

BAB II TINJAUAN PUSTAKA

2.1 Retak Jalan Raya

Jalan merujuk kepada semua komponen jalan, termasuk infrastruktur tambahan dan fasilitas pendukungnya yang digunakan oleh masyarakat umum untuk transportasi, dan dapat berlokasi pada berbagai tingkat, mulai dari permukaan tanah, hingga kedalaman bawah tanah dan air, serta di atas permukaan air. Ini tidak termasuk jalur kereta api dan jalur kabel [22]. Secara umum, kerusakan pada jalan dapat digolongkan menjadi dua kategori, yaitu kerusakan yang bersifat struktural dan kerusakan yang bersifat fungsional. Kerusakan struktural melibatkan kegagalan pada struktur perkerasan atau kerusakan pada beberapa komponen perkerasan. Jenis kerusakan ini mengakibatkan perkerasan tidak lagi mampu menahan beban dari kendaraan yang melintas. Sementara itu, kerusakan fungsional merupakan jenis kerusakan yang mempengaruhi aspek keamanan dan kenyamanan para pengguna jalan. Beberapa contoh kerusakan jalan meliputi retak-retak pada permukaan jalan dan gangguan pada marka jalan. [23].

Retak kulit buaya adalah jenis keretakan yang menyerupai pola kulit buaya. Kerusakan ini mencakup berbagai istilah seperti retak kulit buaya, *chickenwire cracks*, *alligator cracks*, *polygonal cracks*, dan *crazing*. Retakan ini memiliki lebar minimal sekitar 3 mm dan biasanya terhubung membentuk serangkaian pola kotak-kotak kecil yang mirip dengan kulit buaya atau jaring ayam. Wilayah di mana retak kulit buaya terjadi dapat bervariasi dalam ukuran, dengan beberapa daerah yang memiliki kerusakan tersebut cenderung lebih luas daripada yang lain. Hal ini disebabkan oleh beban lalu lintas berulang yang melebihi kapasitas lapisan permukaan untuk menahan beban tersebut.

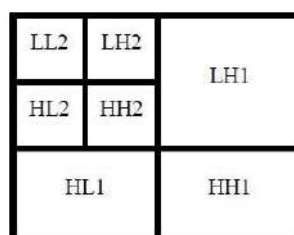
Retak garis adalah jenis keretakan yang berbentuk garis dan dapat terjadi dalam arah memanjang, melintang, atau diagonal. Beberapa jenis kerusakan retak yang termasuk dalam kategori ini meliputi retak di tepi perkerasan, retak di pertemuan antara perkerasan dan bahu jalan, retak di sambungan antara segmen jalan, dan retak di

sambungan saat pelebaran jalan. Biasanya, retak garis mengacu pada keretakan yang terletak di sisi tepi perkerasan dekat bahu jalan dan memiliki bentuk garis yang membentang, kadang-kadang dengan cabang-cabang yang mengarah ke bahu. Retak ini bisa terdiri dari beberapa celah yang berjajar sejajar..

2.2 Transformasi Wavelet

Ekstraksi fitur adalah tahap di mana citra mengalami perubahan agar dapat mengungkapkan informasi yang tersembunyi dengan lebih jelas. Transformasi citra, dalam konteks ini, merujuk pada perubahan citra dari satu domain ke domain lainnya. Melalui langkah transformasi ini, citra dapat dijelaskan sebagai hasil kombinasi linear dari sinyal dasar yang biasanya disebut sebagai fungsi dasar [24].

Wavelet merupakan gelombang singkat atau gelombang kecil. Transformasi Wavelet adalah proses yang mengubah suatu sinyal menjadi rangkaian gelombang-gelombang singkat. Gelombang-gelombang singkat ini adalah fungsi yang memiliki lokasi pada waktu yang berbeda. Transformasi Wavelet memiliki kapasitas untuk mengenali informasi tentang frekuensi yang terkandung dalam sinyal dan juga menawarkan wawasan tentang skala, durasi, atau waktu. Wavelet digunakan untuk memeriksa karakteristik bentuk sinyal gelombang yang timbul dari perpaduan antara faktor waktu atau skala serta faktor frekuensi. Berikut tahapan proses dan hasil dari transformasi Wavelet pada ilustrasi yang ada di Gambar 2.1.



Gambar 2.1 Dekomposisi Wavelet

Gambar 2.1 menggambarkan proses dekomposisi citra setelah menjalani transformasi Wavelet. Dalam fase ini, citra awal dibagi menjadi empat sub-citra baru yang akan menggantikannya. Masing-masing sub-citra ini memiliki ukuran yang merupakan sepertiga dari citra asli. Tiga sub-citra yang terletak di sudut kanan atas,

sudut kanan bawah, dan sudut kiri bawah akan menampilkan tingkat detail yang lebih tinggi dibandingkan dengan citra aslinya karena mereka mengandung komponen frekuensi tinggi dari citra asli. Sebaliknya, sub-citra yang berada di sudut kiri atas akan lebih menyerupai citra asli dan lebih halus karena mengandung komponen frekuensi rendah dari citra asli tersebut. Selanjutnya, sub-citra di sudut kiri atas yang berisi frekuensi rendah juga dapat dibagi menjadi empat sub-citra baru. Proses ini dapat diulangi sesuai dengan tingkat transformasi yang diterapkan.

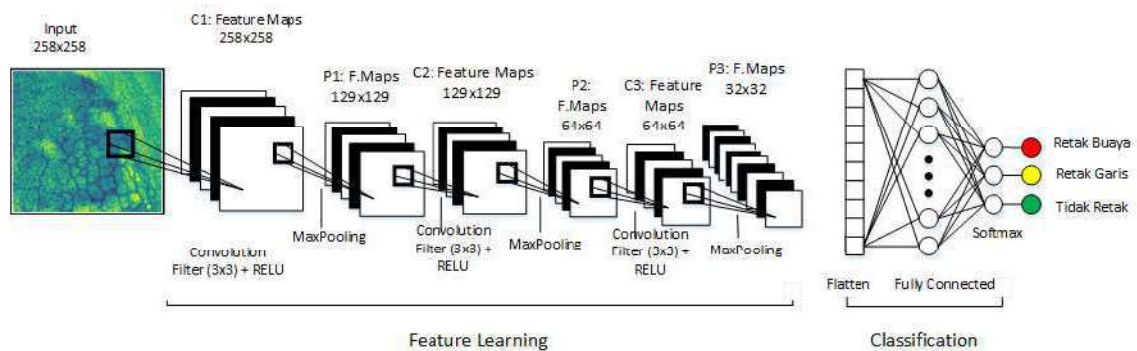
Dalam situasi citra dua dimensi, terdapat dua metode yang digunakan untuk mengubah atau memisahkan nilai-nilai piksel, yakni pendekatan standar dan pendekatan yang tidak biasa. Kedua metode ini berdasarkan pada transformasi Wavelet 1 dimensi. Pendekatan dekomposisi standar melibatkan penerapan transformasi Wavelet 1 dimensi terlebih dahulu pada setiap baris citra, dan kemudian pada setiap kolom citra. Dengan kata lain, citra pertama-tama diubah menggunakan transformasi Wavelet pada baris, dan kemudian hasilnya diubah lagi menggunakan transformasi Wavelet pada kolom. Proses ini dapat diiterasikan sesuai dengan tingkat transformasi yang diinginkan. Sementara itu, dalam pendekatan dekomposisi tidak standar, transformasi Wavelet 1 dimensi diterapkan bergantian antara kolom citra dalam langkah pertama, dan langkah ini diulang seiring dengan tingkat yang diinginkan. Pendekatan ini memberikan fleksibilitas dalam proses dekomposisi dan dapat menghasilkan representasi citra yang berbeda dari pendekatan standar.

2.3 Convolution Neural Network

Convolutional Neural Network (CNN) merupakan perkembangan dari metode *Multi Layer Perceptron* (MLP) [26]. CNN memiliki lebih banyak dimensi daripada MLP. CNN menerima *input* dalam bentuk array dengan dua dimensi atau lebih. Sebagai contoh, jika kita memproses gambar sepotong buah tomat, dimensi pertama mungkin menggambarkan bentuk umum tomat dan orientasinya. Sedangkan dimensi kedua mungkin merepresentasikan fitur-fitur penting seperti bagian depan dan belakang, kelopak bunga, serta tekstur permukaan tomat.

Setiap lapisan (*layer*) dalam CNN memiliki banyak fitur yang terbentuk dari lapisan sebelumnya. Fitur-fitur ini bisa mencakup informasi tentang sudut, tekstur, hingga lekukan yang signifikan. Proses konvolusi diterapkan pada setiap *filter* dengan ukuran kernel yang ditentukan. Proses *pooling* sering kali digunakan bersamaan dengan proses konvolusi. Tujuan dari proses *pooling* adalah untuk mengambil nilai perwakilan, seperti nilai maksimum, dari satu kernel.

Selain itu, metode dropout dapat diterapkan pada beberapa lapisan CNN. *Dropout* berfungsi untuk secara acak menghapus beberapa fitur yang tidak digunakan dalam proses pelatihan, membantu mencegah *overfitting*. Tahapan proses CNN dapat dilihat pada Gambar 2.2 yang menggambarkan arsitektur CNN secara umum.



Gambar 2.2 *Convolution Neural Network (CNN)*

Berdasarkan Gambar 2.2, terlihat bahwa dalam penelitian ini, model yang dikembangkan memiliki satu lapisan yang dapat menjalankan tiga metodologi secara berurutan. Hasil dari lapisan pertama disimpan dan digunakan sebagai *input* untuk lapisan kedua. Lapisan kedua kemudian dapat menjalankan metode yang sama seperti yang dilakukan oleh lapisan pertama. Jumlah lapisan yang akan digunakan dapat disesuaikan dengan kebutuhan spesifik penelitian.

Keputusan mengenai jumlah lapisan dan jumlah neuron dalam setiap lapisan adalah penting. Semakin banyak lapisan dan neuron yang ditentukan, semakin rumit modelnya, tetapi ini juga akan meningkatkan waktu komputasi yang diperlukan untuk satu iterasi metode. Oleh karena itu, penentuan jumlah lapisan dan neuron harus

mempertimbangkan *trade-off* antara kinerja model dan waktu komputasi yang tersedia. CNN mencapai tingkat akurasi yang tinggi karena memiliki sejumlah besar fitur yang diekstraksi melalui proses konvolusi, jumlah neuron, dan koneksi antar neuron yang disusun dengan bantuan bobot yang diperbarui secara berkala. Akurasi yang tinggi dapat terwujud melalui penggabungan yang optimal dari semua elemen tersebut. Kombinasi terbaik ini sering kali menjadi titik fokus utama dan dapat menghasilkan akurasi yang sangat baik. Kombinasi ini sering mengalami modifikasi dan penyesuaian untuk meningkatkan kinerja model:

- a. Ukuran dari operasi konvolusi biasanya dibatasi untuk mengendalikan jumlah lapisan pada model. Semakin banyak iterasi konvolusi yang digunakan dalam jaringan, semakin lama waktu yang dibutuhkan untuk melatih dan menjalankan model tersebut. Di sisi lain, jika kita menggunakan terlalu sedikit iterasi, ini dapat memengaruhi kemampuan model untuk mendekati kebenaran atau akurasi yang diharapkan, karena jumlah fitur yang digunakan oleh model menjadi terbatas. Oleh karena itu, pemilihan ukuran dan kedalaman konvolusi harus memperhitungkan keseimbangan antara performa dan efisiensi komputasi yang diinginkan.
- b. Ukuran kernel pada operasi konvolusi berfungsi sebagai sub-*matrix* yang digunakan untuk melakukan operasi konvolusi pada *input*. Pada dasarnya, ukuran kernel ini memungkinkan kita untuk merubah sejumlah nilai dalam *matrix input* menjadi satu nilai dalam *matrix output* dengan melakukan perkalian dan penjumlahan.
- c. Semakin kecil ukuran kernel yang digunakan, semakin detail informasi yang dapat diambil dari *input*, dan ini dapat menghasilkan representasi yang lebih halus dan detail pada citra atau data. Namun, memang benar bahwa semakin kecil ukuran kernel, semakin banyak perhitungan yang diperlukan, yang pada gilirannya dapat membuat waktu komputasi semakin lama. Oleh karena itu, pemilihan ukuran kernel harus mempertimbangkan keseimbangan antara keakuratan representasi dan efisiensi komputasi yang diinginkan.
- d. Jumlah lapisan dalam Convolutional Neural Network (CNN) berperan sebagai penampung hasil konvolusi dari *input*. Setiap lapisan dalam CNN memiliki tugas

khusus dalam mengekstraksi fitur-fitur yang semakin kompleks dari *input*. Semakin banyak lapisan yang ditambahkan dalam arsitektur CNN, semakin banyak pula fitur yang dapat dihasilkan dan dipelajari oleh model.

- e. Namun, perlu diingat bahwa semakin banyak lapisan dalam jaringan juga akan meningkatkan kompleksitas model, dan ini dapat memengaruhi waktu komputasi yang diperlukan baik untuk pelatihan maupun inferensi (pengujian). Selain itu, penggunaan lapisan yang terlalu dalam juga dapat meningkatkan risiko *overfitting* jika tidak diatur dengan baik. Dalam merancang CNN, perlu dilakukan penyesuaian yang cermat untuk menemukan keseimbangan antara jumlah lapisan yang digunakan, kompleksitas model, dan waktu komputasi yang tersedia. Ini akan memastikan bahwa model memiliki performa yang baik tanpa mengorbankan efisiensi waktu komputasi. Jumlah lapisan Fully-Connected bertugas mengintegrasikan fitur-fitur yang telah diekstraksi ke dalam kelas yang sesuai. Metodenya adalah dengan memberikan nilai perkalian acak dari bobot dan bias. Ketika nilai ini masih jauh dari yang diharapkan, perkalian acak tersebut diperbarui dan diulang hingga ditemukan bobot dan bias yang optimal untuk mencapai pendekatan kelas yang tepat. Waktu yang dibutuhkan untuk proses ini cukup bervariasi, tergantung pada jumlah fitur yang dihasilkan.
- f. Lapisan *Pooling* adalah komponen dalam *Convolutional Neural Network* (CNN) yang bertugas mengurangi dimensi fitur-fitur hasil ekstraksi. Melalui proses *pooling*, informasi yang paling relevan dari suatu wilayah dalam fitur-fitur tersebut diambil, sementara dimensi yang kurang relevan dihilangkan. Dalam *pooling*, biasanya digunakan metode seperti pengambilan nilai maksimum atau perhitungan nilai rata-rata untuk menggambarkan wilayah tersebut [26]

Dari penjelasan sebelumnya, terdapat banyak parameter dalam Convolutional Neural Network (CNN) yang dapat disesuaikan agar mencapai tingkat presisi dan akurasi yang tinggi, sambil juga meminimalkan waktu yang dibutuhkan. Ini dapat dicapai dengan mencari kombinasi parameter yang optimal. Hasil dari pengoptimalan parameter pada suatu dataset tertentu dapat diaplikasikan pada dataset lain, yang sering disebut sebagai *transfer learning*. Gabungan dari arsitektur *transfer learning* yang

sangat terkenal adalah Googlenet dan Alexnet. Meskipun keduanya memiliki konfigurasi model yang berbeda-beda, keduanya mampu mencapai akurasi yang sangat baik. Arsitektur Googlenet dan Alexnet telah menunjukkan performa yang sangat tinggi dan telah diuji coba pada jutaan gambar. Meskipun ada perbedaan dalam parameter *layer* dan *filter* di kedua arsitektur ini, konsep dasarnya yang digunakan adalah sama, yaitu *Convolutional Neural Network* (CNN).

2.4 YOLO

Saat ini, deteksi objek melibatkan penggunaan classifier untuk mengidentifikasi objek dalam citra dan mengevaluasinya pada berbagai lokasi dan skala. Sistem *Deformable Parts Model* (DPM) mengadopsi metode jendela geser di mana *classifier* dijalankan secara merata di seluruh citra. Di sisi lain, *Regional Convolution Neural Network* (R-CNN) menggunakan pendekatan proposal wilayah untuk pertama-tama menghasilkan kotak pembatas yang berpotensi di dalam citra dan kemudian menjalankan *classifier* pada proposal-proposal tersebut. Setelahnya, proses klasifikasi melibatkan *langkah post-processing* yang digunakan untuk menyaring kotak pembatas, menghilangkan deteksi objek yang berlebihan, dan membandingkan kotak prediksi dengan objek lainnya. Proses yang kompleks ini memerlukan waktu yang lama dan sulit untuk dioptimalkan karena setiap komponen harus dilatih secara terpisah.

YOLO mengubah pendeteksian objek menjadi sebuah permasalahan regresi tunggal, yang mengolah informasi secara langsung dari piksel gambar hingga mendapatkan koordinat kotak pembatas dan probabilitas kelasnya. YOLO dikenal sebagai sistem "*you only look once*" yang berarti bahwa sistem hanya perlu melihat gambar sekali untuk memprediksi objek apa yang ada di dalamnya dan di mana objek tersebut berada.

2.5 Evaluasi

Evaluasi performa suatu sistem klasifikasi adalah aspek yang krusial. Performa sistem klasifikasi mencerminkan sejauh mana kemampuan sistem dalam mengelompokkan data. Confusion matrix adalah salah satu alat yang digunakan untuk

mengukur performa metode klasifikasi tertentu. [28]. Pengukuran kinerja *Confussion matrix* memiliki 4 istilah representasi hasil proses klasifikasi yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) seperti pada tabel 2.1.

Tabel 2.1 Definisi Parameter TP, FP, FN, TN

<i>True label</i>	<i>Prediction Label</i>	
	TP	FN
	FP	TN

Berdasarkan Tabel 2.1 dijelaskan tentang definisi dari parameter TP, FP, FN, TN yaitu parameter yang ada dalam pengujian *confussion matrix*. TP adalah data dengan kelas positif yang terklasifikasi positif, TN adalah data dengan kelas negatif yang terklasifikasi negatif, FP adalah data dengan kelas negatif yang terklasifikasi positif dan FN adalah data dengan kelas negatif yang terklasifikasi negatif. Hasil *output* dari *confussion matrix* adalah *accuracy*, *precision*, *recall*, dan *F1score* dengan persamaan sebagai berikut.

Accuracy adalah seberapa akurat metode klasifikasi dalam menentukan kelas atau label pada data. Persamaan matematis untuk *accuracy* dapat dilihat pada Persamaan (2.1)

$$Accuracy = \frac{(TP)+(TN)}{total\ citra} \times 100\% \quad (2.1)$$

Persamaan (2.1) merupakan nilai *true positive* atau dapat didefinisikan sebagai data dengan kelas positif yang terklasifikasi positif. Sedangkan (TN) merupakan nilai *true negatif* atau bisa didefinisikan sebagai data dengan kelas negatif yang terklasifikasi negatif. *Accuracy* merupakan hasil bagi antara penjumlahan TP dan TN dengan jumlah keseluruhan data citra kemudian hasil tersebut dikali 100% untuk memperoleh nilai *accuracy* dalam persen.

Precision adalah nilai fraksi tiap kelas yang diklasifikasi dengan benar. Persamaan matematis untuk *precision* dapat dilihat pada Persamaan (2.2)

$$Precision = \frac{(TP)}{(TP)+(FP)} \times 100\% \quad (2.2)$$

Persamaan (2.2) merupakan nilai *true* positif atau dapat didefinisikan sebagai data dengan kelas positif yang terklasifikasi positif atau hasil bagi antara TP dengan hasil penjumlahan TP dan FP kemudian hasil tersebut dikali 100% untuk memperoleh nilai *precision* dalam persen. Persamaan matematis lain untuk menghitung *precision* seperti pada Persamaan (2.3)

$$Precision = \frac{(TN)}{(TN)+(FN)} \times 100\% \quad (2.3)$$

Persamaan (2.3) merupakan nilai *true* negatif atau dapat didefinisikan sebagai data dengan kelas negatif yang terklasifikasi negatif. *Precision* merupakan hasil bagi antara TN dengan hasil penjumlahan TN dan FN kemudian hasil tersebut dikali 100% untuk memperoleh nilai *precision* dalam persen.

Recall adalah fraksi jumlah kelas atau label yang berhasil didapatkan dibagi dengan jumlah kelas yang seharusnya didapatkan. Persamaan matematis untuk *recall* dapat dilihat pada Persamaan (2.4)

$$Recall = \frac{(T+)}{(T+)+(F-)} \times 100\% \quad (2.4)$$

Persamaan (2.4) merupakan nilai *true* positif atau dapat didefinisikan sebagai data dengan kelas positif yang terklasifikasi positif. Sedangkan *recall* merupakan nilai *true* negatif atau bisa didefinisikan sebagai data dengan kelas positif yang terklasifikasi negatif. *Recall* merupakan hasil bagi antara TP dengan hasil penjumlahan TP dan FN kemudian hasil tersebut dikali 100% untuk memperoleh nilai *recall* dalam persen. Persamaan matematis lain untuk menghitung *precision* seperti pada Persamaan (2.5)

$$Recall = \frac{(TN)}{(TN)+(FP)} \times 100\% \quad (2.5)$$

Persamaan (2.5) TN merupakan nilai *true* negatif atau dapat didefinisikan sebagai data dengan kelas negatif yang terklasifikasi negatif. *Recall* merupakan hasil bagi antara TN dengan hasil penjumlahan TN dan FP kemudian hasil tersebut dikali 100% untuk memperoleh nilai *recall* dalam persen.

F1score adalah fungsi *harmonic mean* dari *precision* dan *recall*. Persamaan matematis untuk *f1score* dapat dilihat pada Persamaan (2.6)

$$F1score = 2 \frac{Precision \ Recall}{Precision + Re} \times 100\% \quad (2.6)$$

Berdasarkan Persamaan (2.6) diketahui jika *f1score* merupakan 2 kali hasil bagi antara perkalian *precision recall* dengan penjumlahan antara *precision recall* yang kemudian hasil tersebut dikali 100% untuk mendapatkan nilai dalam persen.

2.6 Kajian Pustaka

Kajian pustaka membahas mengenai rangkuman dari beberapa penelitian sebelumnya yang digunakan sebagai referensi dan acuan untuk penelitian ini. Beberapa penelitian tersebut membahas mengenai metode pengolahan citra untuk deteksi retak pada jalan raya. Berikut penjabaran dari beberapa penelitian tersebut:

1. Penelitian sebelumnya memanfaatkan konsep penginderaan jauh dengan teknologi jaringan syaraf tiruan deep learning. YOLO digunakan untuk melakukan deteksi kerusakan jalan, dan hasil pendeteksian ini kemudian dikombinasikan dengan informasi posisi atau lokasi yang diperoleh melalui GNSS. Dengan demikian, hasil deteksi dapat memberikan informasi yang akurat tentang posisi atau lokasi kerusakan. Hasil penelitian ini menciptakan sebuah model identifikasi kerusakan jalan yang memiliki tingkat akurasi keseluruhan sebesar 88% dan tingkat akurasi kappa sebesar 86%, sementara untuk akurasi posisi sebaran kerusakan, koordinat posisinya memiliki akurasi RMSE sekitar $\pm 5,6$ meter. [28].
2. Penelitian selanjutnya yang membahas mengenai deteksi retak jalan deteksi retak perkerasan dan mengusulkan metode baru untuk mendeteksi retak perkerasan menggunakan *deep learning* dengan *transfer learning*. Penelitian ini menganalisis kinerja model yang diusulkan untuk arsitektur jaringan yang berbeda yaitu, *googlenet*, *alexnet* dan *resent* dan menyimpulkan bahwa Googlenet memberikan kinerja yang lebih baik dalam mendeteksi keretakan jalan raya [29].

3. Penelitian berikutnya tentang deteksi retak jalan menggunakan metode *localized thresholding technique* untuk melakukan klasifikasi. Metode ini tidak terbatas pada jenis tekstur perkerasan, dan metode ini efisien karena membutuhkan waktu kurang dari 20 detik per gambar untuk menghasilkan hasil. Metode ini diuji pada 130 gambar permukaan beton semen *portland* dan beton aspal. Hasil pengujian memperoleh hasil evaluasi yaitu *presisi* 0,89 sedangkan *recall* 0,83, dan *F1score* 0,86 serta kurasi pengukuran panjang retak 80% [30].
4. Penelitian selanjutnya yang menggunakan metode *multi-scale retinex fused* dengan *Wavelet transform*. Algoritma *multi-scale retinex fused* dapat menutupi kekurangan dari transformasi *Wavelet*, sehingga kombinasi keduanya dapat memperoleh efek peningkatan retakan yang lebih baik. Selain itu, *preprocessing* berupa penghilangan bayangan dilakukan sebelum peningkatan retakan, yang secara efektif menghilangkan gangguan bayangan intensitas tinggi. Melalui perbandingan kinerja objektif, metode ini dapat menyoroti informasi retakan dengan lebih baik. Metode yang diusulkan secara efektif mewujudkan fungsi penghilangan bayangan dan peningkatan retak, sehingga pengenalan akurasi sistem deteksi secara keseluruhan mencapai 95,8%, menunjukkan bahwa algoritma tersebut memiliki signifikansi penelitian yang tinggi [31].
5. Penelitian berikutnya bertujuan untuk menggunakan CNN untuk mendeteksi mendeteksi keretakan jalan. Data dalam bentuk gambar telah digunakan sebagai *input*, *preprocessing*, dan segmentasi ambang batas diterapkan pada data masukan. *Output* yang diproses menggunakan CNN untuk ekstraksi fitur dan klasifikasi. Akurasi pelatihan sebesar 96,20%, akurasi validasi menjadi 96,50%, dan pengujian akurasi menjadi 94,5% [32].

Berdasarkan penjabaran dari enam penelitian yang telah dijabarkan, dapat diperoleh masukan untuk penelitian yang akan dilakukan yaitu merancang sistem untuk mendeteksi retak pada jalan raya menggunakan metode pengolahan citra. Metode pengolahan citra yang digunakan adalah *Wavelet transform* untuk ekstraksi ciri dan *Convolution Neural Network* untuk ekstraksi ciri dan klasifikasi.