Indonesian Journal of Electronics and Instrumentation Systems (IJEIS)
Vol.x, No.x, April xxxx, pp. 1~5
ISSN (print): 2088-3714, ISSN (online): 2460-7681
DOI: 10.22146/ijeis.xxxx

PENGENALAN TEKS BAHASA INDONESIA PADA CITRA TULISAN TANGAN BERBASIS TRANSFORMER

Abstrak

Digitalisasi dokumen dapat dipercepat berkat kemajuan teknologi. Banyak upaya telah dilakukan untuk mengenali teks dari foto. Banyak arsitektur mampu mengenali teks, khususnya citra tulisan tangan salah satunya adalah transformer. Pada penelitian sebelumnya masih banyak yang menggunakan dataset citra dengan aksara tegak sehingga kurang variatif. Untuk meningkatkan keahlian pemodelan pembelajaran, proyek ini berfokus pada pengimplementasian dan pengembangan sistem pada Transformers dengan pengujian dataset yang lebih bervariasi.

Dataset yang digunakan terdiri dari foto dengan tulisan Indonesia. setelah langkah pra-pemrosesan kemudian akan diubah menjadi token dengan label kelas dan koordinat kotak pembatas untuk anotasi gambar. Dataset akan dilatih menggunakan arsitektur transformer. Encoder-decoder merupakan dasar dari arsitektur Transformer ini. Pengujian data dilakukan setelah model dilatih menggunakan mean Average Precision (mAP).

Sistem yang dibuat mampu mengenali dan mengklasifikasikan objek secara akurat dari data gambar tulisan tangan, termasuk objek yang mewakili kata-kata bahasa Indonesia. Hyperparameter yang paling optimal didapatkan batch dan jumlah epoch masing-masing 32 dan 40. Dengan menggunakan parameter terbaik, evaluasi model menghasilkan data dari sampel latih dan uji dengan masing-masing nilai mAP 0,97 dan 0,95.
Kata kunci*—Transformer, Image Recognition, Tulisan Tangan*

Abstract

Document digitization can be accelerated thanks to advances in technology. Many attempts have been made to recognize text from photos. Many architectures are able to recognize text, especially handwritten images, one of which is a transformer. In previous studies, there were still many who used image datasets with upright characters so that they were less varied. To enhance learning modeling skills, this project focuses on implementing and developing systems on Transformers with a more varied testing dataset.

The dataset used consists of photos with Indonesian writing. after the pre-processing step it will then be converted into a token with a class label and bounding box coordinates for the image annotation. The dataset will be trained using the transformer architecture. The encoder-decoder is the basis of this Transformer architecture. Data testing was carried out after the model was trained using the mean Average Precision (mAP).

The system created is able to accurately recognize and classify objects from handwritten image data, including objects that represent Indonesian words. The most optimal hyperparameters were obtained from batches and the number of epochs, respectively 32 and 40. By using the best parameters, the model evaluation produced data from training and test samples with mAP values of 0.97 and 0.95, respectively.
Keywords—*Transformer, Image Recognition, Handwriting*

Received June 1st,2012; Revised June 25th, 2012; Accepted July 10th, 2012

1. PENDAHULUAN

Deteksi objek semakin banyak diterapkan dalam berbagai industri di Indonesia. Misalnya, deteksi objek digunakan dalam bidang keamanan dan pengawasan, transportasi, manufaktur, pertanian, kesehatan, dan sektor lainnya. Dalam sektor pendidikan, deteksi objek juga dapat diterapkan sebagai proses pembelajaran teknologi yang dapat dikembangkan secara luas serta dapat membuat para pelajar menjadi lebih inovatif dalam mengembangkan sistem deteksi objek. Salah satu pengembangan deteksi objek ini yaitu dengan mengubah input data berupa tulisan dengan berbagai variasi. Objek tulisan ini dapat ditemukan di banyak tempat terutama di sekolah-sekolah maupun universitas. Deteksi objek dengan input tulisan ini dapat bermanfaat untuk mengecek keaslian tulisan dari seseorang maupun membaca tulisan yang tidak mampu terbaca oleh mata.

Perkembangan teknologi saat ini tentang pengklasifikasian objek benda sudah sangat banyak. Pengembangan arsitektur baru sudah mulai bermunculan dengan berbagai kelebihan dan kekurangan. Pada saat ini, konsep mengenai pengklasifikasian objek mulai merambah ke arsitektur transformer. Sistem yang dikembangkan [6] merupakan permulaan awal dari sebuah pengembangan teknologi baru saat ini.

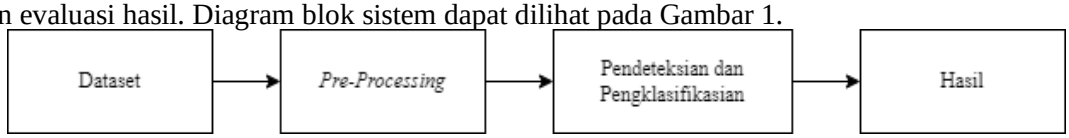
Sejak [6] berhasil menerapkan Transformer pada berbagai tolok ukur pengenalan citra, ada banyak sekali karya tindak lanjut yang menunjukkan bahwa CNN mungkin bukan arsitektur yang optimal untuk Computer Vision lagi.

Dalam arsitektur Transformer, *self-attention* memungkinkan model untuk memperhatikan interaksi global antara piksel dalam gambar, yang dapat menghasilkan pemahaman yang lebih baik tentang konteks spasial dalam citra. Ini membantu model untuk menangkap fitur-fitur yang relevan dan memperoleh pemahaman yang lebih holistik tentang citra secara keseluruhan, sehingga meningkatkan kualitas representasi yang dihasilkan dan memperbaiki performa klasifikasi [1].

Penelitian ini akan fokus merancang dan mengimplementasikan arsitektur Transformer yang dilakukan oleh [1] sebagai basis model image text recognition untuk mendeteksi tulisan bahasa Indonesia pada citra. Pada penelitian ini, penulis mengusulkan arsitektur yang berbeda yaitu DeiT sebagai proses *embedding* untuk memproses input citra kemudian dilakukan pembelajaran menggunakan Transformer untuk mendapatkan hasil deteksi dan klasifikasi objek.

2. METODE PENELITIAN

Sistem dibagi menjadi beberapa bagian proses, yaitu anotasi dataset, pre-processing (augmentasi citra), Membuat konstruksi urutan token, Membangun arsitektur, proses pelatihan, dan evaluasi hasil. Diagram blok sistem dapat dilihat pada Gambar 1.



Gambar 1 Diagram blok Rancangan sistem

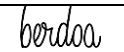
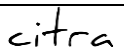
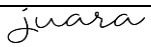
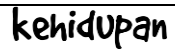
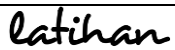
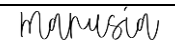
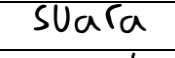
Dataset yang digunakan berupa citra dengan objek tulisan kata-kata dalam bahasa Indonesia. Citra yang sudah dianotasikan sebelumnya akan dibagi menjadi dua data yaitu data train dan data validasi. Kedua data ini dimasukkan ke tahapan *pre-processing* di mana citra akan di-*resize*. Untuk anotasi citra, akan diubah menjadi token yang berisi label kelas dan titik koordinat *bounding box*. Proses ini disebut dengan tokenizer. Setelah tahap ini, dataset akan dilatih dengan menggunakan arsitektur transformer. Komponen utama dari arsitektur Transfomer adalah *encoder-decoder*. Encoder bekerja untuk menangkap hubungan antara piksel pada peta fitur. *Encoder* ini menggunakan model DeiT sebagai kerangka kerja. *Decoder* bekerja untuk menangkap *patch embedding* yang dihasilkan oleh *encoder* dan belajar memprediksi

urutan dari token [1]. Setelah model dilatih, tahapan selanjutnya yaitu pengujian. Pengujian ini dilakukan dengan mencari nilai *mean Average Precision* (mAP).

2.1 Anotasi Dataset

Ada 500 dataset citra tulisan tangan dengan 50 macam bentuk gaya tulisan dari sepuluh kata. Beberapa kata yang dipakai terlihat pada Tabel 1. Proses anotasi pada citra dataset diperlukan untuk memberikan informasi berupa titik koordinat *bounding box* dan memberi nama kelas untuk proses pendeteksian dan pengklasifikasian. Anotasi dilakukan menggunakan aplikasi labelImg dengan format PascalVOC yang menghasilkan file berekstensi .xml.

Tabel 1 Kelas Kata

No	Kelas	Gambar
1	absolut	
2	berdoa	
3	citra	
4	gerhana	
5	juara	
6	kehidupan	
7	latihan	
8	manusia	
9	suara	
10	wanita	

2.2 Pre-Processing

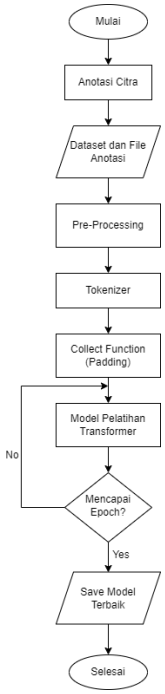
Proses *pre-processing* yang dilakukan ini dibagi antara citra dengan anotasi. Proses *pre-processing* pada citra dilakukan dengan me-*resize* bentuk citra. Pada proses *pre-processing* anotasi dilakukan dengan membuat sebuah dataframe dari hasil anotasi yang berisi token dari kelas dan *bounding box*. Setiap token diawali dengan token BOS dan diakhiri dengan EOS [1].

2.3 Tokenizer

Pembuat token disebut tokenizer. Dalam membuat sebuah token perlu dilakukan beberapa langkah yaitu memberi tanda awal dan akhir dengan menggunakan token BOS dan EOS, mengkuantisasi nilai-nilai titik koordinat, menyandikan kelas objek tulisan menjadi token yang sesuai, dan mengacak urutan objek tulisan apabila terdapat tulisan lebih dari satu [1].

2.4 Pelatihan Data

Pada tahap ini, pelatihan data dilakukan agar dapat mengklasifikasikan kelas-kelas dari setiap citra tulisan yang dilatih menggunakan arsitektur transformer dengan beberapa layer *encoder-decoder*.

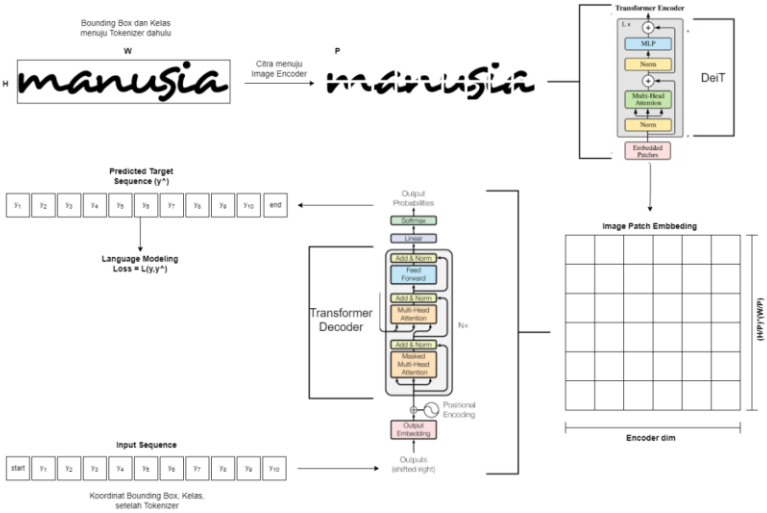


Gambar 2 Diagram Blok Pelatihan Transformer

Citra yang sudah dianotasikan kelasnya akan dipisah dalam proses pre-processing. Citra akan dilakukan resize menjadi ukuran 384 × 384. Sedangkan hasil anotasi dan kelasnya akan melewati proses tokenizer. Pada proses ini, titik-titik *bounding box* akan dikuantisasi menjadi nilai diskrit dan kelas akan disandikan dengan nilai yang disesuaikan. Tokenizer juga akan membuat sequence yang terdiri dari urutan nilai titik-titik koordinat yang sudah dikuantisasi dan kelas yang sudah disandikan. Setelah melalui proses tokenizer, dilakukan padding agar setiap sequence memiliki panjang yang sama. Setelah itu, dibangun sebuah model transformer untuk proses pelatihan. Selama proses ini, model perlu memprediksi token berikutnya hanya dengan melihat token sebelumnya.

2.5 Arsitektur Transformer

Encoder dan *Decoder* adalah dua bagian utama dari arsitektur transformer. *Encoder* untuk menangkap hubungan antara piksel pada peta fitur. *Decoder* untuk menghasilkan urutan target satu token pada satu waktu.



Gambar 3 Arsitektur Transformer

Piksel dari gambar input ditangkap oleh transformator *encoder*. *Encoder* bertugas mengumpulkan piksel dan mengkodekannya ke dalam bentuk tersembunyi. Struktur Transformer awalnya dibuat untuk aplikasi pemrosesan bahasa alami (NLP), tetapi DeiT mengadaptasinya untuk penggunaan dalam konteks pengenalan gambar.

Patch embeddings dari gambar input digunakan oleh Transformer Decoder untuk belajar memprediksi urutan *bounding box*. Berdasarkan token sebelumnya dan representasi gambar yang disandikan, dekoder membuat satu token dalam satu waktu.

2.6 Rancangan Pengujian

Pengujian sistem dilakukan untuk mengetahui seberapa akurasi dalam tingkat keakuratan setiap pendeteksian dan pengklasifikasian tulisan pada citra. Proses pengujian menggunakan dataset baru yang belum pernah dilatih sebelumnya. Sistem yang dirancang dikatakan berhasil ketika model yang telah dibuat mampu untuk mendeteksi dan mengklasifikasi tulisan pada citra dengan baik.

Ada beberapa parameter yang dibutuhkan untuk melihat seberapa baik peforma sistem. Parameter yang digunakan untuk perbandingan akan divariasikan sedangkan parameter yang lain akan dibuat konstan. Parameter yang divariasikan antara lain adalah nilai *batch* dan jumlah *epoch*. Terakhir, pengujian sistem menggunakan data test yang belum pernah dilatih sebelumnya juga akan menjadi parameter pengujian. Pengujian data test dilakukan untuk mengetahui seberapa baik prediksi yang dihasilkan dari model.

3. HASIL DAN PEMBAHASAN

3.1 Dataset

500 gambar dengan 10 kelas membentuk dataset. Dataset ini dibagi menjadi dua bagian: data untuk pelatihan dan data untuk validasi. Fungsi StratifiedGroupKFold() digunakan untuk membagi dataset. Menggunakan teknik K-Fold, fungsi ini dapat membagi kerangka data menjadi set pelatihan dan set validasi. Tabel 2 menampilkan jumlah partisi dataset.

Tabel 2 Dataset yang digunakan

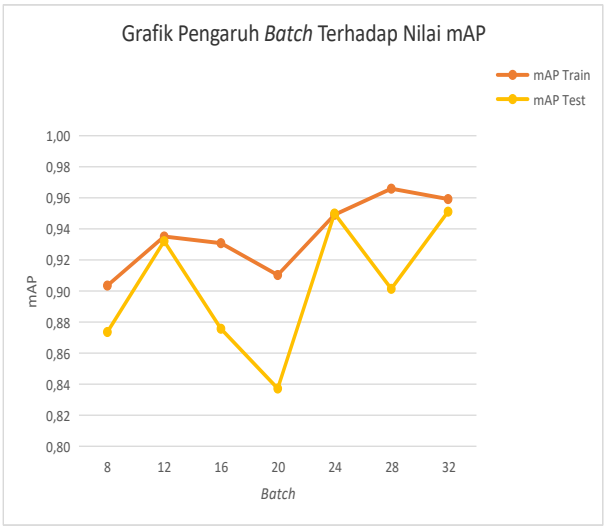
Data *Training*	Data *Validation*	Total
400	100	500

3.2 Hasil Pengujian Model

Menemukan hyperparameter terbaik untuk membangun model Transformer adalah bagaimana pengujian model dilakukan. Nilai *batch*, jumlah total *epoch*, dan *learning rate* adalah variabel yang patut dipertimbangkan. Model selanjutnya akan dilatih dan dievaluasi menggunakan parameter terbaik yang telah ditemukan. Nilai mAP menentukan cara menemukan parameter dengan performa terbaik.

3.2.1 Pengaruh Batch

Tujuan dari pengujian ini adalah untuk menentukan nilai *batch* yang ideal. Hasil evaluasi seringkali akan semakin ideal semakin besar nilai *batch*-nya. Dalam pengujian ini, tujuh nilai *batch* yang berbeda—8, 12, 16, 20, 24, dan 32—disediakan dari rentang 8 hingga 32. Dalam pengujian ini, nilai *epoch* dan *learning rate* akan ditetapkan pada 50 dan 0,0001. Gambar 4 menampilkan temuan pengujian sebagai grafik variasi *batch* pada mAP.

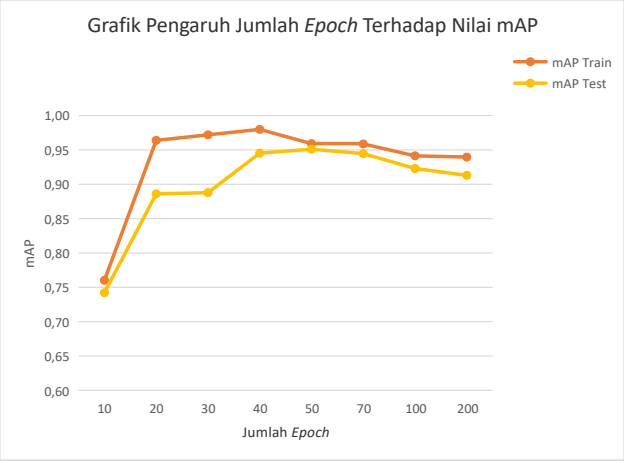


Gambar 4 Grafik Pengaruh Variasi *Batch* Terhadap Nilai mAP

Menurut hasil pengujian variasi *batch*, telah terjadi peningkatan dan penurunan. Nilai *batch* 16 dan 20 mengalami penurunan. Hal ini dapat terjadi karena memproses terlalu banyak data sekaligus, yang menurunkan performa hasil deteksi objek dari data uji. Sementara nilai pada *batch* 28 menurun, data uji terus meningkat. Ini bisa berarti bahwa tidak cukup banyak contoh-contoh individual data pelatihan yang dipelajari oleh model. Pengujian ini telah menunjukkan bahwa nilai 32 adalah nilai *batch* terbaik untuk penelitian ini.

3.2.2 Pengaruh Jumlah Epoch

Untuk mendapatkan hasil terbaik dalam pengujian ini, jumlah *epoch* akan disesuaikan. Dalam pengujian ini, nilai kumpulan dari pengujian sebelumnya akan diterapkan, dan nilai kecepatan pembelajaran ditetapkan. Ada 10, 20, 30, 40, 50, 70, 100, dan 200 nilai *epoch* yang berbeda. Performa terbaik untuk setiap nilai *epoch* yang digunakan ditentukan dengan menggunakan variasi nilai ini.



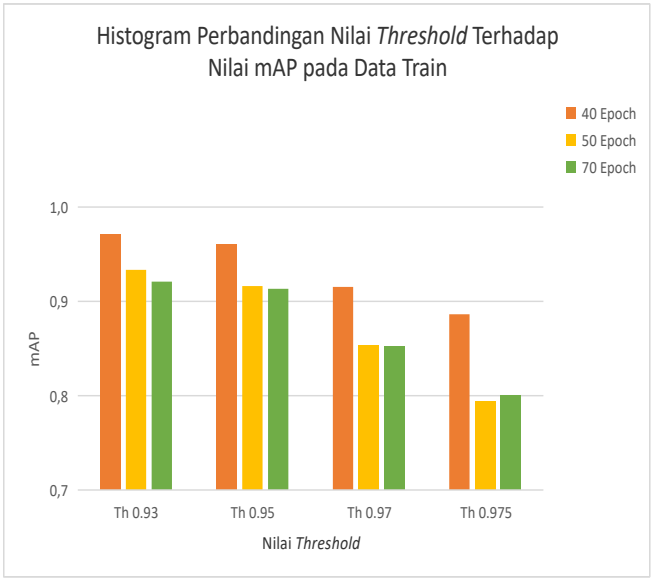
Gambar 4 Grafik Pengaruh Variasi Jumlah *Epoch* Terhadap Nilai mAP

Semua nilai variasi *epoch* bekerja dengan baik dalam pengujian ini. Namun, performa menurun saat jumlah *epoch* naik di atas 70. Karena kemampuan model yang terbatas untuk menganalisis pola rumit dalam data, *overfitting* adalah penyebabnya. Model memiliki lebih banyak peluang untuk menemukan pola dalam data seiring bertambahnya jumlah *epoch*. Bahkan jika lebih banyak waktu dihabiskan untuk melatih model yang terlalu sederhana atau memiliki kapasitas terbatas, model tersebut tidak akan dapat mengenali pola rumit dalam data. Tiga nilai *epoch*—40, 50, dan 70—menunjukkan nilai persentase mAP tertinggi dalam pengujian yang disebutkan di atas. Kami akan menguji ketiga nilai *epoch* ini sekali lagi.

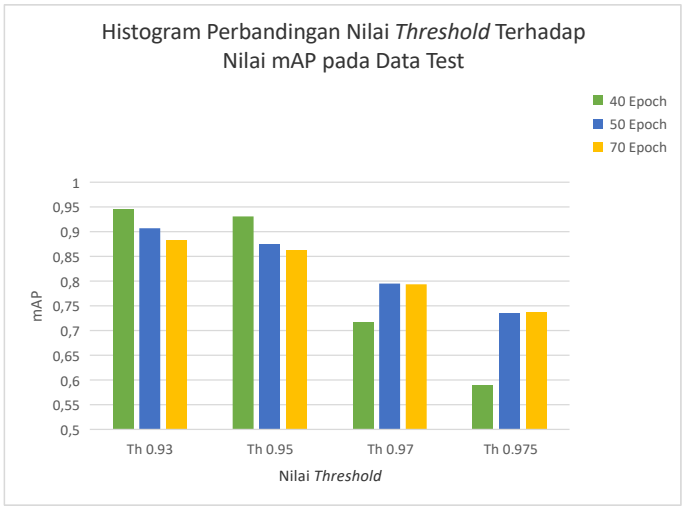
3.2.2.1 Pengujian Model dengan Threshold Tinggi

Tujuan dari pengujian ini adalah untuk menentukan mana dari ketiga model tersebut di atas yang memiliki kinerja terbaik. Nilai *threshold* dalam pengujian sebelumnya ditetapkan sebesar 0,5 sesuai dengan norma industri. Oleh karena itu, model dengan performa yang hampir sama dapat diuji sekali lagi menggunakan *threshold* yang lebih tinggi.

Empat nilai *threshold*—0,93, 0,95, 0,97, dan 0,975—digunakan dalam pengujian ini. Metode penilaian deteksi objek dianggap lebih akurat semakin tinggi nilai ambang batas yang digunakan. Skor kepercayaan yang diperoleh dari deteksi objek menentukan hal ini. Hanya prediksi objek dengan peringkat *confidents* yang dianggap benar ketika ambang batas yang lebih tinggi diterapkan. Ini dapat mengurangi jumlah *false positive* dan jumlah deteksi yang dihasilkan model.



Gambar 5 Histogram Perbandingan Nilai *Threshold* Terhadap Nilai mAP pada Data Train



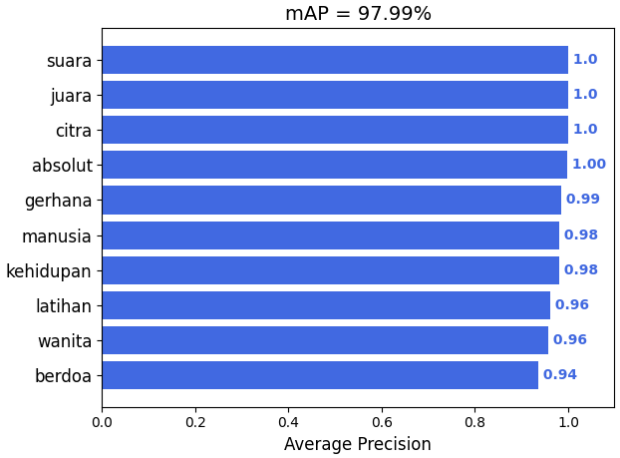
Gambar 6 Histogram Perbandingan Nilai *Threshold* Terhadap Nilai mAP pada Data Test

Tujuan dari pengujian ini adalah untuk menentukan jumlah *epoch* data latih dan data uji yang paling optimal. Gambar 5 menunjukkan bahwa pada setiap pengaturan threshold, model dengan 40 *epoch* memiliki nilai mAP tertinggi. Sedangkan nilai mAP untuk model dengan *epoch* 50 dan 70 hampir sama. Model dengan nilai mAP terbaik dari Gambar 6 adalah model dengan 40 *epoch* dengan threshold di bawah 0,95. Nilai mAP terbaik dihasilkan oleh model dengan *epoch* 50 dan 70 dengan nilai threshold lebih dari 0,95. Nilai *threshold* yang lebih rendah dapat dipertimbangkan untuk memperluas cakupan prediksi positif, menghasilkan penarikan kembali yang lebih tinggi dengan mengorbankan presisi. Karena model dengan 40 *epoch* memiliki nilai mAP terbaik pada *threshold* tinggi, pengujian ini hanya akan berkonsentrasi pada peningkatan cakupan prediksi positif.

3.2.3 Pegujian dengan Parameter Optimal

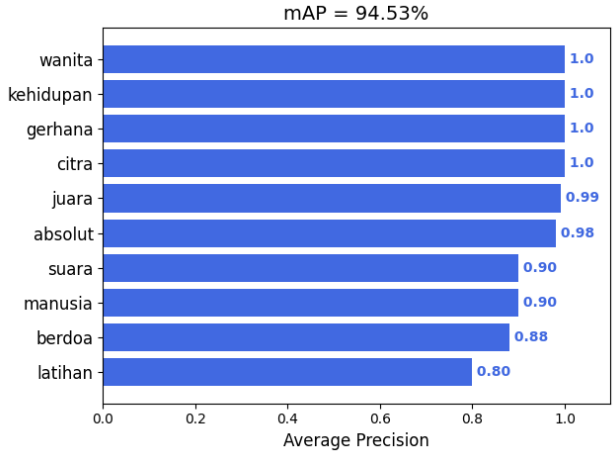
Nilai *batch* dan jumlah *epoch* telah diperoleh pada pengujian sebelumnya. Berdasarkan nilai-nilai optimal ini, model akan diuji untuk melihat bagaimana ia berfungsi pada setiap kelas yang diprediksi. Pada pengujian ini, model digunakan dengan nilai *batch* 32, jumlah *epoch* 40,

dan tingkat pembelajaran 0,0001. Untuk pengujian ini, nilai ambang mAP akan dikembalikan ke nilai standar sebelumnya, yaitu 0,5.



Gambar 7 Hasil Nilai AP Setiap Kelas dengan Data Train

Gambar 7 di atas menunjukkan bahwa model yang digunakan sangat baik untuk pelatihan data Train karena nilai AP untuk semua kelas lebih dari 0,9. Empat kelas, "suara", "citra", "juara", dan "absolut", memiliki nilai AP sempurna. Kelas "berdoa" memiliki nilai AP terendah, yaitu 0,94.



Gambar 8 Hasil Nilai AP Setiap Kelas dengan Data Test

Nilai AP masing-masing kelas pada pengujian ini dinilai sangat baik karena banyak yang mendapat skor 0,9. Gambar 8 menampilkan hasil pengujian nilai AP pada data Test. Empat kelas "wanita", "kehidupan", "gerhana", dan "berdoa" adalah yang terbaik untuk hasil AP. Kelas "berdoa" dan "latihan" adalah satu-satunya dua kelas dengan nilai AP di bawah 0,9.

Pada tahapan evaluasi ini, model sudah dapat memprediksi setiap kelas dengan cukup baik saat diuji dengan parameter ideal. Gambar 9 menggambarkan hasil deteksi objek dan klasifikasi.

Gambar 9 Contoh Hasil Deteksi Objek

4. KESIMPULAN

Sistem yang dikembangkan dapat secara efektif mengenali dan mengklasifikasikan benda-benda dari data gambar tulisan tangan yang termasuk objek kata bahasa Indonesia. Menggunakan *learning rate* 0,0001, *hyperparameter* terbaik yang ditemukan selama pengujian menghasilkan nilai *batch* 32 dan total 40 *epoch*. Hasil evaluasi model menggunakan data dari sampel Train dan Test dengan nilai mAP masing-masing 0,98 dan 0,95.

5. SARAN

Berdasarkan penelitian yang telah dilakukan, terdapat saran yang dapat dilakukan untuk penelitian selanjutnya:

1. Perluas jumlah dan jangkauan dataset, misalnya dengan menyertakan banyak objek tulisan dalam satu gambar dan meningkatkan kelas penulisan.
2. Mencoba membandingkan arsitektur ini dengan yang lain, padahal sudah sangat bagus, agar bisa diperbaiki.
3. Untuk mempelajari lebih lanjut tentang hasil deteksi dan klasifikasi terlatih, tambahkan data penilaian termasuk akurasi, presisi, *recall*, dan *f1-score*.
4. Tingkatkan kinerja sistem dengan memanfaatkan alat berkualitas lebih tinggi untuk pelatihan dan pemodelan sistem.

DAFTAR PUSTAKA

[1] Chen T, Saxena S, Li L et al., "Pix2seq: A Language Modeling Framework for Object Detection", ArXiv, ID: 2109.10852, 2021. [Online]. Available: http://arxiv.org/abs/2109.10852. [Accessed: Feb-2023]

[2] Ly V, Doan T, Ly N, "Transformer-based model for Vietnamese Handwritten Word Image Recognition", Proceedings - 2020 7th NAFOSTED Conference on Information and Computer Science, NICS 2020 page 163-168, 2020 doi: 10.1109/NICS51282.2020.9335877 Scopus: 2-s2.0-85101122800.

[3] Vaswani A, Shazeer N, Parmar N et al., "Attention Is All You Need", ArXiv, ID: 1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762. [Accessed: Feb-2023].

[4] Touvron H, Cord M, Jégou H, "DeiT III: Revenge of the ViT", ArXiv, ID: 2204.07118 2022. [Accessed: Jun 2023] Available: https://arxiv.org/pdf/2204.07118

[5] Touvron H, Cord M, Douze M et al., "Training Data-Efficient Image Transformers & Distillation Through Attention", ArXiv, ID: 2012.12877, 2020. [Accessed: Jun 2023] Available: http://arxiv.org/abs/2012.12877

[6] Dosovitskiy A, Beyer L, Kolesnikov A et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ArXiv, ID: 2010.11929, 2020. [Accessed: Nov 2022] Available: http://arxiv.org/abs/2010.11929

[7] Bi J, Zhu Z, Meng Q, "Transformer in Computer Vision", 2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology, CEI 2021 page 178-188, 2021. doi: 10.1109/CEI52496.2021.9574462 Scopus: 2-s2.0-85118925905

[8] Bhunia A, Khan S, Cholakkal H et al., "Handwriting Transformers", ArXiv, ID: 2104.03964, 2021. [Accessed: Nov 2022] Available: http://arxiv.org/abs/2104.03964.

[9] Pu Y, Apel D, Szmigiel A et al. "Image Recognition of Coal and Coal Gangue Using a Convolutional Neural Network and Transfer Learning", Energies, Vol. 12(9), 2019. doi: 10.3390/en12091735 Scopus: 2-s2.0-85065980182

[10] Shaikh M, Duan T, Chauhan M et al., "Attention based Writer Independent Verification", Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR, page 373-379, 2020. doi: 10.1109/ICFHR2020.2020.00074 Scopus: 2-s2.0-85097792375

[11] Bayat O, Aljawarneh S, Carlak H et al. "Understanding of a Convolutional Neural Network", Book: Proceedings of 2017 International Conference on Engineering & Technology (ICET'2017) : Akdeniz University, Antalya, Turkey, 21-23 August, 2017. ISBN: 9781538619490