

Segmentasi Semantik Dengan Objek Api Menggunakan U-Net

1st Delfia Nur Anrianti Putri

Dept. Ilmu Komputer dan Elektronika
Universitas Gadjah Mada
Yogyakarta, Indonesia
delfia.nur2004@mail.ugm.ac.id

2nd Abid Nujaiaba

Dept. Ilmu Komputer dan Elektronika
Universitas Gadjah Mada
Yogyakarta, Indonesia
abid.nujaiaba@mail.ugm.ac.id

3rd Jovincia Claudia Amarissa

Dept. Ilmu Komputer dan Elektronika
Universitas Gadjah Mada
Yogyakarta, Indonesia
jovincia.claudia.amarissa@mail.ugm.ac.id

4th Andyan Yogawardhana

Dept. Ilmu Komputer dan Elektronika
Universitas Gadjah Mada
Yogyakarta, Indonesia
andyan.yogawardhana@mail.ugm.ac.id

5th Muhammad Mahdi

Dept. Ilmu Komputer dan Elektronika
Universitas Gadjah Mada
Yogyakarta, Indonesia
muhammad.mahdi@mail.ugm.ac.id

6th Ahnaf Bagus Aryandika

Dept. Ilmu Komputer dan Elektronika
Universitas Gadjah Mada
Yogyakarta, Indonesia
ahnafaryandika21@gmail.com

Abstrak—Pentingnya suatu sistem untuk melakukan pendeteksian api sebagai bentuk upaya preventif terhadap kasus kebakaran menjadi salah satu hal yang perlu diperhatikan. Sistem yang dapat membantu menurunkan potensi terjadinya kecelakaan yang lebih parah dapat dikembangkan dengan memberikan peringatan dini jika terindikasi adanya kemungkinan terjadinya kebakaran berdasarkan indikator-indikator tertentu. Pada paper ini, dilakukan pengembangan suatu sistem yang memenuhi kriteria tersebut berupa model yang dapat melakukan pendeteksian objek api berdasarkan input citra yang diberikan. Model tersebut mengimplementasikan kemampuan segmentasi objek yang dimiliki oleh arsitektur U-Net berbasis *convolutional neural network* (CNN). Model diharapkan dapat meningkatkan performa dan kecepatan sistem dalam pendeteksian objek api. Dengan menggunakan arsitektur tersebut, model berhasil membangun sebuah model yang memiliki tingkat akurasi sebesar 76%. Performa model yang relatif tinggi tersebut diharapkan dapat memenuhi tujuan dari pengembangan paper ini, yaitu suatu sistem yang dapat melakukan pendeteksian api dengan cepat dan akurat.

Kata Kunci—penglihatan komputer, analisis citra, deteksi api, segmentasi, konvolusi, *neural network*, U-Net

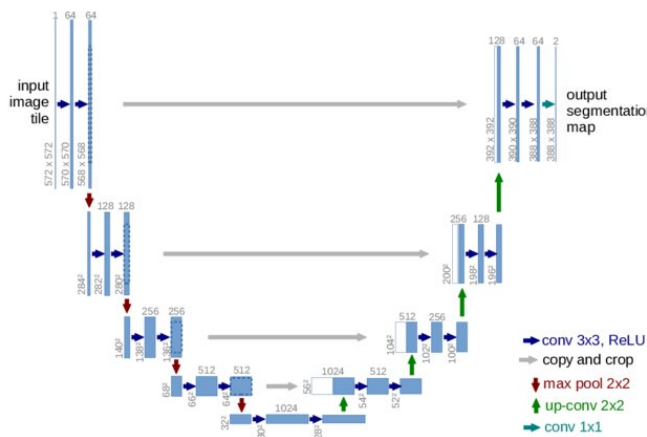
I. LATAR BELAKANG

Maraknya kasus kebakaran di berbagai tempat dari waktu ke waktu menjadi salah satu fenomena yang perlu diperhatikan. Sebagian besar dari kasus kecelakaan yang telah terjadi menimbulkan kerugian dalam jumlah yang tidak sedikit, baik itu kerugian finansial, kerusakan properti, atau dampak negatif lainnya. Bahkan, banyak pula korban jiwa yang berjatuh akibat dari kejadian ini. Berdasarkan data dari International Association of Fire and Rescue Services, sepanjang tahun 2020, sebanyak 20,6 ribu jiwa tercatat sebagai korban yang meninggal akibat kasus kebakaran di 48 negara, sedangkan 69.5 ribu korban luka-luka.[1] Maka dari itu, untuk mengurangi risiko terjadinya kasus kebakaran

dalam jumlah yang lebih besar, salah satu upaya yang dapat dilakukan adalah dengan mengembangkan sebuah model yang dapat melakukan pendeteksian api. Model ini diharapkan dapat mengenali area munculnya keberadaan api serta besarnya api yang berkobar di suatu tempat dengan cepat, akurat, dan dapat dengan mudah dipahami oleh pihak terkait agar dapat dilakukan tindakan pencegahan dan penanganan selanjutnya.

Salah satu algoritma yang telah terbukti efektif dalam pendeteksian objek pada bidang penglihatan komputer adalah U-Net. Algoritma U-Net dirancang khusus untuk tugas segmentasi, yang dapat memisahkan objek dari latar belakang dalam gambar. Dalam konteks pendeteksian api, U-Net dapat digunakan untuk mengenali area-area yang terkena api dan memisahkannya dari latar belakang. U-Net pada awalnya dikembangkan untuk segmentasi gambar biomedis, tetapi sejak saat itu telah diterapkan pada berbagai macam tugas lainnya, seperti segmentasi pemandangan, pendeteksian objek, dan analisis gambar medis[2].

Selain itu, model yang dikembangkan juga memanfaatkan teknik *upsampling* dan operasi *concatentation* yang dimiliki algoritma U-Net dalam proses segmentasi objeknya. Secara umum, teknik *upsampling* melakukan peningkatan resolusi citra yang bertujuan untuk meningkatkan kualitas dan mengembalikan detail citra yang hilang setelah melalui tahap-tahap sebelumnya, seperti *max pooling* untuk meningkatkan kuantitas fitur. Teknik ini memanfaatkan *layer* konvolusi dalam melakukan augmentasi pada nilai piksel tambahan citra. Proses *upsampling* dilakukan secara bertahap dalam beberapa langkah yang pada setiap tahapnya juga dilakukan operasi *concatentation*. Operasi ini membantu proses peningkatan kualitas dan resolusi citra agar memperoleh hasil yang akurat dan sesuai dengan citra asli sebelum dilakukannya *max pooling*.



Gambar 1. Skema arsitektur U-Net.

Pemanfaatan algoritma U-Net dalam proses segmentasi untuk mengatasi permasalahan pendeteksian api ini diharapkan dapat menghasilkan suatu solusi yang akurat dan adaptif terhadap berbagai kasus yang diamati. Model yang dihasilkan juga diharapkan memiliki tingkat akurasi yang tinggi sehingga dapat diandalkan untuk melakukan analisis yang tepat di berbagai kasus. Dengan dikembangkan model tersebut, harapannya dapat dilakukan upaya-upaya pencegahan maupun penanganan lanjutan berdasarkan deteksi api yang dihasilkan untuk mengurangi risiko-risiko kecelakaan lain yang lebih besar.

II. PENELITIAN TERKAIT

Terdapat beberapa penelitian terdahulu yang telah mengaplikasikan pembelajaran mesin mendalam atau *deep learning* pada pendeteksian objek api dalam sebuah citra. Penelitian yang dilakukan oleh Hossain dan Zhang (2021) [3] menunjukkan bahwa teknik *upsampling* RDSTC dan CETC memiliki performa yang baik dengan akurasi masing-masing sebesar 88,15% dan 88,32%, sedangkan metode *bilinear interpolation* dan *transposed convolution* memiliki akurasi yang sedikit lebih rendah. Penelitian ini mengusulkan tiga jaringan konvolusi, yaitu U-Net, U2-Net, dan EfficientSeg, dalam mendeteksi kebakaran hutan dan mengidentifikasi area yang terkena api. Metode yang diusulkan dievaluasi menggunakan dataset FLAME. Dalam eksperimen, operasi *upsampling* pada lapisan decoding U-Net diganti dengan *bilinear upsampling*, *transposed convolution* asli, RDSTC, dan CETC.

Pada eksperimen pendeteksian kebakaran hutan yang dilakukan oleh Zhang et al. (2021)^[X], arsitektur ATT Squeeze U-Net digunakan dengan menyisipkan unit Attention Gate (AG) dalam koneksi skip pada struktur U-shape. Dalam perbandingan waktu operasi enam model, U-Net menunjukkan hasil yang kompetitif dibandingkan dengan model lainnya. Sementara, pada tingkat akurasi, ATT Squeeze U-Net menunjukkan performa akurasi yang lebih baik dibanding lainnya [4].

Dalam penelitian Saima Majid et al. (2022), dilakukan perbandingan performa dan efektivitas dari *attentive-based*

convolutional neural network model dengan beberapa metode lain untuk deteksi kebakaran. Penelitian ini menggunakan metode Grad-CAM untuk visualisasi dan lokalisasi kebakaran dalam gambar. Model EfficientNetB0 yang diusulkan memiliki jumlah parameter yang lebih sedikit dibandingkan solusi lainnya dengan performa akurasi secara keseluruhan lebih dari 95% dan recall yang sangat tinggi sebesar 97,61% [5].

Penelitian deteksi dan segmentasi objek api pada kebakaran hutan juga dilakukan oleh Mseddi et al. (2021) menggunakan arsitektur YOLOv5 and U-NET. Model U-Net, dalam melakukan segmentasi api, berhasil mencapai performa yang sangat baik yaitu koefisien Dice 92% dan akurasi 99,6%. YOLO v3 yang diterapkan untuk mendeteksi citra api memberikan akurasi sebesar 96,8%. Dapat dikatakan bahwa keunggulan U-Net terletak pada kemampuannya, tidak hanya untuk mengkonfirmasi keberadaan kebakaran hutan, tetapi juga untuk mendeteksi bentuk yang tepat dari api. Dengan menggabungkan kedua arsitektur tersebut, penelitian berhasil mendeteksi & segmentasi kebakaran hutan dengan akurat[6].

Pada segmentasi objek api pada citra hutan menggunakan *Deep Convolutional Neural Network* oleh Ghali et al. (2021)^[X], U2-Net, EfficientSeg, U-Net mampu mengsegmentasi dengan efisien dan mendeteksi bentuk api dengan baik. Dalam pengujian dengan dataset CorsicanFire, EfficientSeg, U-Net, dan U2-Net menunjukkan performa yang baik dengan skor F1 masing-masing sebesar 0,95, 0,94, dan 0,92, serta akurasi masing-masing sebesar 0,96, 0,98, dan 0,97 dalam mengsegmentasi api hutan dan mendeteksi area kebakaran, terutama area kecil. Berdasarkan skor F1, EfficientSeg adalah metode yang paling baik dalam mengsegmentasi kebakaran hutan di lingkungan luar yang kompleks[7].

III. METODOLOGI

Bagian ini diawali dengan penjelasan tentang dataset yang digunakan dan tahapan-tahapan pra pemrosesan data yang dilakukan sebelum melakukan pelatihan pada model yang dirancang. Pada bagian ini juga, akan dijabarkan rancangan arsitektur yang diusulkan untuk melakukan deteksi objek api pada citra. Permasalahan ini adalah permasalahan identifikasi objek api dimana keluaran yang dihasilkan adalah apakah citra tersebut memuat objek api.

A. Dataset

Dataset yang digunakan pada eksperimen ini bersumber dari data yang disediakan oleh Bilkent University dalam konteks penelitian mengenai segmentasi objek api [8]. Sumber data ini dikembangkan secara khusus oleh tim peneliti dan mungkin telah melibatkan pengumpulan data dari berbagai sumber, seperti citra kebakaran hutan, rekaman visual, atau sumber data lain yang telah dilabeli dengan baik untuk segmentasi objek api.

Citra-citra dalam dataset ini mencakup berbagai situasi kebakaran hutan dengan tingkat kompleksitas yang bervariasi, seperti ukuran api yang berbeda, tingkat kecerahan yang beragam, dan latar belakang yang berbeda. Dataset ini

dikembangkan secara khusus untuk mendukung penelitian dalam bidang segmentasi semantik objek api menggunakan model U-Net. Dengan menggunakan dataset ini, penulis dapat melakukan evaluasi dan pengujian model mereka untuk segmentasi objek api dengan akurasi yang lebih baik.



Gambar 2. Sampel dari dataset yang digunakan.

Secara keseluruhan, data yang digunakan dalam eksperimen ini terdiri atas 2684 citra yang memuat objek api. Kemudian, digunakan pula 2648 citra yang merupakan hasil dari *masking* 2648 citra sebelumnya sebagai label untuk merincikan posisi objek api pada citra. Sebanyak 80% dari data ini digunakan untuk pelatihan dan validasi dan 20% untuk pengujian.

Terdapat pra-pemrosesan yang dilakukan pada citra yang merupakan hasil *masking*. Pertama, citra diubah menjadi skala abu-abu *grayscale* untuk mengurangi dimensi dan fokus pada intensitas cahaya. Kemudian, dilakukan *resizing* pada citra untuk menyesuaikan ukuran dengan kebutuhan analisis atau pemrosesan selanjutnya. Terakhir, dimensi pada citra disesuaikan dengan ditambahkan agar citra memiliki dimensi yang sesuai dengan format yang diharapkan oleh model. Langkah-langkah ini membantu mempersiapkan citra agar siap untuk diproses lebih lanjut.

Selanjutnya, sejumlah 698 citra yang memuat objek api dan 832 citra yang tidak memuat objek api diambil dari data yang telah tersedia sebelumnya dan digunakan untuk melatih model CNN. Dalam melatih model CNN, dilakukan proses augmentasi pada dataset ini. Proses augmentasi ini dilakukan dengan menambahkan variasi data menggunakan *ImageDataGenerator* berdasarkan beberapa jenis transformasi dan parameter yang dapat dilihat pada tabel 2.

Tabel 2. Pengaturan parameter augmentasi.

Parameter Augmentasi	Value
Rotation Range	30
Width Shift Range	0.20
Height Shift Range	0.20
Shear Range	0.15
Zoom Range	0.15
Fill Mode	Nearest

B. CNN

Pada eksperimen ini, untuk melakukan klasifikasi pada citra api/non api, digunakan CNN atau Convolutional Neural Network. Komponen pertama pada CNN adalah layer konvolusi. Proses konvolusi sendiri adalah proses esensial dalam ekstraksi fitur pada data citra [9]. Output dari proses ini adalah *feature map*. Proses ini memanfaatkan kernel atau filter yang diaplikasikan ke seluruh bagian dari gambar dengan perhitungan secara umum, seperti yang ditunjukkan pada persamaan (1),

$$Y[m, n] = \sum_{i=1}^K \sum_{j=1}^K K[i, j] X[m-1+i, n-1+j] \quad (1)$$

dimana X adalah citra input, K adalah kernel, k adalah ukuran kernel K , dan Y adalah citra output.

Output final dari layer konvolusi didapatkan dengan menambahkan nilai bias dan mengaplikasikan fungsi aktivasi pada *feature map* yang dihasilkan seperti yang tercantum pada persamaan (2)

$$\hat{Y} = f(K * X + B) = f(Y + B), \quad (2)$$

$$f(x) = \max(0, x) \quad (3)$$

dimana \hat{Y} adalah output final dari neuron, B adalah matriks bias, dan f adalah fungsi aktivasi. Fungsi aktivasi yang umum digunakan adalah ReLU dimana output dari fungsi ini adalah seperti yang ditunjukkan dalam persamaan (3).

Layer berikutnya adalah pooling layer yang berfungsi untuk mereduksi ukuran spasial dari *feature map* yang diekstrak dari layer konvolusi. Metode umum yang digunakan dalam melakukan pooling adalah *max pooling* dan *average pooling*. Pada *max pooling*, nilai yang disimpan adalah nilai maksimal dari area reseptif dari kernel pooling, sedangkan pada *average pooling*, nilai yang disimpan adalah rata-rata dari seluruh nilai yang terdapat pada area tersebut.

Pada bagian terakhir, layer yang digunakan adalah *fully connected layer*. Layer ini terdiri atas neuron-neuron yang tersusun dalam layer dan saling terkoneksi dengan seluruh neuron di layer berikutnya. Output dari layer ini adalah hasil pengaplikasian fungsi aplikasi pada *propagation function* u_i^l untuk layer l melalui persamaan (4),

$$u_i^l = \sum_{j=1}^m W_{ij}^l o_j^{l-1} + b_i^l \quad (4)$$

dimana o^{l-1} adalah output dari layer sebelumnya, W^l adalah bobot pada neuron di layer l , dan b^l adalah bias untuk layer l [10].

Selain layer-layer yang disebutkan, penggunaan jenis layer tambahan seperti *batch normalization layer* dapat digunakan sebagai langkah optimasi untuk menstabilisasi dan mempercepat proses pelatihan. Layer ini mengaplikasikan normalisasi dengan cara mengurangi nilai input dengan *mean*

dan membagi dengan standar deviasi dari seluruh unit input pada batch tersebut. Proses *backpropagation* pada layer ini akan mempelajari parameter yang digunakan untuk melakukan penskalaan dan pergeseran (*shifting*) hasil normalisasi.

C. U-Net

Arsitektur U-Net terdiri dari dua bagian utama: *contracting path* dan *expansive path*. *Contracting path* bertanggung jawab untuk mengekstraksi fitur dari gambar input, sedangkan *expansive path* bertanggung jawab untuk meningkatkan fitur dan menghasilkan *mask* segmentasi akhir. *Contracting path* terdiri dari serangkaian lapisan konvolusi, diikuti oleh lapisan *max-pooling*. Lapisan *convolutional* mengekstrak fitur dari gambar input, sedangkan lapisan *max-pooling* mengurangi ukuran peta fitur. Hal ini memungkinkan jaringan untuk belajar mengidentifikasi fitur-fitur lokal yang kecil pada gambar input.

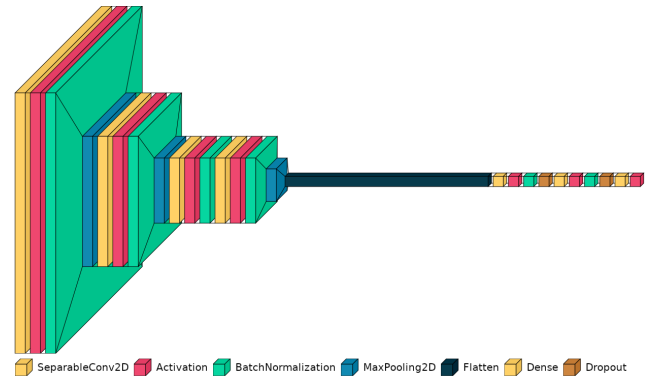
Dalam arsitektur U-Net, upsampling layer and concatenation memainkan peran penting dalam memungkinkan jaringan melakukan tugas segmentasi semantik. Pada Upscaling layer bertanggung jawab untuk meningkatkan resolusi spasial peta fitur. Hal ini bertujuan untuk memulihkan detail yang lebih halus yang hilang selama operasi downsampling atau pooling di jalur kontrak U-Net. Teknik yang paling umum digunakan untuk upsampling adalah konvolusi yang ditransposisikan (juga dikenal sebagai dekonvolusi atau konvolusi bertahap fraksional). Teknik ini melakukan operasi kebalikan dari konvolusi, memperluas peta fitur. Konvolusi yang ditransposisikan melibatkan penggunaan parameter yang dapat dipelajari (filter konvolusi) untuk melakukan upsampling pada peta fitur dan juga mempelajari pola upsampling dari data [11].

Upsampling layer membantu memulihkan informasi spasial yang hilang secara bertahap di jalur kontraksi dan memungkinkan jaringan untuk menghasilkan prediksi yang padat. Sedangkan Concatenation adalah operasi penting dalam arsitektur U-Net yang membantu mempertahankan informasi tingkat rendah dan tingkat tinggi selama proses upsampling. Pada setiap langkah upsampling, peta fitur dari jalur kontraksi digabungkan dengan peta fitur yang sesuai dari jalur upsampling. Concatenation ini memungkinkan jaringan untuk memanfaatkan informasi kontekstual yang kaya yang dipelajari di jalur contracting sambil mempertahankan detail spasial yang ditangkap di jalur upsampling.

Dengan menggabungkan peta fitur dari berbagai tingkat jaringan, arsitektur U-Net dapat secara efektif menangkap konteks lokal dan global, sehingga membantu dalam segmentasi yang akurat. Kombinasi upsampling layer dan concatenation memungkinkan arsitektur U-Net melakukan segmentasi semantik yang efisien dan tepat. Upsampling layer memulihkan informasi spasial yang hilang, sementara concatenation memfasilitasi integrasi fitur multi-skala untuk akurasi segmentasi yang lebih baik [12].

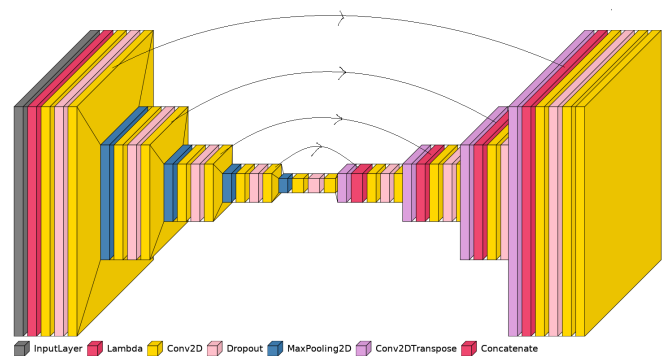
D. Arsitektur yang Diusulkan

Terdapat dua arsitektur yang digunakan untuk melakukan segmentasi api pada eksperimen ini. Arsitektur pertama adalah CNN yang digunakan untuk melakukan klasifikasi gambar api dan non api. Arsitektur ini terdiri atas tiga blok konvolusi yang terdiri atas *convolution layer* dengan unit masing-masing 16, 32, dan 64-64 yang diikuti dengan *batch normalization layer* dan *pooling layer* pada akhir blok. Ilustrasi dari arsitektur ini terdapat pada gambar 2.



Gambar 3. Arsitektur CNN yang digunakan.

Selanjutnya, untuk melakukan segmentasi pada gambar api, digunakan Arsitektur U-Net. Arsitektur dari model U-Net ini utamanya terdiri atas lima blok konvolusi untuk mengekstrak fitur dan diikuti oleh blok *upsampling* yang digunakan untuk mengembalikan informasi spasial yang hilang selama proses penurunan resolusi. Blok konvolusi terdiri atas *convolution layer* dengan unit masing-masing 16-16, 32-32, 64-64, 128-128, dan 256-256 dengan *dropout layer* di antara layer konvolusi, dimana setiap blok diakhiri oleh *max pooling layer* dengan ukuran 2x2 dan terkoneksi dengan blok *upsampling* melalui *concatenate layer*. Ilustrasi dari arsitektur ini terdapat pada gambar 3.



Gambar 4. Arsitektur U-Net yang digunakan.

Output dari proses klasifikasi yang dilakukan dengan memanfaatkan model CNN adalah gambar-gambar yang memiliki kelas api dan non-api. Pada proses pelatihan, hanya digunakan gambar dengan kelas api untuk melatih model segmentasi agar model segmentasi dapat mempelajari fitur non api dengan lebih baik. Untuk proses inferensi, model klasifikasi akan digunakan untuk memutuskan apakah gambar

perlu diteruskan ke dalam model segmentasi atau tidak. Jika gambar tidak masuk ke dalam kelas non-api, gambar diteruskan ke dalam model segmentasi. Apabila sebaliknya, gambar tidak diteruskan ke dalam model segmentasi dan diberikan output berupa gambar hitam yang mengindikasikan tidak ada api yang terdeteksi.

IV. HASIL DAN PEMBAHASAN

Model klasifikasi api/non api dilatih menggunakan parameter batch size=64 selama 50 epoch. Pada proses training, digunakan optimizer RMSProp dengan learning rate mula-mula sebesar 0.1. Adapun loss function yang digunakan adalah binary cross-entropy dengan formula yang ditunjukkan pada persamaan (5) dan metrik evaluasi yang digunakan adalah akurasi yang menggunakan perhitungan yang ditunjukkan pada persamaan (6).

Kemudian, model segmentasi dilatih menggunakan parameter batch size=32 selama 70 epoch. Pada proses training, digunakan optimizer RMSProp dengan learning rate sebesar 0.001. Adapun loss function yang digunakan adalah binary cross-entropy dengan formula yang ditunjukkan pada persamaan (5) dan metrik evaluasi yang digunakan adalah akurasi dan binary IoU dengan perhitungan yang ditunjukkan pada persamaan (6) dan (7) secara berurutan.

$$\text{Loss} = -\frac{1}{N} \sum_{i=1}^n y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i) \quad (5)$$

dimana y^i dan \hat{y}^i adalah target dan nilai prediksi dari model untuk sampel ke- i .

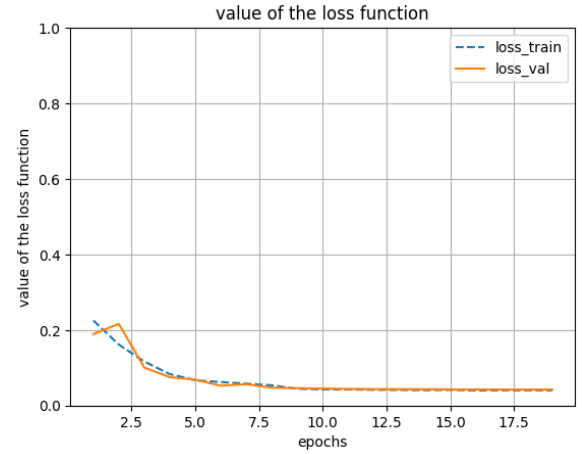
$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

dimana TP adalah *True Positive*, TN adalah *True Negative*, FP adalah *False Positive*, dan FN adalah *False Negative*.

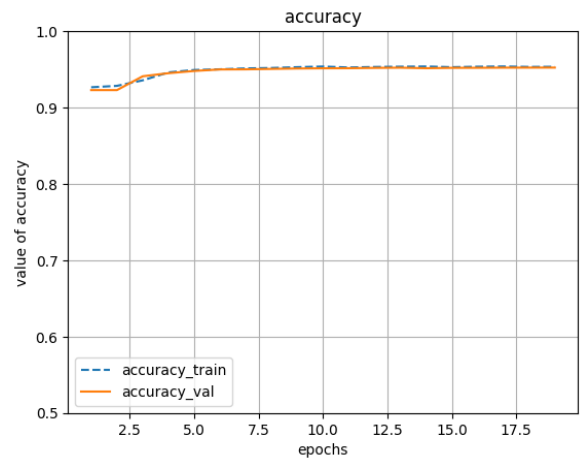
$$\text{IoU} = \frac{TP}{(TP + FP + FN)} \quad (7)$$

dimana TP adalah *True Positive*, FP adalah *False Positive*, dan FN adalah *False Negative*.

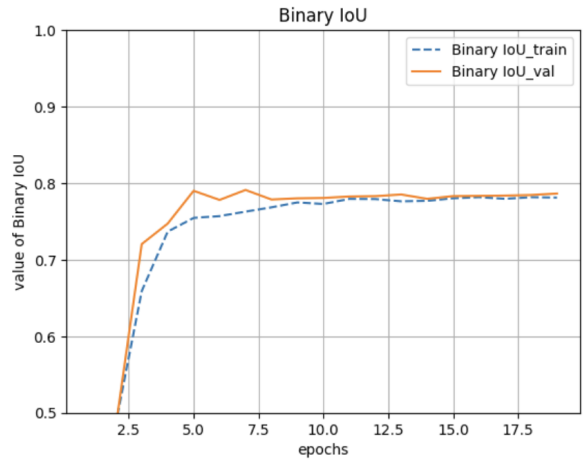
Plot dari akurasi dan loss model selama proses training terdapat pada gambar 5 dan gambar 6. Akurasi dari kedua model dirangkum pada tabel 4, dimana terlihat model menghasilkan akurasi pada data uji yang cukup tinggi baik untuk data latih maupun data uji.



Gambar 5. Plot loss dari pelatihan model U-Net

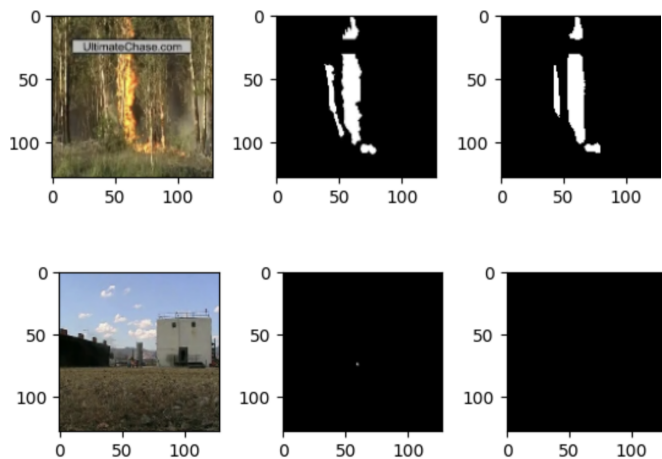


Gambar 6. Plot accuracy dari pelatihan model U-Net



Gambar 7. Plot accuracy dari pelatihan model U-Net

Hasil dari segmentasi dari dua sampel uji juga menunjukkan ketepatan segmentasi yang tinggi, dimana empat area utama dimana api terdeteksi berhasil teridentifikasi untuk gambar dengan api dan tidak mendeteksi apapun pada gambar yang tidak memiliki api.



Gambar 7. Hasil segmentasi data uji.

Tabel 4. Metrik evaluasi pada data uji.

Metrik	Value
Akurasi	0.9533
IoU	0.7871
Loss	0.0437

V. Kesimpulan

Pengembangan model yang telah dilakukan berhasil memenuhi tujuan utamanya, yaitu mendeteksi kemunculan api pada suatu citra. Model yang berhasil dikembangkan memiliki tingkat akurasi yang cukup baik, yaitu sejumlah 95%.

Nilai akurasi dan IoU sudah cukup tinggi untuk memprediksi lokasi api pada citra. Arsitektur U-Net yang dikombinasikan dengan CNN dinilai cukup cocok untuk menangani kasus pada penelitian ini karena berhasil memperoleh nilai yang cukup optimal pada metrik evaluasi yang telah ditentukan.

Namun, masih terdapat beberapa kekurangan yang dimiliki oleh model ini. Ditemukan beberapa citra api yang tidak terlihat seperti api dan model gagal melakukan analisis dengan baik sehingga segmentasi masih belum sempurna. Selain itu, masih ditemukan juga objek non-api yang terkategori sebagai kelas api dan disegmentasikan.

Berdasarkan hasil evaluasi terhadap model ini, masih diperlukan pengembangan lebih lanjut untuk meningkatkan kemampuan klasifikasi. Perlu dilakukan pelatihan pada dataset yang lebih besar dan lebih beragam agar hasil keakuratan yang diperoleh lebih maksimal, terutama pada objek non-api yang menyerupai api dan objek api yang tidak menyerupai api, seperti tertutup asap, berjarak jauh, dan lain sebagainya.

Referensi

- [1] D. N. Brushlinsky, "World Fire Statistics Estadísticas Mundiales de Bomberos Die Feuerwehrstatistik der Welt," 2022.
- [2] O. Ronneberger, P. Fischer, and T. Brox, "U-Net:

Convolutional Networks for Biomedical Image Segmentation." arXiv, May 18, 2015. doi: 10.48550/arXiv.1505.04597.

- [3] F. M. A. Hossain and Y. Zhang, "Development of New Efficient Transposed Convolution Techniques for Flame Segmentation from UAV-captured Images," in *2021 3rd International Conference on Industrial Artificial Intelligence (IAI)*, Nov. 2021, pp. 1–6. doi: 10.1109/IAI53119.2021.9619442.
- [4] J. Zhang, H. Zhu, P. Wang, and X. Ling, "ATT Squeeze U-Net: A Lightweight Network for Forest Fire Detection and Recognition," *IEEE Access*, vol. 9, pp. 10858–10870, 2021, doi: 10.1109/ACCESS.2021.3050628.
- [5] S. Majid, F. Alenezi, S. Masood, M. Ahmad, E. S. Gündüz, and K. Polat, "Attention based CNN model for fire detection and localization in real-world images," *Expert Syst. Appl.*, vol. 189, p. 116114, Mar. 2022, doi: 10.1016/j.eswa.2021.116114.
- [6] W. S. Mseddi, R. Ghali, M. Jmal, and R. Attia, "Fire Detection and Segmentation using YOLOv5 and U-NET," in *2021 29th European Signal Processing Conference (EUSIPCO)*, Aug. 2021, pp. 741–745. doi: 10.23919/EUSIPCO54536.2021.9616026.
- [7] R. Ghali, M. A. Akhloufi, M. Jmal, W. S. Mseddi, and R. Attia, "Forest Fires Segmentation using Deep Convolutional Neural Networks," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2021, pp. 2109–2114. doi: 10.1109/SMC52423.2021.9658905.
- [8] Wahyono, A. Dharmawan, A. Harjoko, Chrystian, and F. D. Adhinata, "Region-based Annotation Data of Fire Images for Intelligent Surveillance System." Zenodo, Jan. 23, 2022. doi: 10.5281/zenodo.5893854.
- [9] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.
- [10] J. Raitoharju, "Chapter 3 - Convolutional neural networks," in *Deep Learning for Robot Perception and Cognition*, A. Iosifidis and A. Tefas, Eds., Academic Press, 2022, pp. 35–69. doi: 10.1016/B978-0-32-385787-1.00008-7.
- [11] J. S. Suri *et al.*, "UNet Deep Learning Architecture for Segmentation of Vascular and Non-Vascular Images: A Microscopic Look at UNet Components Buffered With Pruning, Explainable Artificial Intelligence, and Bias," *IEEE Access*, vol. 11, pp. 595–645, 2023, doi: 10.1109/ACCESS.2022.3232561.
- [12] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications," *IEEE Access*, vol. 9, pp. 82031–82057, 2021, doi: 10.1109/ACCESS.2021.3086020.