

## Identifikasi Hukum Tajwid pada Citra Teks Al Quran menggunakan SSD MobileNet v2

Arrie Kurniawardhani \*, Ihya Fathurrahman

<sup>1,2</sup>Informatika, Universitas Islam Indonesia, Sleman, Yogyakarta, 55584, Indonesia

### Info Artikel

#### Riwayat Artikel:

Received 2024-09-16

Revised 2024-10-09

Accepted 2024-10-17

**Abstract** – Tajweed contains a set of rules for reciting the Qur'an correctly. These rules must be complied with to ensure each letter is pronounced accurately. Arabic script and language compose the Qur'an, yet not all readers are fluent in Arabic. Tajweed serves as a guide to prevent readers from making mistakes when reciting the Qur'an that could alter the meaning. However, Tajweed rules are quite numerous and diverse, causing readers to struggle in memorizing these rules. To address this issue, a preliminary development of a Quran reading assistance system will be established, focusing on detecting Tajweed rules in images of Quranic text. SSD MobileNet v2, a Deep Learning technique for object detection, will be utilized for detecting Tajweed rules. The development of the Tajweed rule identification model begins with the data collection stage by capturing screens of the Al-Quran text pages from the Kemenag Qur'an Application. A total of 520 collected data were divided into 80:10:10 for training, validation, and test data, respectively. All data were subsequently annotated and enclosed in bounding boxes using the tool labellmg. The pre-trained model, SSD MobileNet V2 FPNLite 320x320, was used as the initial weight configuration of the model. Then the identification model was constructed during the training stage using training and validation data. The reliability of the constructed model was tested using test data. The test results indicated that the model could successfully recognize two Tajweed rules, Mad Aridlisukun and Mad Layyin, achieving the minimum loss around 0.15 and the maximum precision around 0.96.

**Keywords:** Deep Learning; Images of Quranic text; Object Detection; SSD MobileNet v2; Tajweed.

#### Corresponding Author:

Suttichai Premrudeeprechacharn

Email: suttichai@mail.com



This is an open access article under the [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/) license.

**Abstrak** – Hukum Tajwid berisi tata cara bagaimana membaca Al Qur'an dengan benar. Hukum tersebut perlu diketahui dan dipelajari agar setiap hurufnya dapat diucapkan dengan benar dan dengan bacaan panjang pendek yang tepat pula. Al Qur'an ditulis dengan huruf Arab dan berbahasa Arab, namun tidak semua pembaca Al Qur'an fasih berbahasa Arab. Tajwid menjadi panduan agar pembaca terhindar dari kesalahan saat membaca Al Qur'an yang dapat berakibat mengubah arti. Namun, hukum Tajwid cukup banyak dan bervariasi, tidak jarang pembaca mengalami kesulitan dalam memahami dan menghafal hukum-hukum tersebut. Untuk mengatasi hal tersebut, identifikasi hukum Tajwid pada citra teks Al Qur'an yang menjadi pengembangan awal dari sistem pendampingan membaca Al Qur'an akan dibangun. Huruf-huruf yang mengandung hukum Tajwid akan diidentifikasi menggunakan SSD MobileNet v2. SSD MobileNet v2 merupakan salah satu teknik Deep Learning untuk deteksi objek dengan kompleksitas komputasi cukup ringan. Pengembangan model identifikasi hukum Tajwid diawali dengan tahap pengumpulan data citra teks Al Qur'an yang diperoleh dari tangkapan layar Aplikasi Qur'an Kemenag. Sejumlah 520 data yang terkumpul dibagi menjadi 80:10:10 untuk data latih, data validasi, dan data uji. Seluruh data tersebut kemudian diberi label dan bounding box menggunakan labellmg. Pre-trained model, SSD MobileNet V2 FPNLite 320x320, dimanfaatkan untuk konfigurasi bobot awal model. Kemudian model identifikasi dibangun pada tahap pelatihan dengan data latih dan data validasi. Model yang terbentuk diuji keahliannya menggunakan data uji. Hasil uji coba menunjukkan model dapat mengenali dua hukum Tajwid, Mad Aridlisukun dan Mad Layyin, dengan loss terendah di sekitar 0.15 dan precision tertinggi di sekitar 0.96.

**Kata Kunci:** Citra Teks Al Qur'an, Deep Learning, Deteksi Objek, SSD MobileNet v2, Tajwid

## I. PENDAHULUAN

Ilmu tajwid menjadi salah satu ilmu penting bagi umat muslim untuk memandu mereka dalam membaca Al Quran dengan benar. Al Quran ditulis dalam bahasa dan huruf Arab, sedangkan tidak semua pembaca berbahasa Arab. Tajwid berisi panduan mengenai cara pelafalan setiap huruf yang tepat dan hukum-hukum bacaan [1-2]. Tujuan mempelajarinya adalah agar terhindar dari kesalahan membaca Al Quran di mana dampak terburuknya adalah berubahnya makna. Tidak semua pembaca dapat mempelajari ilmu tajwid dengan mudah sehingga pembaca menjadi ragu dan was-was saat membaca Al Quran. Hal tersebut diakibatkan salah satunya karena kesulitan dalam memahami dan menghafal hukum-hukum tersebut.

Untuk mengatasi masalah tersebut, penelitian kali ini akan memanfaatkan teknologi Machine Learning agar dapat mengenali hukum tajwid secara otomatis pada teks Al Quran berbasis citra. Solusi tersebut terinspirasi dari

aplikasi Mathway yang dapat memberikan solusi jawaban dari suatu persamaan matematika, hanya dengan memotret persamaan tersebut [3]. Penelitian pengenalan hukum tajwid menjadi tahap awal pengembangan aplikasi yang dapat mendampingi pengguna dalam belajar membaca Al Quran sesuai kaidah Tajwid. Dari penelitian tersebut ke depannya diharapkan dapat dikembangkan suatu aplikasi yang dapat membaca teks Al Quran sesuai kaidah tajwid dan membandingkan hasil pembacaan tersebut dengan pelafalan pembaca. Jika terjadi kesalahan pelafalan, aplikasi akan memberi peringatan ke pembaca.

Beberapa penelitian telah diusulkan agar dapat mengidentifikasi hukum Tajwid melalui citra Al Quran. Bray Curtis Distance (BCD) digunakan untuk mendeteksi dua hukum tajwid yaitu Tanwin Izhar dan Qalqalah, pada citra Al Quran [4]. BCD digunakan untuk mengukur kemiripan dua citra, citra acuan dan citra uji, berdasarkan vektor ciri. Vektor ciri diperoleh dari hasil deteksi tepi objek. Hasil uji coba menunjukkan akurasi pendeteksian rata-rata mencapai 7.8%. Tiga tahun kemudian, penulis yang sama membandingkan BCD dengan Euclidean Distance (ED) untuk mendeteksi hukum Tajwid Ikhfa Syafawi dan Idgham Mimi [5]. Hasil uji coba menunjukkan, ED lebih efisien dibandingkan BCD dengan akurasi pendeteksian rata-rata mencapai 82%. Pada penelitian selanjutnya, dengan langkah-langkah *pre-processing* dan ekstraksi ciri yang sama, penulis yang sama mengaplikasikan Fuzzy Associative Memory (FAM) untuk mengenali tujuh pola Tajwid Idgham Mutajansain [6]. Hasil uji coba menunjukkan, FAM dapat mencapai akurasi pendeteksian sekitar 80%.

Metode pengenalan pola lain, k-Nearest Neighbour (k-NN), juga diimplementasikan untuk mengenali dua jenis hukum, Idgham Maal Ghunnah and Idgham Bila Ghunnah [7]. Berbeda dengan sebelumnya, penelitian kali ini menggunakan Teknik Thinning dan Flip di tahap *pre-processing*, serta pengidentifikasi kata untuk memisahkan satu kata dengan kata yang lain. Kemudian, enam ciri bentuk seperti major axis length, minor axis length, eccentricity, filled area, solidity, dan perimeter diekstraksi untuk menjadi masukkan di k-NN. Hasil uji pada 180 citra menunjukkan bahwa teknik yang diusulkan dapat mencapai akurasi pendeteksian hingga 84%. Dalam penelitian lain, Fuzzy K-Nearest Neighbor (fK-NN) yang merupakan pengembangan metode k-NN dengan Fuzzy juga diimplementasikan untuk mengenali hukum Tajwid [8]. Selain itu, penelitian tersebut juga menerapkan Local Mean pada k-NN, sehingga jarak Euclidian tidak hanya dihitung antar data tetapi juga antar kelas target dan berdasarkan derajat keanggotaan di masing-masing kelas. Hasil uji pada 180 citra menunjukkan bahwa teknik yang diusulkan dapat mencapai akurasi pendeteksian hingga 96%.

Metode Deep Learning, You Only Look Once (YOLO) v8, diimplementasikan untuk mendeteksi sepuluh hukum Tajwid, menggunakan 210 citra Al Qur'an [9]. Sejumlah 66% data digunakan untuk pelatihan, 28% untuk validasi, dan 11% untuk pengujian. Data dengan dan tanpa augmentasi dibandingkan untuk mengetahui scenario yang paling optimal. Hasil uji coba menunjukkan bahwa data yang diaugmentsi dapat mencapai kinerja paling tinggi dengan mAP50 sebesar 92.9%.

Single Shot Detector (SSD) MobileNet v2 juga merupakan salah satu teknik di Deep Learning yang digunakan untuk mendeteksi objek [10-11]. SSD MobileNet v2 dibuat untuk mengurangi kompleksitas komputasi agar lebih ringan dan dapat dijalankan di *embedded systems* dengan sumber daya komputasi terbatas. Oleh karena itu, SSD MobileNet v2 akan dimanfaatkan pada penelitian kali ini, agar dapat menghasilkan arsitektur pendeteksian Tajwid yang lebih ringan sehingga ke depannya dapat diimplementasikan ke perangkat bergerak, seperti *smartphone*.

## II. METODE

Pengembangan model identifikasi huruf yang memiliki hukum Tajwid dengan SSD MobileNet v2 pada citra teks Al Qur'an terdiri atas delapan langkah seperti yang ditunjukkan pada Gambar 1. Kedelapan langkah tersebut yaitu pengumpulan data, pembagian data, pelabelan data, pemilihan pre-trained model, pengaturan hyperparameter, pelatihan model, evaluasi, dan pengujian.

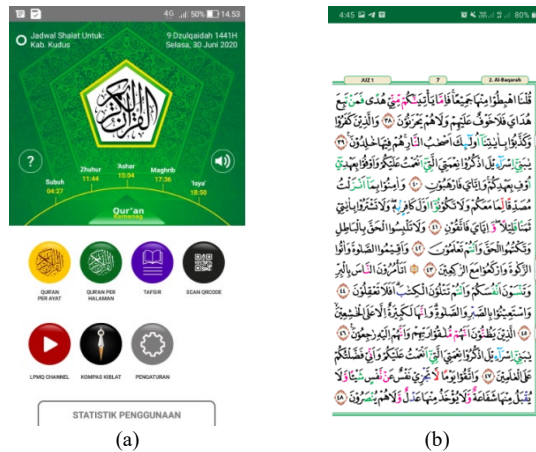


Gambar 1. Langkah-langkah pengembangan model identifikasi hukum Tajwid dengan SSD MobileNet v2.

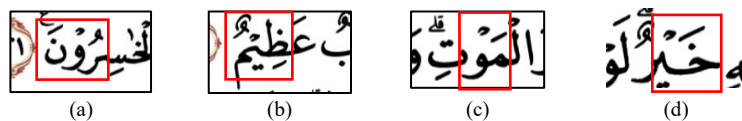
### A. Pengumpulan Data

Data berupa citra teks Al Qur'an diperoleh dari hasil tangkapan layar *smartphone* pada aplikasi Qur'an Kemenag yang dirilis oleh Lajnah Pentahsinan Mushaf Al-Quran [12-13]. Jenis *font* dalam citra teks Al-Qur'an tersebut sama dengan jenis *font* LMPQ Isep Misbah yang menjadi jenis *font* standar pada cetakan Al-Quran di Indonesia [14-15]. Citra teks Al Qur'an hasil tangkapan layar berukuran 1080x2340 pixel dengan ekstensi .jpg. Aplikasi Qur'an Kemenag ditunjukkan pada Gambar 2.

Dua hukum yang ingin dikenali adalah Mad Aridlisukun dan Mad Layyin [1-2]. Mad Aridlisukun dan Mad Layyin adalah bagian dari hukum Mad di mana suatu huruf akan dibaca lebih penjang karena suatu sebab. Mad Aridlisukun adalah hukum untuk membaca panjang dua atau empat atau enam ketukan (harakat), apabila huruf mad (alif <sup>ا</sup>, wawu <sup>و</sup>, ya <sup>ي</sup>) bertemu dengan huruf yang disukunkan (◌◌) karena berhenti (waqaf). Contoh Mad Aridlisukun ditunjukkan pada Gambar 3(a) dan Gambar 3(b). Sedangkan Mad Layyin adalah hukum untuk membaca panjang dua atau empat atau enam harakat, apabila di akhir suatu kalimat terdapat huruf Mad Layyin, yaitu wawu sukun (◌◌) atau ya sukun (◌◌), yang didahului oleh huruf yang bertanda (berharakat) fathah (◌◌). Contoh Mad Layyin ditunjukkan pada Gambar 3(c) dan Gambar 3(d).



Gambar 2. Aplikasi Qur'an Kemenag (a) Halaman depan, (b) Halaman teks Al Qur'an.



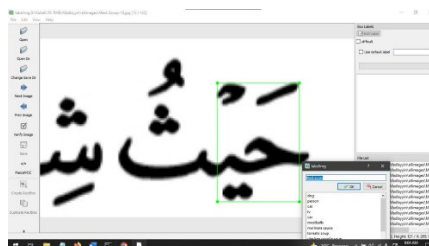
Gambar 3. Contoh Mad (a) Mad Aridlisukun wawu, (b) Mad Aridlisukun ya, (c) Mad Layyin wawu, (d) Mad Layyin ya.

### B. Pembagian Data

Dari hasil proses pengumpulan data diperoleh 520 objek dengan perbandingan 50% Mad Aridlisukun dan 50% Mad Layyin. Data tersebut kemudian dibagi menjadi dua dengan perbandingan 80% untuk pelatihan dan 20% untuk pengujian [11]. Selanjutnya, data pengujian dibagi dua dengan proporsi yang sama untuk validasi dan pengujian.

### C. Pelabelan Data

Tahap pelabelan bertujuan untuk memberi tanda di mana letak objek atau huruf yang mengandung hukum Tajwid pada suatu citra. Masing-masing data citra akan dilabeli dengan cara menggambar *bounding box* pada huruf-huruf yang mengandung hukum Tajwid dan memberi anotasi pada *bounding box* tersebut dengan label berupa nama hukum Tajwid. Pelabelan dilakukan menggunakan *tools* bernama 'labelImg' [16]. Hasil dari proses pelabelan berupa berkas dengan ekstensi .xml yang berisi informasi nama label dan koordinat posisi dari *bounding box* huruf yang bertajwid. Contoh proses pelabelan dengan 'labelImg' ditunjukkan pada Gambar 4.



Gambar 4. Pelabelan huruf yang mengandung hukum Tajwid menggunakan *tools* 'labelImg'.

#### D. Pemilihan Pre-Trained Model

Pada penelitian kali ini, arsitektur SSD MobileNet v2 tidak akan dilatih dari awal, namun akan menggunakan model yang telah dilatih dengan data lain yang lebih umum sifatnya (pre-trained model) [17]. Langkah ini dapat mempercepat proses pelatihan karena model telah mengenali karakteristik umum sebagian besar objek. Pre-trained model untuk SSD MobileNet dapat dipilih dari berbagai jenis model yang terdapat di TensorFlow 2 Detection Model Zoo [18]. Model-model tersebut telah dilatih pada dataset COCO 2017 [19-20]. Pre-trained model yang digunakan pada penelitian kali ini adalah SSD MobileNet V2 FPNLite 320x320. Selanjutnya, di tahap pelatihan akan dilakukan *fine tuning*, penyesuaian model atau bobot-bobot dari arsitektur SSD MobileNet v2, dengan data yang ingin dideteksi.

#### E. Pengaturan Hyperparameter

Sebelum pelatihan dimulai ada beberapa nilai parameter (hyperparameter) yang harus ditentukan terlebih dahulu. Hyperparameter yang ditentukan berkaitan dengan arsitektur model jaringan maupun algoritma pelatihan, seperti Learning Rate dan Batch Size [21]. Penentuan hyperparameter dapat mempengaruhi kinerja model yang akan dibentuk, sehingga harus dicari konfigurasi yang paling optimal. Learning rate yang umum digunakan adalah 0.001. Untuk mengetahui kinerja model pada nilai Learning Rate yang lain, maka nilai Learning Rate dicoba pada nilai lain yaitu 0.01 dan 0.1. Smaller Batch Size akan digunakan pada penelitian kali ini agar dapat berjalan dengan baik pada spesifikasi GPU yang digunakan yaitu 2, 4, dan 8. Nilai Learning Rate yang rendah akan dipasangkan dengan nilai Batch Size yang rendah pula, agar proses proses pelatihan dapat berjalan dengan stabil. Sehingga untuk mencari nilai hyperparameter yang tepat pada kasus deteksi Tajwid kali ini, tiga konfigurasi nilai akan dicoba. Nilai Learning Rate dan Batch Size untuk konfigurasi pertama adalah 0.001 dan 2, untuk konfigurasi kedua adalah 0.01 dan 4, dan untuk konfigurasi ketiga adalah 0.1 dan 8. Ketiga konfigurasi dijalankan dengan nilai *num steps* sebanyak 10000.

#### F. Pelatihan

SSD MobileNet v2 terdiri dari dua bagian arsitektur. Arsitektur pertama bertugas untuk mengekstraksi ciri menggunakan MobileNet v2 [22]. Arsitektur kedua bertugas untuk mengklasifikasikan atau melokalisasi posisi objek pada citra menggunakan SSD [23]. Data yang digunakan pada tahap ini adalah citra pada data latih dan data validasi. Proses pelatihan akan menghasilkan model yang paling optimal berupa informasi bobot dari arsitektur SSD MobileNet v2.

#### G. Evaluasi

Untuk mengetahui kehandalan, kinerja model SSD MobileNet v2 diukur menggunakan Classification Loss dan Localization Loss [24]. Localization Loss mengukur ketidakakuratan *bounding box* yang digambar oleh model, sedangkan Classification Loss mengukur ketidakakuratan model dalam mengidentifikasi label dari suatu objek yang berada di dalam *bounding box*. Semakin kecil nilai-nilai tersebut maka semakin handal model SSD MobileNet v2 yang diperoleh. Selain itu, Accuracy, Precision, Recall, dan F-1 Score juga diukur untuk mengetahui kinerja model pada masing-masing label [25]. Semakin besar nilai-nilai tersebut maka semakin handal model SSD MobileNet v2 yang diperoleh.

#### H. Pengujian

Pada tahap pengujian, model paling optimal yang diperoleh pada hasil pelatihan digunakan untuk mendeteksi objek pada data citra yang belum pernah dikenali sebelumnya. Hal ini bertujuan untuk mengetahui kemampuan generalisasi suatu model. Data yang digunakan pada tahap ini adalah citra pada data uji.

### III. HASIL DAN PEMBAHASAN

Pada penelitian kali ini, SSD MobileNet v2 dibangun menggunakan Python 3.7.10 dengan TensorFlow 2.5.0. Pre-trained model, SSD MobileNet V2 FPNLite 320x320 yang diperoleh dari TensorFlow 2 Detection Model Zoo [18], dimanfaatkan untuk menentukan konfigurasi bobot awal dari model. Selanjutnya model tersebut dilatih pada tiga konfigurasi Learning Rate dan Batch Size yang berbeda, untuk mencari nilai yang tepat pada kasus identifikasi hukum Tajwid, Mad Aridlisukun dan Mad Layyin, dengan data berupa citra teks Al Qur'an.

Hasil pelatihan dari tiga konfigurasi ditunjukkan pada Gambar 5. Gambar 5 menunjukkan perubahan nilai *loss* selama pelatihan dari epoch ke-0 sampai ke-10000. Terlihat bahwa nilai *loss* terus mengalami penurunan seiring bertambahnya epoch walaupun di beberapa tempat terjadi osilasi. Classification Loss untuk Konfigurasi 1 ditunjukkan

pada Gambar 5(a), Classification Loss Konfigurasi 2 ditunjukkan pada Gambar 5(b), Classification Loss Konfigurasi 3 ditunjukkan pada Gambar 5(c), Localization Loss Konfigurasi 1 ditunjukkan pada Gambar 5(d), Localization Loss Konfigurasi 2 ditunjukkan pada Gambar 5(e), dan Localization Loss Konfigurasi 3 ditunjukkan pada Gambar 5(f).

Di langkah ke-10000, Classification Loss pada model dengan Konfigurasi 2 dan 3 berhasil turun hingga di sekitar 0.04, sedangkan model dengan Konfigurasi 1 masih berada pada di sekitar 0.1. Hal yang sama terjadi pada Localization Loss, pada langkah ke-10000, model dengan Konfigurasi 2 berhasil turun hingga di bawah 0.02. Walau tidak di bawah 0.02 model dengan Konfigurasi 3 berhasil turun hingga di sekitar 0.02. Sedangkan model dengan Konfigurasi 1 masih berada pada di sekitar 0.1.

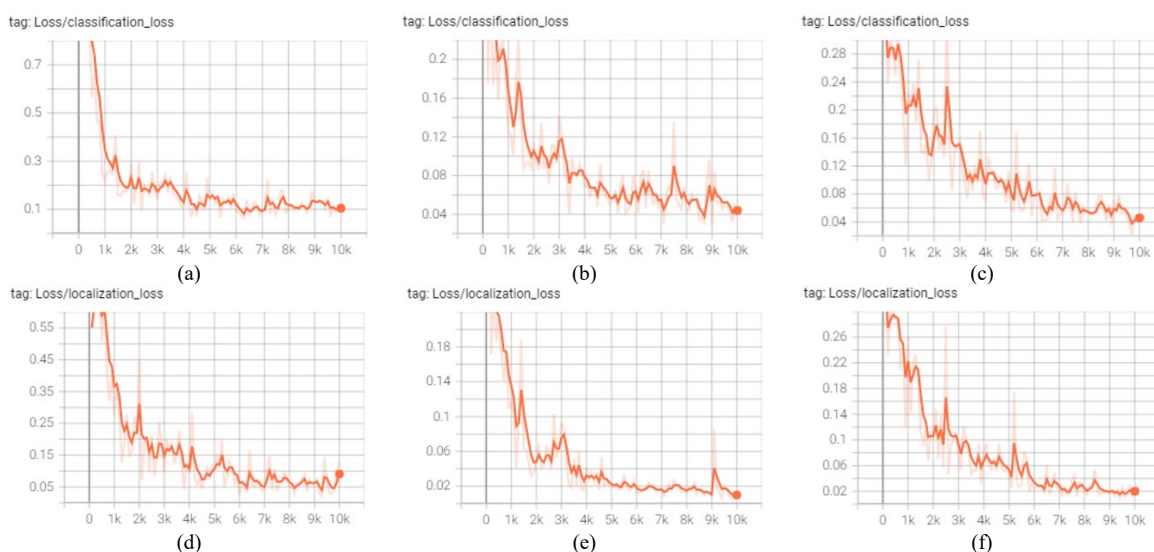
Grafik Classification Loss dan Localization Loss memiliki sifat yang sama pada masing-masing konfigurasi. Grafik nilai loss milik model dengan Konfigurasi 1 mengalami osilasi dengan rentang nilai 0.7-0.1 pada Clasification Loss dan 0.55-0.05 pada Localization Loss. Berbeda jauh dengan Konfigurasi 1, grafik loss dari model dengan Konfigurasi 3 hanya memiliki rentang nilai sekitar 0.24. Tidak jauh berbeda dengan Konfigurasi 3, rentang nilai loss dari grafik model dengan Konfigurasi 2 memiliki rentang nilai sekitar 0.16.

Pada Konfigurasi 1, nilai loss turun secara tajam pada 1000 langkah awal. Pada langkah selanjutnya, loss model Konfigurasi 1 turun secara perlahan namun belum dapat mencapai serendah loss di Konfigurasi 2 hingga langkah ke-10000. Hal ini dapat disebabkan karena nilai Learning Rate 0.001 membuat proses pembelajaran model (*update* bobot) lebih lambat, sehingga model membutuhkan langkah yang lebih banyak untuk mencapai nilai loss yang sama dengan Konfigurasi 2. Pada Konfigurasi 2, nilai loss model sudah berada di sekitar 0.15 di langkah ke-1000 dan dapat mencapai loss terendahnya di sekitar langkah ke-9000. Hampir sama dengan Konfigurasi 2, model dengan Konfigurasi 3 dapat mencapai nilai loss sekitar 0.2 di langkah ke-1000 dan dapat mencapai loss terendahnya di sekitar langkah ke-10000, namun nilainya belum serendah Konfigurasi 2.

Konfigurasi 1 dengan Learning Rate yang relatif kecil menunjukkan penurunan loss yang relative lama dan belum dapat mencapai nilai loss serendah Konfigurasi 2 dan 3 pada epoh ke 10000. Konfigurasi 2 dapat mencapai konvergen lebih awal. Hal ini menunjukkan Learning Rate 0.01 dapat belajar lebih cepat dengan pengurangan Classification dan Localization loss lebih cepat. Konfigurasi 3 menunjukkan proses penurunan loss yang hampir sama dengan Konfigurasi 2, namun lebih beresilasi. Hal ini menunjukkan bahwa model di Konfigurasi 3 mengalami kesulitan untuk konvergen secara stabil pada Learning Rate yang besar.

Terlihat bahwa seiring bertambahnya nilai Learning Rate dari 0.001, 0.01, ke 0.1, proses konvergen juga bertambah lebih cepat. Namun pada Learning Rate yang paling tinggi, model mengalami kesulitan dalam hal stabilitas, yang ditunjukkan seringnya osilasi dan tidak teraturnya pembelajaran. Sedangkan pada Learning Rate yang paling rendah, model membutuhkan waktu lebih lama dalam belajar dan belum dapat mencapai nilai loss terendahnya.

Nilai Batch Size memengaruhi stabilitas pembelajaran namun tidak terlalu signifikan seperti laju pembelajaran. Nilai Batch Size yang lebih kecil, pada Konfigurasi 1, menghasilkan pembelajaran yang lebih lambat namun lebih stabil. Sedangkan nilai Batch Size yang lebih besar, pada Konfigurasi 2 dan 3, dengan Learning Rate yang lebih tinggi menunjukkan pola pembelajaran yang lebih cepat namun lebih tidak stabil. Kombinasi nilai pertengahan pada Learning Rate dan Batch Size di Konfigurasi 2 menunjukkan kinerja terbaik pada model identifikasi hukum Tajwid kali ini dengan konvergensi yang cepat dan stabil.



Gambar 5. Grafik hasil pelatihan SSD MobileNet v2, (a) Classification Loss Konfigurasi 1, (b) Classification Loss Konfigurasi 2, (c) Classification Loss Konfigurasi 3, (d) Localization Loss Konfigurasi 1, (e) Localization Loss Konfigurasi 2, (f) Localization Loss Konfigurasi 3.

Perbandingan dari hasil pelatihan di masing-masing konfigurasi berdasarkan nilai Accuracy, Average Precision, Averagr Recall, dan F-1 Score ditunjukkan pada Tabel 1. Terlihat bahwa Konfigurasi 2 dengan nilai Learning Rate 0.01 dan Batch Size 4 memiliki kinerja paling optimal diantara Konfigurasi 1 dan 3. Accuracy, Average Precision, Averagr Recall, dan F-1 Score milik Konfigurasi 2 dapat mencapai nilai lebih dari 0.95. Kinerja dari model pada Konfigurasi 3 sedikit lebih baik dari Konfigurasi 1, dengan selisih nilai hanya sekitar 0.01. Berdasarkan Tabel 1, nilai Learning Rate dan Batch Size tidak berbanding lurus dengan nilai Accuracy, Precision, Recall, maupun F-1 Score. Nilai Learning Rate yang kecil belum tentu menghasilkan nilai Accuracy yang tinggi, begitu pula sebaliknya. Oleh sebab itu, nilai hyperparameter yang tepat perlu dicari.

TABEL 1  
PERBANDINGAN KINERJA SSD MOBILENET V2 PADA TIGA KONFIGURASI LEARNING RATE DAN BATCH SIZE

Konfigurasi	Learning Rate	Batch Size	Accuracy	Average Precision	Average Recall	F-1 score
1	0.001	2	0.9038	0.9093	0.9038	0.9035
2	0.01	4	<b>0.9615</b>	<b>0.9643</b>	<b>0.9615</b>	<b>0.9615</b>
3	0.1	8	0.9038	0.9194	0.9038	0.9029

Model SSD MobileNet v2 yang terbentuk dari hasil pelatihan selanjutnya diuji pada data uji, data yang belum pernah dilihat oleh model sebelumnya. Hal ini bertujuan untuk mengukur kemampuan model dalam mengenali data baru. Hasil pengujian model, yang ditampilkan menggunakan *confusion matrix*, dari Konfigurasi 1 ditunjukkan pada Tabel 2, Konfigurasi 2 ditunjukkan pada Tabel 3, dan Konfigurasi 3 ditunjukkan pada Tabel 4. Aktual adalah label sesungguhnya atau seharusnya, sedangkan prediksi adalah label yang diberikan oleh model SSD MobileNet v2.

Dari hasil pengujian pada Tabel 2-4 terlihat bahwa SSD MobileNet v2 dapat mengenali hukum Tajwid Mad Aridlisukun dengan mudah. Model pada Konfigurasi 2 dan 3 dapat mengenali semua huruf ber hukum Mad Aridlisukun, sedangkan model Konfigurasi 1 hanya salah satu. Kesalahan tersebut berupa tidak dapat mengidentifikasi keberadaan huruf yang mengandung hukum Mad Aridlisukun. Walau masih terdapat kesalahan dalam mengenali hukum Tajwid Mad Layyin, namun kesalahan identifikasi tidak terlalu besar. Konfigurasi 2 lebih unggul dalam mengenali hukum Tajwid Mad Layyin, di mana hanya 2 dari 26 data yang salah dikenali. Dua data tersebut dikenali sebagai Mad Aridlisukun. Konfigurasi 3 paling lemah dalam mengenali hukum Tajwid Mad Layyin, di mana 4 dari 26 data dikenali sebagai Mad Aridlisukun dan 1 data tidak dapat dikenali sama sekali.

Hasil pengujian yang dapat mengenali huruf ber hukum Tajwid dengan benar ditunjukkan pada Gambar 6. Gambar 6(a) adalah *ground truth* atau gambar acuan di mana SSD MobileNet v2 diharapkan dapat mengenali hukum Tajwid Mad Aridlisukun dengan memberi *bounding box* berwarna merah di posisi hukum tersebut muncul. Hasil deteksi hukum Tajwid Mad Aridlisukun dari SSD MobileNet v2 yang ditunjukkan dengan *bounding box* berwarna kuning pada konfigurasi 1 ditunjukan pada Gambar 6(b), untuk konfigurasi 2 ditunjukan pada Gambar 6(c), dan untuk konfigurasi 3 ditunjukan pada Gambar 6(d). Terlihat bahwa posisi ketiga *bounding box* berwarna kuning tersebut hamper sama dengan posisi *bounding box* berwarna merah. Sehingga dapat dikatakan model berhasil mendeteksi hukum Tajwid Mad Aridlisukun.

Pada huruf ber hukum Mad Layyin, walaupun keberhasilan sudah mencapai di atas 80%, namun model masih belum dapat mengidentifikasi dengan tepat. Model dengan konfigurasi 2 menjadi model paling baik dalam mengenali Mad Layyin. Kesalahan identifikasi hanya terjadi pada dua data, di mana huruf yang seharusnya ber hukum Mad Layyin diidentifikasi sebagai huruf ber hukum Mad Aridlisukun.

TABEL 2  
CONFUSION MATRIX UNTUK MODEL SSD MOBILENET V2 KONFIGURASI 1

		Prediksi		
		Mad Aridlisukun	Mad Layyin	Tidak Terdeteksi
Aktual	Mad Aridlisukun	25	0	1
	Mad Layyin	2	22	2

TABEL 3  
CONFUSION MATRIX UNTUK MODEL SSD MOBILENET V2 KONFIGURASI 2

		Prediksi		
		Mad Aridlisukun	Mad Layyin	Tidak Terdeteksi
Aktual	Mad Aridlisukun	26	0	0
	Mad Layyin	2	24	0

TABEL 4  
CONFUSION MATRIX UNTUK MODEL SSD MOBILENET V2 KONFIGURASI 3

		Prediksi		
		Mad Aridlisukun	Mad Layyin	Tidak Terdeteksi
Aktual	Mad Aridlisukun	26	0	0
	Mad Layyin	4	21	1



Gambar 6. Contoh Hasil Deteksi Hukum Tajwid Mad Aridlisukun dengan SSD MobileNet v2 (a) Ground Truth, (b) Model Konfigurasi 1, (c) Model Konfigurasi 2, (d) Model Konfigurasi 3.

#### IV. SIMPULAN

Mengenali hukum-hukum Tajwid menjadi salah satu tantangan saat membaca Al Qur'an. Kegagalan dalam mematuhi hukum-hukum Tajwid dapat mengubah makna sehingga membuat pembaca was-was saat membaca Al Qur'an. Untuk membantu pembaca dalam mengenali hukum-hukum Tajwid, penelitian kali ini mengembangkan pendekatan berbasis Deep Learning yang dapat mendeteksi huruf yang mengandung hukum Tajwid. Pendekatan tersebut memanfaatkan SSD MobileNet v2 untuk mengemali dua hukum Tajwid, Mad Aridlisukun dan Mad Layyin. Model dari SSD MobileNet v2 diuji coba pada tiga konfigurasi yang berbeda. Model konfigurasi kedua dengan nilai Learning Rate 0.01 dan Batch Size 4 dapat mencapai kinerja yang paling optimal dengan nilai loss 0.15 dan nilai precision 0.96. Dengan nilai Learning Rate 0.1 dan Batch Size 8, nilai precision dari model konfigurasi ketiga lebih rendah 0.05 dari konfigurasi kedua, namun nilai loss tidak jauh berbeda. Model konfigurasi pertama dengan nilai Learning Rate 0.001 dan Batch Size 2 memiliki kinerja paling rendah. Untuk pengembangan selanjutnya, algoritma akan dikembangkan agar dapat mengidentifikasi hukum-hukum Tajwid lain, seperti Ikhfa, Idgham, dan lainnya. Selain itu, data pengujian akan ditambahkan jumlah dan variasinya. Kelanjutan penelitian tersebut diharapkan dapat mempelajari apakah model tetap dapat mencapai kinerja yang baik saat jumlah hukum-hukum Tajwid yang harus dikenali menjadi lebih banyak dan saat data pengujian lebih bervariasi.

#### DAFTAR PUSTAKA

- [1] M. A. Amir, *Ilmu Tajwid Praktis*. Indonesia: Pustaka Baitul Hikmah Harun Ar-Rasyid, 2019. [Online]. Available: <https://books.google.co.id>
- [2] A. Yahyaa, *Foundation of Tajweed: Learning How to Read the Holy Quraan on the Foundation of Tajweed*, 2nd ed., Los Angeles, USA, 2016. [Online]. Available: <https://books.google.co.id>
- [3] Chegg, Inc., "Mathway: Scan & Solve Problems," *play.google.com*, Sep. 2024. [Online]. Available: <https://play.google.com/store/apps/details?id=com.bagatrix.mathway.android&hl=en&pli=1>
- [4] T. Adek, R. Fadlisyah, and M. Muhathir, "Detection system tajwid al quran on image using bray curtis distance," *Int. J. Comput. Technol.*, vol. 2, no. 8, pp. 293–300, Aug. 2015.
- [5] D. Hamdhana, R. Fadlisyah, and S. Adani, "Sistem Pendeteksi Pola Tajwid Al-Qur'an Hukum Ikhfa Syafawi Dan Idgham Mimi Pada Citra Menggunakan Metode Euclid Distance Dan Bray Curtis Distance," *TECHSI-J. Tek. Inform.*, vol. 10, no. 2, pp. 109–119, 2018.
- [6] R. Rizal, B. Bustami, and D. Azzahra, "Pendeteksi Tajwid Idgham Mutajansain Pada Citra Al-Qur'an Menggunakan Fuzzy Associative Memory (FAM)," *TECHSI-J. Tek. Inform.*, vol. 11, no. 3, pp. 395–407, 2019.
- [7] S. Ibrahim, F. A. A. Rahim, and Z. Rahmad, "Automatic Tajweed Rules Recognition using k-Nearest Neighbour (k-NN)," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2S11, 2019.
- [8] A. Hafizh, "Model Fuzzy K-Nearest Neighbor dengan Local Mean pada Pengenalan Pola Citra Tajwid," Ph.D. dissertation, Univ. Sumatera Utara, 2017.
- [9] D. I. Mulyana, and M. A. I. Rowis, "Optimization of Text Mining Detection of Tajweed Reading Laws Using the Yolov8 Method on the Qur'an," *Qalamuna J. Pendidik. Sos. Agama*, vol. 14, no. 2, pp. 1089–1110, 2022.
- [10] Y. Chiu, C. Tsai, M. Ruan, G. Shen, and T. Lee, "Mobilenet-SSDv2: An improved object detection model for embedded systems," in *2020 Int. Conf. Syst. Sci. Eng. (ICSSE)*, 2020, pp. 1-5.
- [11] W. Kurdthongmee, "A comparative study of the effectiveness of using popular DNN object detection algorithms for pith detection in cross-sectional images of parawood," *Heliyon*, vol. 6, no. 2, 2020.
- [12] Lajnah Pentashihan Mushaf Al-Qur'an, "Qur'an Kemenag," *play.google.com*, Apr. 2023. [Online]. Available: <https://play.google.com/store/apps/details?id=com.quran.kemenag&hl=id>
- [13] Kemenag, "Aplikasi Qur'an Kemenag Makin Lengkap dengan Fitur Baru," *kemenag.go.id*, Jun. 2020. [Online]. Available: <https://kemenag.go.id/nasional/aplikasi-quran-kemenag-makin-lengkap-dengan-fitur-baru-66tnua>
- [14] A. H. Muzakky, "Al-Qur'an Di Era Gadget: Studi Deskriptif Aplikasi Qur'an Kemenag," *J. Studi Al-Qur'an*, vol. 16, no. 1, pp. 55–68, 2020.
- [15] I. Fathurrahman and A. Kurniawardhani, "Pengenalan hukum tajwid pada citra Al-Quran: kajian pustaka," *AUTOMATA*, vol. 2, no. 1, 2021.
- [16] tzatalin, "labelimg," *pypi.org*, Oct. 2021. [Online]. Available: <https://pypi.org/project/labelimg/>
- [17] S. A. Sanchez, H. J. Romero, and A. D. Morales, "A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework," in *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 844, no. 1, p. 012024, 2020.

- [18] V. Birodkar, "TensorFlow 2 Detection Model Zoo," *github.com*, May. 2021. [Online]. Available: [https://github.com/tensorflow/models/blob/master/research/object\\_detection/g3doc/tf2\\_detection\\_zoo.md](https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md)
- [19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 740–755.
- [20] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1209–1218.
- [21] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay," *arXiv preprint arXiv:1803.09820*, 2018.
- [22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.
- [23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 21–37.
- [24] Y. Wu, Y. Chen, L. Yuan, Z. Liu, L. Wang, H. Li, and Y. Fu, "Rethinking classification and localization for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10186–10195.
- [25] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.