

## Deep Convolutional Neural Network untuk Mendeteksi Retak pada Permukaan Beton yang Memiliki Void

Patrick Nicholas Hadinata<sup>1\*</sup>, Djoni Simanta<sup>1</sup> and Liyanto Eddy<sup>1</sup>

Dikirim: 22/08/2021

Diterima: 04/10/2021

### ABSTRAK

*Convolutional neural network* berbasis *encoder-decoder* telah dirancang dan dilatih menggunakan *dataset* eksternal untuk mendeteksi retak pada permukaan beton yang relatif sederhana. Namun, pada kenyataannya permukaan beton memiliki banyak fitur seperti *void* pada permukaan yang disebabkan oleh udara yang terperangkap saat proses pencampuran beton. Oleh karena itu, pada penelitian ini kemampuan *convolutional neural network* akan diteliti lebih lanjut untuk mendeteksi retak pada permukaan beton yang memiliki *void*. Tujuan pertama penelitian ini adalah menguji model yang dilatih dengan *dataset* eksternal pada permukaan beton ber-*void*. Jika model tidak berhasil membedakan *void* dengan retak, maka tujuan kedua penelitian ini adalah menyusun *dataset* pelatihan internal baru yang secara khusus membedakan *void* dengan retak, yang kemudian akan ditambahkan pada *dataset* eksternal untuk diinvestigasi performanya. Penelitian ini menggunakan arsitektur U-Net dan arsitektur DeepLabV3+ sebagai *encoder-decoder* untuk mengoperasikan *semantic image segmentation*. Model *encoder-decoder* yang dilatih dengan *dataset* eksternal tidak berhasil membedakan *void* dengan retak saat pengujian. Maka, *dataset* internal yang terdiri dari gambar beton ber-*void* dibentuk dan digabungkan dengan *dataset* eksternal. Dengan penambahan *dataset* internal yang baru, hasil pengujian menunjukkan bahwa model berhasil membedakan *void* dengan retak pada permukaan beton. U-Net mencapai nilai F1 sebesar 85,92%, sedangkan DeepLabV3+ mencapai nilai F1 sebesar 84,09%.

**Kata kunci:** *convolutional neural network*; DeepLabV3+; deteksi retak; U-Net; *void*

### 1. PENDAHULUAN

Deteksi retak pada permukaan beton merupakan langkah penting untuk menilai kondisi infrastruktur. Retak yang tidak terdeteksi dapat berakibat buruk pada kesehatan infrastruktur. Secara umum, deteksi retak pada permukaan beton dilakukan secara manual. Metode ini memiliki beberapa keterbatasan, yaitu biaya yang mahal, keamanan yang kurang terjamin, dan bersifat subjektif terhadap tenaga ahli. Oleh karena itu, metode yang otomatis dan lebih terukur dapat dirancang menggunakan teknik penglihatan komputer.

*Convolutional neural network* (CNN) merupakan salah satu metode kecerdasan buatan berbasis penglihatan komputer yang dirancang secara khusus untuk mengekstrak fitur spasial dari data gambar. Deskripsi matematika kompleks dari gambar dapat diekstrak menggunakan metode konvolusi dengan teknik *sliding window*. Fitur yang diekstrak dari gambar dapat disebut sebagai *feature map*.

Ketelitian tingkat piksel dapat dicapai menggunakan algoritme *semantic image segmentation*. CNN dapat disusun menjadi *encoder-decoder* untuk melakukan *semantic image segmentation*. *Encoder* akan melakukan *downsampling* pada *feature map* untuk mereduksi kompleksitas dimensi gambar dan menyimpan fitur penting. *Decoder* akan melakukan *upsampling* untuk

<sup>1</sup> Universitas Katolik Parahyangan, Jl. Ciumbuleuit No.94, Bandung 40141

\*pnhadinata@gmail.com

mengembalikan dimensi gambar menjadi ukuran semula. Hasil dari *decoder* adalah *feature map* berupa prediksi kelas masing-masing piksel pada gambar.

Terdapat banyak arsitektur *encoder-decoder* yang telah dikembangkan. Terdapat dua arsitektur yang populer digunakan untuk memecahkan berbagai masalah penglihatan komputer, yaitu U-Net [1] dan DeepLabV3+ [2]. U-Net memenangkan ISBI *cell tracking challenge* 2015 dengan nilai tertinggi. DeepLabV3+ mencapai nilai tertinggi dalam PASCAL VOC 2012 *dataset* dengan mIoU 89%. U-Net dan DeepLabV3+ telah dibandingkan performanya dalam bidang deteksi retak menggunakan *dataset* eksternal [3] dengan hasil U-Net berperforma lebih baik dari DeepLabV3+ [4]. Namun, gambar permukaan beton pada *dataset* eksternal memiliki fitur permukaan yang relatif sederhana, sedangkan pada kenyataannya permukaan beton seringkali memiliki fitur permukaan seperti *void*. Oleh karena itu, penelitian ini bertujuan untuk menginvestigasi kemampuan model yang dilatih dengan *dataset* eksternal dalam membedakan *void* dengan retak. Jika model tidak berhasil membedakan *void* dengan retak, maka *dataset* internal tambahan yang terdiri dari foto beton dengan *void* permukaan akan disusun untuk mengatasi keterbatasan model sebelumnya. Performa model sebelum dan sesudah penambahan *dataset* internal akan dibandingkan dari segi parameter evaluasi F1 dan hasil pengujian gambar. U-Net dan DeepLabV3+ juga akan dibandingkan kembali performanya setelah penambahan *dataset* internal yang baru.

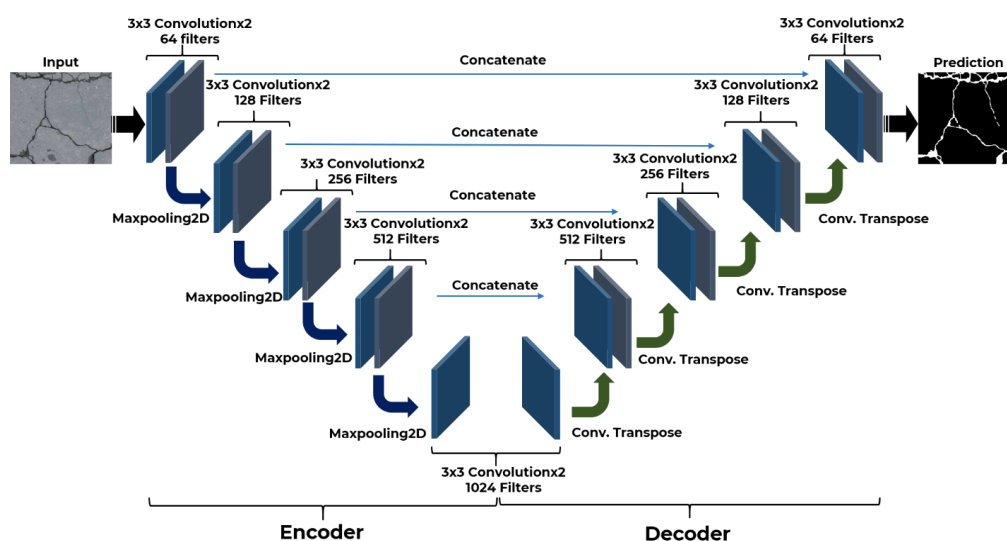
## 2. TINJAUAN PUSTAKA

### Deep Learning dan Convolutional Neural Network

*Deep learning* adalah kemampuan komputer untuk mempelajari pola-pola yang terdapat dari suatu kumpulan data. *Deep learning* banyak diterapkan pada teknik pembelajaran *supervised* ataupun pembelajaran *unsupervised*. Beberapa contoh penerapan *deep learning* adalah pengenalan suara, penglihatan komputer, analisis deret waktu, regresi non-linear, dan klasifikasi data. Salah satu jenis algoritme *deep learning* yang secara khusus disusun untuk menganalisis data gambar adalah CNN. CNN menggunakan teknik *sliding window* untuk mengekstrak fitur level tinggi dari gambar. Pada penelitian ini, fitur yang secara khusus akan diekstrak guna analisis gambar adalah fitur retak pada permukaan beton.

### Arsitektur U-Net

Arsitektur U-Net memiliki ciri khas berbentuk huruf U simetri. Arsitektur ini memperkenalkan *layer concatenate* pada setiap level untuk mentransfer informasi spasial dari *encoder* ke *decoder* secara multilevel.

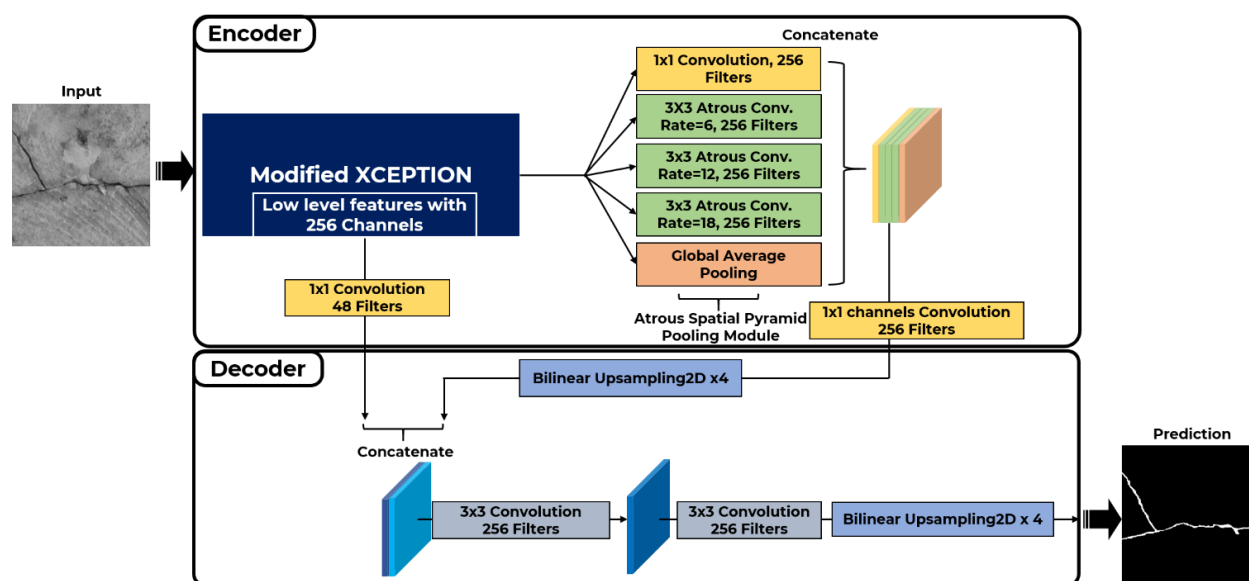


Gambar 1. Arsitektur U-Net

Fitur *concatenate* multilevel ini menghasilkan segmentasi gambar yang halus dan akurat. *Encoder* U-Net terdiri dari *layer convolution2D* dengan *kernel*  $3 \times 3$  dan *maxpooling* dengan *kernel*  $2 \times 2$ . Jumlah filter *convolution2d* dilipatgandakan dari 64 filter di *layer* pertama menjadi 1024 filter di *layer bottleneck*. *Decoder* U-Net terdiri dari *layer convolution2d* dengan *kernel*  $3 \times 3$  dan *transpose convolution* atau *deconvolution* dengan *kernel*  $2 \times 2$  dan *striding* 2. *Layer dropout* [5] dengan nilai 0,1–0,3 juga ditambahkan untuk mencegah *overfitting* dan meregularisasi *neural network*. Jumlah filter pada *decoder* sama seperti *encoder*. Ilustrasi dari U-Net dapat dilihat pada Gambar 1.

### Arsitektur DeepLabV3+

Arsitektur DeepLabV3+ dikembangkan untuk mengatasi kekurangan dari arsitektur DeepLabV3 [6]. DeepLabV3 menggunakan *naive decoder* dengan *conditional random field* (CRF). *Naive decoder* ini memiliki beban komputasi yang besar dengan lokalisasi yang kurang akurat. DeepLabV3+ menggunakan *backbone* Xception [7] yang dimodifikasi dan modul *atrous spatial pyramid pooling* (ASPP) sebagai *encoder*. *Low level feature map* dari Xception digabungkan dengan *output* ASPP untuk menghasilkan lokalisasi multiskala yang akurat. *Decoder* DeepLabV3+ terdiri dari *convolution2d* dengan *kernel*  $3 \times 3$  dan *upsampling2d* bilinear dengan nilai perbesaran 4. *Layer convolution2d* pada *decoder* memiliki jumlah filter 256. *Layer dropout* bernilai 0,1–0,3 juga ditambahkan pada arsitektur ini untuk regularisasi pelatihan. Ilustrasi dari DeepLabV3+ dapat dilihat pada Gambar 2.



Gambar 2. Arsitektur DeepLabV3+

### Optimizer dan Fungsi Loss

Proses pelatihan akan menggunakan *optimizer Adaptive Moment Estimation* (Adam) [9] dan fungsi *dice loss* [10]. *Optimizer* Adam telah banyak digunakan pada berbagai *image recognition task* dan telah terbukti sebagai salah satu *optimizer* dengan konvergensi terbaik dan tercepat. *Dice loss* telah terbukti memiliki konvergensi yang jauh lebih cepat dibandingkan fungsi *cross-entropy* umum. Fungsi *Dice loss* memperhatikan kelas dengan area yang lebih kecil, dimana hal ini berkaitan langsung dengan *highly unbalanced segmentation*. Retak pada permukaan memiliki area yang relatif jauh lebih kecil dibanding latar belakang. *Dice loss* secara langsung membuat variabel retak sebagai parameter utama dalam persamaannya. Fungsi Adam dapat dilihat pada Persamaan (1)-(3), sedangkan fungsi *dice loss* dapat dilihat pada Persamaan (4).

$$\hat{s}_j^i = \frac{\rho_1 s_j^{i-1} + (1-\rho_1)g_j}{1+\rho_1} \quad (1)$$

$$\hat{r}_j^i = \frac{\rho_2 r_j^{i-1} + (1-\rho_2)(g_j)^2}{1+\rho_2} \quad (2)$$

$$W_j^{i+1} = W_j^i + \frac{\epsilon \hat{s}_j}{\delta + \sqrt{\hat{r}_j}} \quad (3)$$

dengan  $W$  adalah pembaruan dari parameter *weight*,  $\epsilon$  adalah *learning rate*,  $\delta$  adalah konstanta bernilai  $10^{-7}$  untuk mencegah fungsi yang tidak terdefinisi,  $\rho_1$  dan  $\rho_2$  adalah *hyperparameter* untuk estimasi momen, dan  $g_j$  adalah gradien stokastik pada langkah pembelajaran tertentu. Pada studi ini, nilai *learning rate* untuk U-Net diambil sebesar  $4 \times 10^{-4}$  dengan *batch size* 8, sedangkan nilai *learning rate* untuk DeepLabV3+ diambil sebesar  $2 \times 10^{-4}$  dengan *batch size* 4. Nilai *learning rate* dan *batch* diambil sedemikian rupa agar penurunan gradien yang dihasilkan kedua arsitektur maksimal.

$$DL = 1 - F1 = 1 - \frac{2 \times TP + \gamma}{2 \times TP + FN + FP + \gamma} \quad (4)$$

dengan TP adalah *true positives*, FN adalah *false negatives*, FP adalah *false positives*, dan  $\gamma$  adalah faktor *smoothing* bernilai 1 untuk mencegah fungsi yang tidak terdefinisi.

### 3. METODOLOGI

#### Dataset Pelatihan

Kedua arsitektur di atas dilatih menggunakan 2 jenis *dataset*, yaitu *dataset* eksternal dan *dataset* gabungan. Penjelasan mengenai *dataset* eksternal dan *dataset* gabungan adalah sebagai berikut:

- Dataset* eksternal bersumber dari Yang, et al. (2018). *Dataset* eksternal terdiri dari beberapa kategori jenis retak dengan skema warna yang berbeda-beda. Gambar retak pada *dataset* eksternal memiliki lingkungan dan fitur permukaan yang relatif sederhana. Seperti yang terlihat pada Gambar 3, *dataset* eksternal tidak mengandung foto permukaan beton yang memiliki *void*. *Dataset* eksternal terdiri dari 776 gambar RGB beserta label yang berkaitan. Gambar RGB pada *dataset* ini memiliki rentang DPI dari 72 hingga 300.
- Dataset* gabungan terdiri atas *dataset* eksternal yang diambil dari Yang, et al. (2018) dan *dataset* internal yang dibuat sendiri. *Dataset* internal diambil menggunakan *smartphone* Samsung Galaxy Note 8 dengan resolusi kamera 9,1 Mega-Pixels dan bukaan diafragma f/1.7. Foto diambil menggunakan rasio 1 : 1 yang kemudian direduksi ukurannya menjadi  $256 \times 256$  piksel. Foto-foto yang telah diambil dilabelkan menggunakan *software* Adobe Photoshop CC 2017. Retak dilabelkan dengan warna putih yang berkomposisi 255 *red*, 255 *green*, dan 255 *blue*, sedangkan *foreground* dilabelkan sebagai warna hitam (0 *red*, 0 *green*, dan 0 *blue*). Ilustrasi dari proses pembuatan label dapat dilihat pada Gambar 4. Untuk mencegah bias yang tinggi karena keterbatasan data, *dataset* internal akan diperbanyak menggunakan teknik *dataset augmentation*. *Dataset augmentation* yang dipakai adalah rotasi  $90^\circ$  sebanyak 3 kali. Secara lugas, satu gambar akan diaugmentasi menjadi 4 gambar baru. Ilustrasi dari *data augmentation* dapat dilihat pada Gambar 5. *Dataset* internal disusun sebanyak 35 gambar yang kemudian diaugmentasi menjadi 140 gambar. Maka, total *dataset* internal adalah 140 gambar permukaan beton silinder yang memiliki retak dan *void*. *Dataset* internal yang dibuat digabungkan dengan *dataset* eksternal menjadi *dataset* gabungan. Maka, *dataset* gabungan memiliki total 916 gambar.

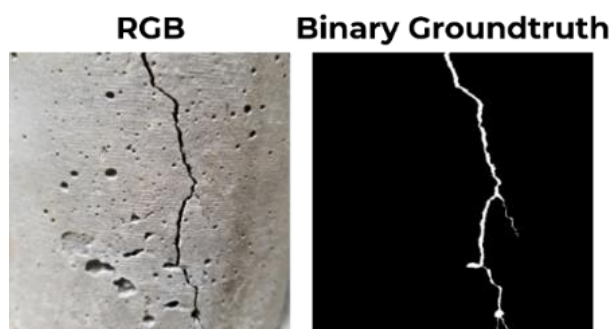
#### Parameter Evaluasi

Model akan dievaluasi menggunakan parameter akurasi, F1 [8], *precision* dan *recall*. F1 adalah nilai seimbang antara *precision* dan *recall*. F1 akan dijadikan parameter perbandingan utama

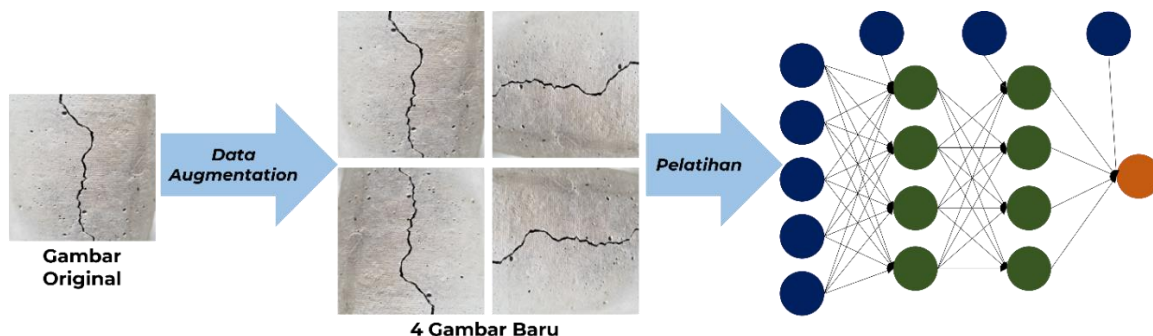
karena F1 memiliki fungsi yang hanya memperhitungkan *error* satu kelas yang ditentukan. Akurasi tidak akan dijadikan parameter perbandingan utama karena hasil akurasi akan cenderung berorientasi pada kelas yang lebih besar, yaitu *foreground*. Model U-Net dan model DeepLabV3+ yang telah dilatih dengan *dataset* eksternal akan dievaluasi untuk mendeteksi retak pada gambar-gambar *dataset* eksternal dan *dataset* gabungan. Hal yang sama juga dilakukan untuk model U-Net dan model DeepLabV3+ yang dilatih menggunakan *dataset* gabungan.



Gambar 3. Jenis Lingkungan Retak pada *Dataset* Eksternal



Gambar 4. Contoh Pasangan *Dataset* Internal (RGB *Input* dan *Groundtruth* Biner)



Gambar 5. *Data Augmentation*

**Konfigurasi Pelatihan**

Data gambar akan melalui proses normalisasi terlebih dahulu. Nilai piksel akan dibagi 255 agar piksel bernilai 0–1. Hal ini diperlukan untuk mempermudah konvergensi *neural network* dan menyesuaikan data *input* dengan nilai probabilitas dari *layer output*. Kemudian, semua gambar akan diubah ukurannya menjadi 256×256 piksel. Nilai 256 dipakai untuk menyesuaikan angka

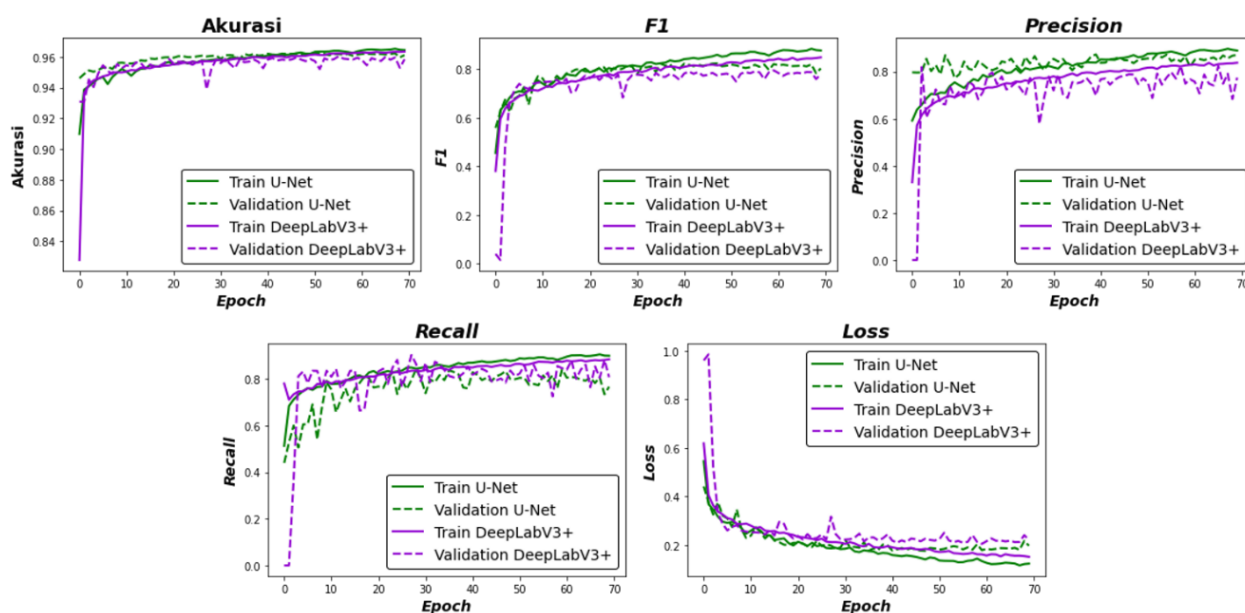
komputasi dengan bit komputer yang bernilai  $2^n$ . Proses komputasi akan menjadi lebih cepat jika data komputasi bernilai  $2^n$ .

Data akan dibagi menjadi 80% pelatihan dan 20% validasi dengan konfigurasi *5-fold cross-validation*. Proses pembagian ini diperlukan untuk mencegah kasus *high variance* atau *overfitting*. *High variance* adalah kasus dimana fungsi polinomial *neural network* memiliki derajat terlalu tinggi sehingga mengabaikan konteks regresi yang sebenarnya. Model akan dilatih sebanyak 70 *epoch* menggunakan *checkpoint* untuk menyimpan model dengan nilai F1 terbaik.

Model akan dilatih pada fasilitas *cloud computing* menggunakan perangkat keras *graphic processing unit* NVIDIA Tesla V100 dengan *random access memory* 27,4 *gigabyte*. *Framework* yang digunakan adalah Tensorflow 2.3.0 dengan dukungan Keras 2.4.3. Tensorflow adalah salah satu *high level* sekaligus *low level* API untuk *machine learning*, sedangkan Keras adalah salah satu *high level* API untuk *machine learning* yang menjadi *backbone* bagi tensorflow.

#### 4. HASIL PELATIHAN DAN PEMBAHASAN

Hasil pelatihan dengan *dataset* eksternal telah dibahas pada penelitian sebelumnya [4], sedangkan hasil kurva pelatihan dengan *dataset* gabungan dapat dilihat pada Gambar 6. Garis putus-putus menunjukkan tren validasi, sedangkan garis tegas menunjukkan tren pelatihan. Konvergensi U-Net digambarkan dengan garis hijau, sedangkan konvergensi DeepLabV3+ digambarkan dengan garis ungu. Garis pelatihan dan garis validasi yang berdekatan mengindikasikan bahwa model tidak *overfitting*. Hasil akhir *checkpoint* dapat dilihat pada Tabel 1. U-Net memiliki nilai F1, akurasi, *precision*, dan *recall* masing-masing sebesar 85,92%, 96,36%, 91,08%, dan 84,70%, sedangkan DeepLabV3+ memiliki nilai F1, akurasi, *precision*, dan *recall* masing-masing sebesar 84,09%, 96,16%, 81,98%, dan 88,77%. Berdasarkan parameter F1 yang merupakan parameter pembandingan utama, U-Net berperforma lebih baik dari DeepLabV3+. Hasil dari akurasi kedua arsitektur bernilai di atas 90%. Hal ini terjadi karena nilai akurasi akan berorientasi pada jumlah kelas yang lebih besar, yaitu jumlah piksel *foreground* yang benar atau *true negative*. Nilai *precision* yang lebih tinggi dari *recall* mengindikasikan bahwa hasil prediksi model cenderung memiliki lebih sedikit *false positive*, dimana *false positive* adalah nilai prediksi retak yang tidak sesuai dengan label. U-Net memiliki hasil *precision* lebih tinggi dari *recall*, sedangkan DeepLabV3+ memiliki hasil *recall* lebih tinggi dari *precision*. Oleh karena itu, dapat disimpulkan bahwa U-Net lebih banyak membuat prediksi negatif yang tidak sesuai dengan label, sedangkan DeepLabV3+ lebih banyak membuat prediksi positif yang tidak sesuai dengan label.



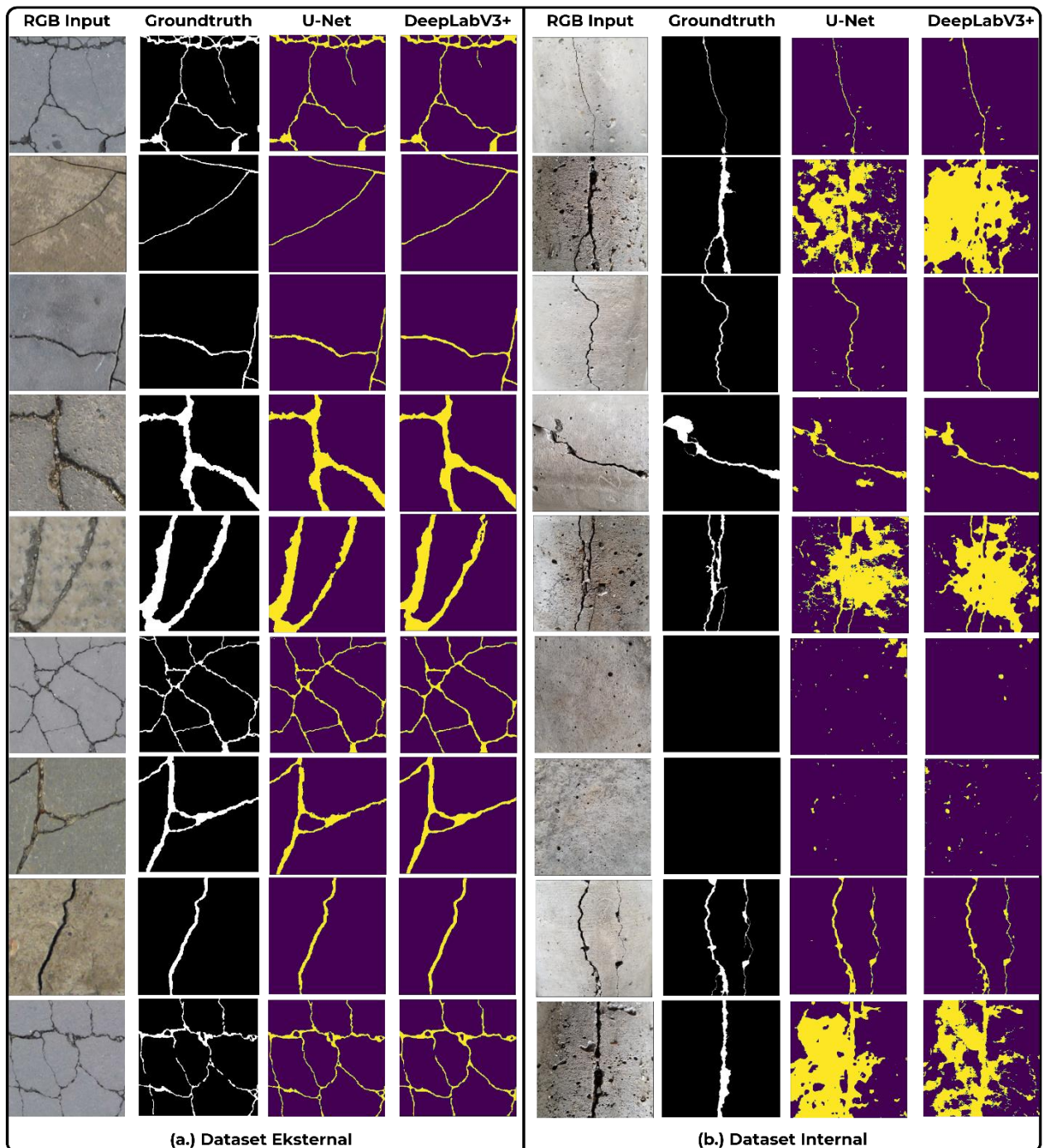
Gambar 6. Hasil Pelatihan dengan *Dataset* Gabungan

Berdasarkan Tabel 1, dapat dilihat bahwa model yang hanya dilatih menggunakan *dataset* eksternal hanya memiliki nilai F1 sebesar 75,71% dan 76,47% masing-masing untuk U-Net dan DeepLabV3+ saat di tes menggunakan *dataset* gabungan. Nilai ini berbeda jauh saat model dites dengan *dataset* eksternal, dimana U-Net memiliki nilai F1 sebesar 87,31% dan DeepLabV3+ memiliki nilai F1 sebesar 87,12%. Hal ini menunjukkan bahwa foto beton ber-void pada *dataset* gabungan memberikan dampak negatif pada performa model yang hanya dilatih dengan *dataset* eksternal. Oleh karena itu, diperlukan *feature engineering* pada *dataset* pelatihan, yaitu melatih model dengan *dataset* gabungan. Setelah model dilatih dengan *dataset* gabungan, nilai F1 mengalami peningkatan performa secara signifikan. Nilai F1 meningkat menjadi 85,92% dan 84,09% masing-masing untuk U-Net dan DeepLabV3+ saat di tes menggunakan *dataset* gabungan. Saat dievaluasi terhadap *dataset* eksternal pada model ini memiliki nilai F1 87,72% dan 85,54% masing-masing untuk U-Net dan DeepLabV3+, dimana hasil ini relatif sama dengan model yang hanya dilatih dengan *dataset* eksternal.

Tabel 1. Hasil Evaluasi Model

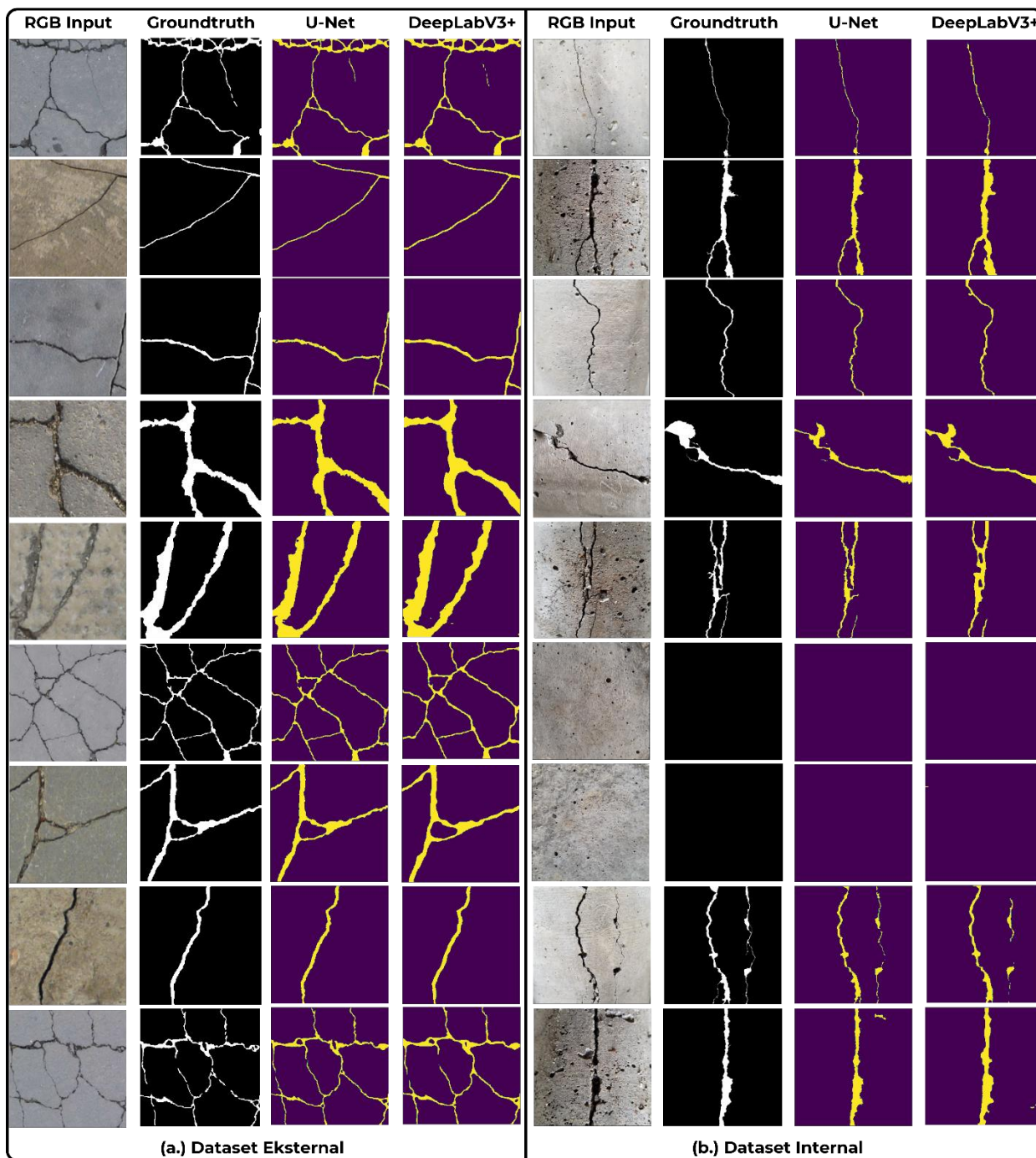
	Tes Eksternal	Tes Gabungan
<b>Pelatihan U-Net (Eksternal)</b>	Akurasi = 96,32% F1 = 87,31% <i>Precision</i> = 86,54% <i>Recall</i> = 90,05%	Akurasi = 93,34% F1 = 75,71% <i>Precision</i> = 69,97% <i>Recall</i> = 88,38%
<b>Pelatihan DeepLabV3+ (Eksternal)</b>	Akurasi = 96,41% F1 = 87,12% <i>Precision</i> = 84,79% <i>Recall</i> = 90,42%	Akurasi = 93,68% F1 = 76,47% <i>Precision</i> = 70,94% <i>Recall</i> = 87,64%
<b>Pelatihan U-Net (Gabungan)</b>	Akurasi = 96,48% F1 = 87,72% <i>Precision</i> = 89,80% <i>Recall</i> = 87,71%	Akurasi = 96,36% F1 = 85,92% <i>Precision</i> = 91,08% <i>Recall</i> = 84,70%
<b>Pelatihan DeepLabV3+ (Gabungan)</b>	Akurasi = 96,19% F1 = 85,54% <i>Precision</i> = 81,58% <i>Recall</i> = 90,84%	Akurasi = 96,16% F1 = 84,09% <i>Precision</i> = 81,98% <i>Recall</i> = 88,77%

Untuk evaluasi lebih lanjut, maka akan dilakukan pengujian model. Pengujian dilakukan menggunakan *dataset* eksternal dan *dataset* internal. Hasil pengujian pada model yang dilatih dengan *dataset* eksternal dapat dilihat pada Gambar 7. Baik U-Net atau DeepLabV3+ dapat mengsegmentasi retak dengan baik pada *dataset* eksternal. Namun, kedua model ini masih menganggap *void* sebagai retak pada *dataset* internal. Hal ini mungkin disebabkan karena model masih menganggap retak sebagai fungsi dari intensitas warna secara sederhana. Model belum memperhitungkan fitur kompleks yang seharusnya dijadikan filter atau *kernel* utama dalam *layer convolution*. Hasil yang kurang baik dari model ini berbanding lurus dengan nilai F1 (75,71% dan 76,47%) yang kurang baik saat dites dengan *dataset* gabungan. Hasil pengujian pada model yang dilatih dengan *dataset* gabungan dapat dilihat pada Gambar 8. Baik U-Net atau DeepLabV3+ dapat membedakan *void* dengan retak. Hasil ini sesuai dengan peningkatan nilai F1 yang telah dibahas di atas. Penambahan *dataset* internal juga tidak mempengaruhi kualitas segmentasi pada *dataset* eksternal. Hal ini tercermin dari nilai F1 yang relatif sama dengan model sebelumnya saat diuji dengan *dataset* eksternal. Semua foto yang digunakan untuk pengujian adalah foto validasi agar hasil tidak memiliki bias.



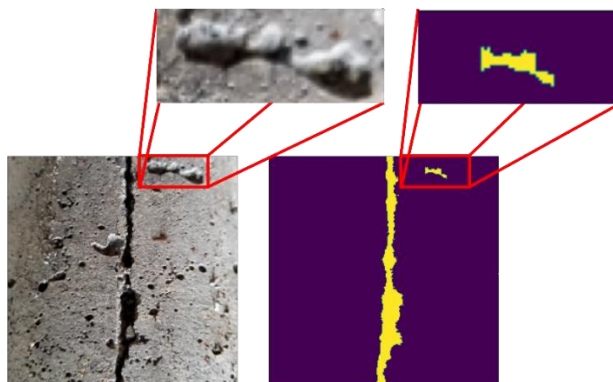
Gambar 7. Hasil Pengujian Model yang Dilatih dengan *Dataset Eksternal*





Gambar 8. Hasil Pengujian Model yang Dilatih dengan *Dataset* Gabungan

Terdapat sebuah keterbatasan pada model yang dilatih menggunakan *dataset* gabungan. Model terkadang menganggap *void* yang berbentuk memanjang seperti retak sebagai retak. Seperti yang terlihat pada Gambar 9, *void* yang berbentuk memanjang dan berliku seperti retak dianggap sebagai retak oleh model. Hal ini cukup umum terjadi pada pengujian *neural network* dimana seringkali terdapat bias yang susah dihindari. Namun, mengingat data yang digunakan sangat terbatas, dapat diambil kesimpulan bahwa *neural network* berperforma sangat baik dalam membedakan *void* dengan retak untuk keseluruhan *dataset*.



Gambar 9. Void Berbentuk seperti Retak Seringkali Terdeteksi sebagai Retak

## 5. SIMPULAN

Penelitian ini mengimplementasikan sistem deteksi retak permukaan beton menggunakan dua arsitektur *encoder-decoder* CNN, yaitu U-Net dan DeepLabV3+. Terdapat keterbatasan pada *dataset* eksternal yang dipakai penelitian sebelumnya, yaitu *void* permukaan seringkali terdeteksi sebagai retak. Oleh karena itu, *dataset* internal yang secara khusus membedakan *void* dengan retak disusun. *Dataset* internal digabungkan dengan *dataset* eksternal menjadi *dataset* gabungan, yang kemudian digunakan untuk melatih *neural network* dan dievaluasi pengaruhnya terhadap nilai F1 dan hasil prediksi. Berdasarkan hasil pelatihan dan evaluasi, dapat diambil kesimpulan sebagai berikut:

- 1) Peningkatan nilai F1 menunjukkan bahwa penambahan *dataset* internal berpengaruh signifikan pada performa model. Hasil pelatihan menunjukkan bahwa nilai F1 meningkat signifikan setelah penambahan *dataset* internal. Nilai F1 meningkat dari 75,71% dan 76,47% masing-masing untuk U-Net dan DeepLabV3+ menjadi 85,92% dan 84,09% masing-masing untuk U-Net dan DeepLabV3+ saat dievaluasi menggunakan *dataset* gabungan.
- 2) Berdasarkan gambar hasil pengujian, penambahan *dataset* pelatihan internal yang terdiri dari foto beton ber-*void* berhasil meningkatkan kemampuan model untuk membedakan retak dengan *void*, baik pada arsitektur U-Net atau arsitektur DeepLabV3+.
- 3) U-Net berperforma lebih baik dari DeepLabV3+. Hal ini dapat dilihat pada nilai F1 kedua arsitektur. Model U-Net yang dilatih baik menggunakan *dataset* eksternal ataupun *dataset* gabungan memiliki nilai F1 yang lebih tinggi dari model DeepLabV3+.

## DAFTAR RUJUKAN

- [1] O. Ronneberger, P. Fischer, dan T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," In International Conference on Medical image computing and computer-assisted intervention, 2015, hal. 234-241, doi: 10.1007/978-3-319-24574-4\_28.
- [2] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, dan H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," In Proceedings of the European conference on computer vision (ECCV), 2018, hal. 801-818.
- [3] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang, dan X. Yang, "Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, hal. 1090-1109, 2018, doi: 10.1111/mice.12412.
- [4] P.N. Hadinata, D. Simanta, L. Eddy, dan K. Nagai, "Crack Detection on Concrete Surfaces Using Deep Encoder-Decoder Convolutional Neural Network: A Comparison Study Between U-Net and DeepLabV3+," *Journal of the Civil Engineering Forum*, vol. 7, no. 3, 2021.
- [5] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, dan R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *The journal of machine learning research*, vol. 15, no. 1, hal. 1929-1958, 2014.

- [6] L. C. Chen, G. Papandreou, F. Schroff, dan H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," 2017, arXiv preprint arXiv:1706.05587.
- [7] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, hal. 1251-1258.
- [8] C. Goutte, dan E. Gaussier, "A Probabilistic Interpretation of Precision, Recall And F-Score, With Implication for Evaluation," In European conference on information retrieval, 2005, hal. 345-359, doi: 10.1007/978-3-540-31865-1\_25.
- [9] D. P. Kingma, dan J. Ba, "Adam: A Method for Stochastic Optimization," 2014, arXiv preprint arXiv:1412.6980.
- [10] F. Milletari, N. Navab, dan S. A. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," In 2016 fourth international conference on 3D vision (3DV), 2016, hal. 565-571, doi: 10.1109/3DV.2016.79.