunakan dalam metode tradisional, mudah dikembangkan, dan kuat terhadap outlier. Model DL juga menghasilkan hasil mendetail yang memungkinkan pendekatan kreatif seperti pengaturan ulang halaman untuk OCR.

[16] Ini mengukur tumpang tindih antara karakter yang terdeteksi dan kebenaran dasar, dan maksimumnya adalah 1.

[17] Ini mengukur jumlah suntingan dari teks kebenaran dasar ke teks yang diprediksi, dan lebih rendah lebih baik.



(a) Partial table at the bottom (b) Full page table (c) Partial table at the top (d) Mis-detected text line

Gbr. 6: Detektor tabel ringan ini dapat mengidentifikasi tabel (diuraikan dengan warna merah) dan sel (diarsir dengan warna biru) di lokasi berbeda pada halaman. Dalam beberapa kasus (d), ini mungkin menghasilkan prediksi kesalahan kecil, misalnya, gagal menangkap baris teks teratas dari tabel.

5.2 Ekstraktor Tabel Visual yang ringan

Mendeteksi tabel dan mem-parsing strukturnya (ekstraksi tabel) sangat penting untuk banyak tugas digitalisasi dokumen. Banyak karya sebelumnya [26, 30, 27] dan alat telah dikembangkan untuk mengidentifikasi dan mengurai struktur tabel. Namun mereka mungkin memerlukan pelatihan model rumit dari awal, atau hanya berlaku untuk dokumen PDF lahir digital. Di bagian ini, kami menunjukkan bagaimana LayoutParser dapat membantu membuat ekstraktor tabel visual yang ringan dan akurat untuk tabel map hukum menggunakan sumber daya yang ada dengan sedikit usaha.

Ekstraktor menggunakan model deteksi tata letak terlatih untuk mengidentifikasi area tabel dan beberapa aturan sederhana untuk memasangkan baris dan kolom dalam gambar PDF. Mask R-CNN [12] dilatih pada dataset PubLayNet [38] dari Zoo Model LayoutParser dapat digunakan untuk mendeteksi wilayah tabel. Dengan memfilter prediksi model dengan keyakinan rendah dan menghapus prediksi yang tumpang tindih, LayoutParser dapat mengidentifikasi area tabular di setiap halaman, yang secara signifikan menyederhanakan langkah selanjutnya. Dengan menerapkan fungsi deteksi garis dalam segmen tabular, yang disediakan dalam modul utilitas dari LayoutParser, pipeline dapat mengidentifikasi tiga kolom berbeda dalam tabel. Metode pengelompokan baris kemudian diterapkan melalui analisis koordinat y dari kotak pembatas token di kolom paling kiri, yang diperoleh dari mesin OCR. Algoritma non-maximal suppression digunakan untuk menghapus baris duplikat dengan celah yang sangat kecil. Ditunjukkan pada Gambar 6, pipeline yang dibangun dapat mendeteksi tabel pada posisi yang berbeda pada halaman secara akurat. Tabel lanjutan dari halaman berbeda digabungkan, dan representasi tabel terstruktur dengan mudah dibuat.

[18] https://github.com/atlanhq/camelot, https://github.com/tabulapdf/tabula

6. Kesimpulan

LayoutParser menyediakan perangkat komprehensif untuk analisis gambar dokumen berbasis pembelajaran mendalam. Pustaka off-the-shelf mudah dipasang, dan dapat digunakan untuk membangun jaringan pipa yang fleksibel dan akurat untuk memproses dokumen dengan struktur yang rumit. Ini juga mendukung penyesuaian tingkat tinggi dan memungkinkan pelabelan dan pelatihan model DL yang mudah pada kumpulan data gambar dokumen yang unik. Platform komunitas LayoutParser memfasilitasi berbagi model DL dan pipeline DIA, mengundang diskusi, dan mempromosikan reproduktifitas dan penggunaan ulang kode. Tim LayoutParser berkomitmen untuk terus memperbarui perpustakaan dan menghadirkan kemajuan terbaru dalam DIA berbasis DL, seperti pemodelan dokumen multi-modal [37, 36, 9] (prioritas mendatang), ke beragam pengguna akhir pengguna.

Ucapan Terima Kasih Kami berterima kasih kepada pengulas anonim atas komentar dan saran mereka. Proyek ini didukung sebagian oleh NSF Grant OIA-2033558 dan pendanaan dari Harvard Data Science Initiative dan Harvard Catalyst. Zejiang Shen berterima kasih kepada Doug Downey atas sarannya.

Referensi

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, GS, Davis, A., Dean, J., Devin, M ., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J ., Man'e, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar , K., Tucker, P., Vanhoucke, V., Vasudevan, V., Vi'egas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. , Zheng, X.: TensorFlow: Pembelajaran mesin skala besar pada sistem heterogen (2015), https://www.tensorflow.org/, perangkat lunak tersedia dari tensorflow.org [2] Alberti, M., Pondenkandath, V., W¨ursch, M., Ingold, R., Liwicki, M.: Deepdiva: kerangka kerja python yang sangat fungsional untuk eksperimen yang dapat direproduksi. Di: 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR). hal. 423–428. IEEE (2018)

[3] Antonacopoulos, A., Bridson, D., Papadopoulos, C., Pletschacher, S.: Kumpulan data realistis untuk evaluasi kinerja analisis tata letak dokumen. Di: 2009 Konferensi Internasional ke-10 tentang Analisis dan Pengakuan Dokumen. hlm. 296–300. IEEE (2009)

[4] Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Kesadaran wilayah karakter untuk deteksi teks. Dalam: Prosiding Konferensi IEEE/CVF tentang Visi Komputer dan Pengenalan Pola. hlm. 9365–9374 (2019)

[5] Deng, J., Dong, W., Socher, R., Li, LJ, Li, K., Fei-Fei, L.: ImageNet: Database Gambar Hierarkis Berskala Besar . Di dalam: CVPR09 (2009)

[6] Deng, Y., Kanervisto, A., Ling, J., Rush, AM: Generasi gambar-ke-markup dengan perhatian kasar ke halus. Dalam: Konferensi Internasional tentang Pembelajaran Mesin. hal. 980–989. PMLR (2017)

[7] Ganin, Y., Lempitsky, V.: Adaptasi domain tanpa pengawasan oleh backpropagation. Dalam: Konferensi internasional tentang pembelajaran mesin. hlm. 1180–1189. PMLR (2015)

[8] Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N., Peters, M., Schmitz, M., Zettlemoyer, L.: Allennlp: Platform pemrosesan bahasa alami semantik yang dalam . pracetak arXiv arXiv:1803.07640 (2018)

[9]Lukasz Garncarek, Powalski, R., Stanislawek, T., Topolski, B., Halama, P., Grali´nski, F.: Lambert: Pemodelan tata letak (bahasa) menggunakan bert untuk ekstraksi formasi (2020)

[10] Graves, A., Fern'andez, S., Gomez, F., Schmidhuber, J.: Klasifikasi temporal koneksionis : pelabelan data urutan tidak tersegmentasi dengan jaringan saraf berulang. Dalam: Prosiding konferensi internasional ke-23 tentang Pembelajaran Mesin. hlm. 369–376 (2006)

[11] Harley, AW, Ufkes, A., Derpanis, KG: Evaluasi jaring konvolusional dalam untuk klasifikasi dan pengambilan gambar dokumen. Di: 2015 Konferensi Internasional ke-13 tentang Analisis dan Pengakuan Dokumen (ICDAR). hlm. 991–995. IEEE (2015)

[12] He, K., Gkioxari, G., Doll'ar, P., Girshick, R.: Mask r-cnn. Dalam: Prosiding konferensi internasional IEEE tentang visi komputer. hlm. 2961–2969 (2017)

[13] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning untuk pengenalan gambar. Dalam: Prosiding konferensi IEEE tentang visi komputer dan pengenalan pola. hlm. 770–778 (2016)

[14] Kay, A.: Tesseract: Mesin pengenalan karakter optik sumber terbuka. Linux J. 2007(159), 2 (Jul 2007)

[15] Lamiroy, B., Lopresti, D.: Sebuah arsitektur terbuka untuk pembandingan analisis dokumen end-to-end. Dalam: Konferensi Internasional 2011 tentang Analisis dan Pengakuan Dokumen. hlm. 42–47. IEEE (2011)

[16] Lee, BC, Weld, DS: Navigator surat kabar: Pencarian segi terbuka untuk 1,5 juta gambar. Dalam: Publikasi Tambahan dari Posium ACM Tahunan ke-33 tentang Perangkat Lunak dan Teknologi Antarmuka Pengguna. P. 120–122. UIST '20 Adjunct, Asosiasi Mesin Komputasi, New York, NY, USA (2020). https://doi.org/10.1145/3379350.3416143, https://doi-org.offcampus. lib.washington.edu/10.1145/3379350.3416143 [17] Lee, BCG, Mears, J., Jakeway, E., Ferriter, M., Adams, C., Yarasavage, N., Thomas, D., Zwaard, K. , Weld, DS: The Newspaper Navigator Dataset: Extracting Headlines and Visual Content from 16 Million Historic Newspaper Pages in Chronicling America, hal. 3055–3062. Association for Computing Machinery, New York, NY, USA (2020), https://doi.org/10.1145/3340531.3412767 [18] Li, M., Cui, L., Huang, S., Wei, F., Zhou , M., Li, Z.: Tablebank: Tolok ukur tabel untuk deteksi dan pengenalan tabel berbasis gambar. pracetak arXiv arXiv:1903.01949 (2019)

[19] Lin, TY, Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Doll'ar, P., Zitnick, CL: Microsoft coco: Objek umum dalam konteks . Dalam: Konferensi Eropa tentang visi komputer. hlm. 740–755. Peloncat (2014)

[20] Long, J., Shelhamer, E., Darrell, T.: Jaringan konvolusional penuh untuk segmentasi semantik. Dalam: Prosiding konferensi IEEE tentang visi komputer dan pengenalan pola. hlm. 3431–3440 (2015)

[21] Neudecker, C., Schlarb, S., Dogan, ZM, Missier, P., Sufi, S., Williams, A., Wolsten croft, K.: Platform pengembangan alur kerja eksperimental untuk digitalisasi dan analisis dokumen historis . Dalam: Prosiding lokakarya tahun 2011 tentang pencitraan dan pemrosesan dokumen sejarah. hlm. 161–168 (2011)

[22] Oliveira, SA, Seguin, B., Kaplan, F.: dhsegment: Pendekatan pembelajaran mendalam umum untuk segmentasi dokumen. Di: 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR). hlm. 7–12. IEEE (2018)

[23] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer , A.: Diferensiasi otomatis di python (2017)
[24] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: Pustaka deep learning dengan gaya imperatif dan berperforma tinggi. pracetak arXiv arXiv:1912.01703 (2019)
[25] Pletchacher, S., Antonacopoulos, A.: Kerangka format halaman (analisis halaman dan elemen kebenaran dasar). Di: 2010 Konferensi Internasional ke-20 tentang Pengenalan Pola. hlm. 257–260. IEEE (2010)
[26] Prasad, D., Gadpal, A., Kapadni, K., Visave, M., Sultanpure, K.: Cascadetabnet: Pendekatan untuk deteksi tabel ujung ke ujung dan pengenalan struktur dari dokumen berbasis gambar. Dalam: Prosiding Konferensi IEEE/CVF tentang Visi Komputer dan Lokakarya Pengenalan Pola. hlm. 572–573 (2020)
[27] Qasim, SR, Mahmood, H., Shafait, F.: Memikirkan kembali pengenalan tabel menggunakan jaringan saraf graf. Dalam: Konferensi Internasional 2019 tentang Analisis dan Pengakuan Dokumen (ICDAR). hlm. 142–147. IEEE (2019)
[28] Ren, S., He, K., Girshick, R., Sun, J.: Lebih cepat r-cnn: Menuju deteksi objek real-time dengan jaringan proposal wilayah. Dalam: Kemajuan dalam sistem pemrosesan informasi saraf. hlm. 91–99 (2015)
[29] Scarselli, F., Gori, M., Tsoi, AC, Hagenbuchner, M., Monfardini, G.: Model jaringan saraf grafik. Transaksi IEEE pada jaringan saraf 20(1), 61–80 (2008)
[30] Schreiber, S., Agne, S., Wolf, I., Dengel, A., Ahmed, S.: Deepdesrt: Pembelajaran mendalam untuk deteksi dan pengenalan struktur tabel dalam gambar dokumen. Di: 2017 14th IAPR international conference on document analysis and recognition (ICDAR). vol. 1, hlm. 1162–1167. IEEE (2017)
[31] Shen, Z., Zhang, K., Dell, M.: Kumpulan data besar dokumen Jepang bersejarah dengan tata letak yang rumit. Dalam: Prosiding Konferensi IEEE/CVF tentang Visi Komputer dan Lokakarya Pengenalan Pola. hlm. 548–549 (2020)
[32] Shen, Z., Zhao, J., Dell, M., Yu, Y., Li, W.: Olala: Anotasi tata letak berbasis pembelajaran aktif tingkat objek . pracetak arXiv arXiv:2010.01762 (2020)
[33] Studer, L., Alberti, M., Pondenkandath, V., Goktepe, P., Kolonko, T., Fischer, A., Liwicki, M., Ingold, R.: Sebuah studi komprehensif pra-imagenet pelatihan analisis citra dokumen sejarah. Dalam: Konferensi Internasional 2019 tentang Analisis dan Pengakuan Dokumen (ICDAR). hlm. 720–725. IEEE (2019)
[34] Serigala, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz , M., et al.: Transformator Huggingface: Pemrosesan bahasa alami yang canggih. pracetak arXiv arXiv:1910.03771 (2019)
[35] Wu, Y., Kirillov, A., Massa, F., Lo, WY, Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019)
[36] Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che , W., et al.: Layoutlmv2: Prapelatihan multimodal untuk pemahaman dokumen yang kaya secara visual. pracetak arXiv arXiv:2012.14740 (2020)
[37] Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: Layoutlm: Prapelatihan teks dan layout untuk pemahaman gambar dokumen (2019)
[38] Zhong, X., Tang, J., Yepes, AJ: Publaynet: kumpulan data terbesar yang pernah ada untuk analisis tata letak dokumen. Dalam: Konferensi Internasional 2019 tentang Analisis dan Pengakuan Dokumen (ICDAR). hlm. 1015–1022. IEEE (Sep 2019). https://doi.org/10.1109/ICDAR.2019.00166