



DETEKSI BANJIR AREA PERKOTAAN BERBASIS CITRA DIGITAL DAN CONVOLUTIONAL NEURAL NETWORK (VGG19)

Habibullah Akbar^a, Diah Aryani^b, Muhamad Bahrul Ulum^c

^a Fakultas Ilmu Komputer / Magister Ilmu Komputer, habibullah.akbar@esaunggul.ac.id, Universitas Esa Unggul

^b Fakultas Ilmu Komputer / Teknik Informatika, diah.aryani@esaunggul.ac.id, Universitas Esa Unggul

^c Fakultas Ilmu Komputer / Teknik Informatika, m.bahrul_ulum@esaunggul.ac.id, Universitas Esa Unggul

ABSTRACT

Geographically and demographically, Indonesia has natural conditions that have the potential for floods disaster. There are at least 16,771 islands and 65,017 rivers that fill the archipelago. Unfortunately, the ever-increasing urban population accompanied by a lack of awareness and preparation for protecting the environment has resulted in a higher risk of flooding in urban areas. This study utilizes digital imagery to detect flood conditions in urban areas. In terms of access, digital images are available in urban CCTV monitoring systems as well as office areas, housing, and from people who have smartphones. The detection method used in this study is the VGG19 model which consists of 16 convolution layers and 3 standard classification layers. All convolution layers are divided into 5 blocks followed by a MaxPooling layer for each block to reduce the resolution of the input image. In the last layer, SoftMax layer is used to estimate the probability between flood labels and normal conditions. There are 4 parameters that were optimized during the VGG19 model training process, namely Batch Size, Learning Rate, Dropout and Epoch (training repetition). To test the proposed model, public datasets are used, namely the Roadway Flooding Image Dataset and Road Vehicle Images Dataset. The best flood detection results (or normal conditions) achieve the accuracy of 98.78%. As for the other three performance metrics, namely precision, recall and F1-score, they reach 99%. These results are generated by the VGG19 model with a Batch Size parameter of 20, a Learning Rate of 1e-5 (0.00001), 50% Dropout and 100 Epoch. The achievement values of the four metrics can be considered quite good, so that the VGG19 model has the opportunity to be developed for flood detection applications in order to monitor urban flood conditions.

Keywords: VGG19, convolutional neural networks, flood detection, urban areas.

ABSTRAK

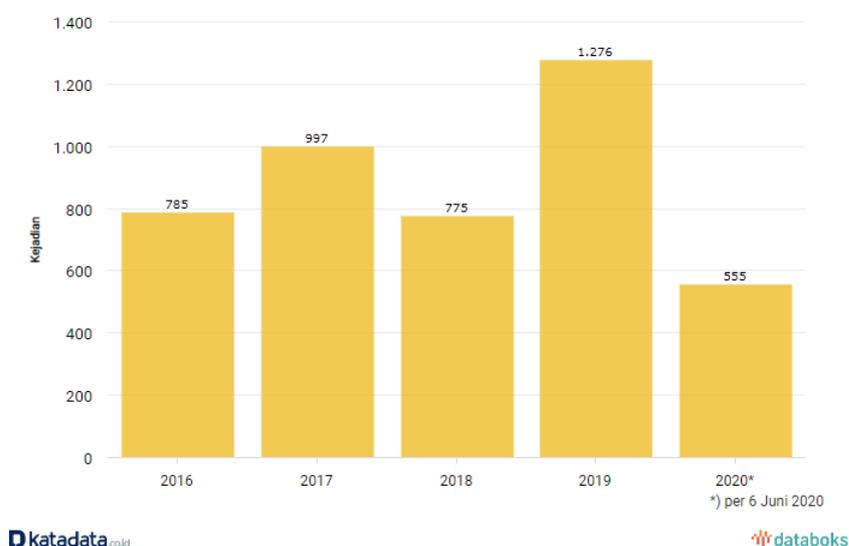
Secara geografis dan demografis, Indonesia memiliki keadaan alam yang berpotensi mengalami bencana banjir. Terdapat setidaknya 16,771 pulau dan 65,017 anak sungai yang mengisi sepanjang nusantara. Sayangnya, penambahan penduduk perkotaan yang terus meningkat disertai kurangnya kesadaran dan persiapan untuk memelihara lingkungan menyebabkan risiko banjir pada perkotaan semakin tinggi. Penelitian ini memanfaatkan citra digital untuk mendeteksi kondisi banjir pada daerah perkotaan. Secara akses, citra digital dapat tersedia di sistem monitoring CCTV perkotaan maupun area perkantoran, perumahan, maupun dari masyarakat yang memiliki smartphone. Metode deteksi yang digunakan adalah model VGG19 yang terdiri dari 16 lapisan konvolusi dan 3 lapisan klasifikasi standar. Seluruh lapisan konvolusi tersebut dibagi menjadi 5 blok yang diikuti lapisan MaxPooling pada setiap blok untuk mereduksi resolusi citra input. Pada lapisan terakhir, digunakan lapisan SoftMax yang melakukan estimasi probabilitas antara label banjir dan kondisi normal. Terdapat 4 parameter yang dioptimasi selama proses training model VGG19 yaitu Batch Size, Learning Rate, Dropout dan Epoch (pengulangan training). Untuk menguji model usulan, digunakan dataset publik yaitu Roadway Flooding Image Dataset dan Road Vehicle Images Dataset. Hasil deteksi banjir (atau kondisi normal) terbaik mencapai akurasi 98.78%. Sedangkan untuk ketiga metrik performa lain yaitu presisi, recall dan F1-score mencapai 99%. Hasil ini dihasilkan oleh model VGG19 dengan parameter Batch Size sebesar 20, Learning Rate sebesar 1e-5 (0.00001), 50% Dropout dan 100 Epoch. Nilai capaian dari keempat metrik tersebut dapat dianggap cukup

baik, sehingga model VGG19 berpeluang untuk dikembangkan pada aplikasi deteksi banjir untuk memonitoring kondisi perkotaan.

Kata Kunci: VGG19, jaringan saraf konvolusi, deteksi banjir, perkotaan.

1. PENDAHULUAN

Secara umum, Indonesia memiliki potensi geografis dan demografis yang kompleks. Terdapat sekitar 16,771 pulau [1] dan 65,017 anak sungai [2] yang mengisi sepanjang nusantara. Banyaknya aliran sungai, kondisi cuaca, dan curah hujan menyebabkan Indonesia memiliki risiko banjir yang cukup tinggi. Khususnya di daerah perkotaan yang padat demografi penduduknya dan pada saat musim hujan di penghujung tahun. Dampak negatif banjir tidak dapat dianggap remeh karena kerugian yang ditimbulkan tidak hanya sebatas masalah ekonomi seperti kerusakan sarana dan prasarana, namun juga dapat berupa korban yang tidak diinginkan. Berdasarkan statistik databoks (katadata media network), bencana banjir di Indonesia mencapai sekitar 4300 kejadian selama 2015-2020 [3] sebagaimana yang ditunjukkan pada gambar 1. Adapun pada tahun 2021, kejadian banjir meningkat menjadi 1288 [4].



Gambar 1. Jumlah Kejadian Bencana Banjir di Indonesia 2015-2020

Menurut [5], risiko banjir meningkat karena beberapa faktor seperti pembangunan yang kurang memiliki drainase yang baik, masyarakat yang kurang peduli dengan sampah lingkungan, kurangnya upaya untuk melakukan konservasi air, berkurangnya ketinggian permukaan tanah dan curah hujan yang deras. Untuk menanggulangi risiko tersebut, diperlukan kerjasama antara pemerintah, sektor swasta dan masyarakat perkotaan yang menggabungkan beberapa disiplin seperti teknik, hukum, ekonomi, sosial, dan lingkungan [6]. Selain itu, perkembangan media sosial dapat dimanfaatkan untuk menyebarkan informasi banjir dan penggalangan dana sosial dan relawan serta membantu koordinasi distribusi bantuan bagi korban banjir. Tanpa partisipasi dan sinergi dari berbagai pihak tersebut, maka pengelolaan banjir akan sulit berkembang. Oleh karena itu diperlukan sistematisasi, komitmen, pembagian peran dan operasional bersama untuk menanggulangi bencana banjir [7].

Terdapat beberapa studi yang telah membahas bagaimana penanggulangan banjir di daerah perkotaan seperti Kota Tangerang [8], Kota Semarang [9], Kota Medan [10], dan Kota Jakarta [11]. Strategi yang diambil mencakup pembangunan sistem drainase (mengalirkan air ke laut), resapan air kedalam tanah, serta program peningkatan kapasitas adaptasi masyarakat terhadap banjir. Selain itu, peran masyarakat juga sangat penting sebagaimana program pencegahan banjir yang disosialisasikan oleh Pemerintah Kota Tangerang dalam bentuk himbuan kerja bakti untuk membersihkan lingkungan. Penjagaan lingkungan ini penting karena sistem drainase perkotaan tidak akan maksimal jika terdapat sumbatan-sumbatan sampah sehingga air tidak dapat dialirkan dengan baik.

Secara teknologi, beberapa penelitian-penelitian yang telah dilakukan sebelumnya mengembangkan metode deteksi banjir dengan perangkat keras seperti sensor ultrasonik [12] dan sensor *water flow* [13]. Namun pengembangan sistem seperti ini boleh dikatakan memerlukan biaya tambahan untuk perangkat keras sensor, prosesor serta instalasi dan pemeliharannya. Adapun, beberapa daerah perkotaan telah dilengkapi dengan CCTV sehingga akan sangat disayangkan jika pemanfaatannya hanya sebatas monitoring kemacetan dan lalu lintas. Oleh karena itu, pada penelitian ini kami akan mengembangkan sistem deteksi banjir dengan memanfaatkan citra digital sebagai data input. Selain itu kekurangan dari penelitian yang dilakukan oleh [13] yaitu masih menggunakan model *machine learning* tradisional seperti algoritma Naive Bayes dan K-NN. Jika kedua model tersebut digunakan pada citra digital maka sistem deteksi banjir akan membutuhkan proses tambahan untuk melakukan perbaikan kualitas citra, segmentasi dan pemilihan fitur banjir.

Adapun pada penelitian ini, kami berfokuskan untuk mengembangkan metode deteksi banjir dengan berdasarkan citra yang sebenarnya tersedia di sistem monitoring CCTV perkotaan maupun area perkantoran dan perumahan. Metode deteksi yang dikembangkan pada studi ini adalah model jaringan saraf konvolusional (*convolutional neural network*) VGG19 yang terdiri dari 16 lapisan konvolusi dan 3 lapisan klasifikasi standar. Seluruh lapisan konvolusi tersebut dibagi menjadi 5 blok yang diikuti lapisan MaxPooling pada setiap blok untuk mereduksi resolusi citra input. Pada lapisan terakhir, digunakan lapisan SoftMax yang melakukan estimasi probabilitas antara label banjir dan kondisi normal. Terdapat 4 parameter yang dioptimasi selama proses training model VGG19 yaitu Batch Size, Learning Rate, Dropout dan Epoch (pengulangan training). Metode ini dipilih karena VGG19 mampu melakukan ekstraksi fitur banjir secara otomatis yang tidak dapat dilakukan oleh model jaringan saraf tiruan standar/tradisional.

2. TINJAUAN PUSTAKA

2.1. Citra Digital

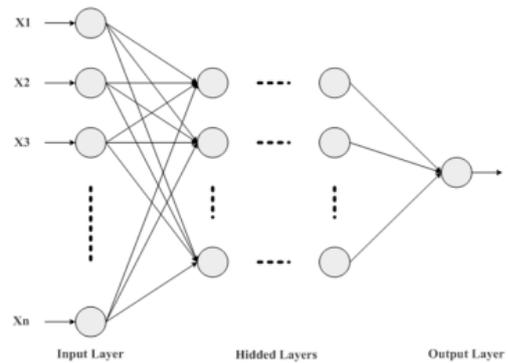
Secara digital, citra dapat didefinisikan sebagai fungsi $f(x,y)$, dimana x dan y adalah sumbu mendatar dan tegak pada koordinat bidang. Adapun fungsi f menunjukkan nilai intensitas atau tingkat keabuan dari citra pada koordinat x dan y . Pengolahan citra digital merupakan teknik atau algoritma yang memanfaatkan daya komputasi mesin untuk peningkatan kualitas citra, segmentasi, hingga menggantikan interpretasi manusia untuk menentukan konten didalam citra. Interpretasi oleh komputer seringkali disebut juga dengan klasifikasi (oleh mesin).

2.2. Artificial Neural Network (ANN)

ANN atau jaringan saraf tiruan merupakan sebuah model sistem klasifikasi yang terinspirasi oleh sistem saraf biologis yang terdapat pada otak manusia yang memiliki kemiripan dengan operasi logis dan matematis (kalkulus) [14]. Sebuah mode ANN terdiri atas lapisan input, saraf, dan output. Berkaitan dengan topik penelitian yang diangkat, sebuah citra $f(x,y)$ akan dijadikan sebagai input yang melewati lapisan atau jaringan saraf dan kemudian mencapai lapisan output yang menginterpretasi apakah citra tersebut dikategorikan sebagai citra banjir atau bukan (normal). Model ANN memiliki keterbatasan dimana ketepatan (akurasi) klasifikasi bergantung dengan kualitas citra input. Biasanya citra input tidak langsung diproses oleh ANN, akan tetapi diekstrak dulu fitur yang dapat meningkatkan ketepatan prediksinya. Sebagai contoh, pada [15] ANN menerima input fitur visual seperti *facial action unit*, bukan citra input mentahnya.

2.3. Deep Learning

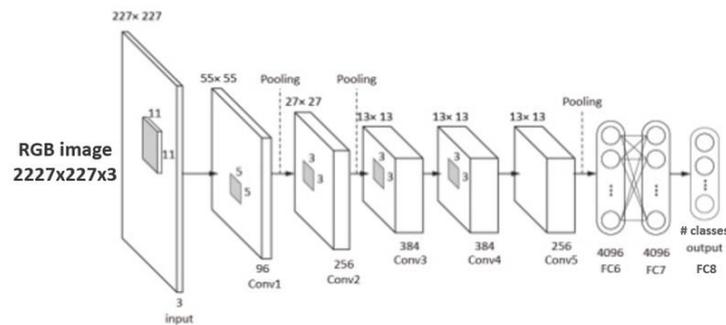
Kelemahan ANN dapat diatasi dengan meningkatkan jumlah lapisan saraf. Pendekatan ini disebut juga dengan model *deep learning*, dimana model jaringan saraf mempelajari fitur terbaik secara otomatis sebelum melakukan klasifikasi pada lapisan output. Model deep learning diberikan pada gambar 2. Pada model tersebut, jumlah lapisan saraf (*hidden layer*) dapat berjumlah lebih dari satu. Namun demikian, *deep learning* tidak sesuai untuk digunakan langsung mengolah citra digital karena citra digital biasanya memiliki 3 lapisan warna (merah, hijau, biru).



Gambar 2. Model lapisan saraf pada *Deep Learning*

2.4. Convolutional Neural Network (CNN)

Untuk mengolah citra input yang terdiri dari 3 lapis warna, model yang lebih sesuai digunakan adalah model CNN [16]. Kelebihan dari model CNN dibandingkan dengan model deep learning standar adalah karena lapisan konvolusinya. Lapisan konvolusi mampu berperan untuk menggantikan proses manual yang secara tradisional dibutuhkan pada proses klasifikasi gambar yaitu untuk melakukan perbaikan kualitas citra, segmentasi dan pemilihan fitur dari setiap label kelas yang akan diprediksi. Terdapat beberapa model CNN yang telah berkembang seperti LeNet [17], AlexNet [18], ResNet [19], GoogleNet/Inception [20], dan VGG [21]. AlexNet memiliki lebih banyak proses konvolusi dibandingkan LeNet sehingga mampu untuk memprediksi lebih banyak kategori objek. Adapun ResNet memiliki lapisan konvolusi yang cukup besar yaitu 34. Pada penelitian deteksi banjir berdasarkan citra, [22] menggunakan AlexNet dan berhasil mencapai akurasi sebesar 73% yang dilatih terhadap citra yang diambil di daerah sungai Sekanak, Kota Palembang, Sumatera Selatan. Sayangnya, pada penelitian mereka tidak dijelaskan nilai dari parameter training yang digunakan. Bentuk dari arsitektur AlexNet ditunjukkan pada gambar 3 [23].



Gambar 3. Arsitektur dari model AlexNet

Perbedaan utama dari kelima model CNN di atas terletak pada kompleksitas arsitekturnya. Namun demikian, model-model tersebut memiliki fundamental lapisan yang serupa yang dijelaskan sebagaimana berikut ini:

- Data masukan yang dapat diterima suatu model CNN adalah input citra yang berasal dari dataset.
- Blok: setiap blok CNN terdiri atas beberapa lapisan konvolusi, ReLU, dan pooling.
 - Lapisan konvolusi mengolah akan memproses piksel-piksel didalam citra input dimana sel atau *node* akan melakukan operasi filtering fitur untuk mengekstrak komponen-komponen visual didalam citra.
 - Kemudian, lapisan ReLU akan melakukan keputusan sekiranya nilai dari hasil operasi filtering dapat diteruskan pada lapis selanjutnya. Contoh fungsi ReLU adalah $f(x)=\max(0,x)$ yang artinya akan menghentikan sinyal jika nilainya negatif atau 0.
 - Berikutnya, lapisan pooling akan mereduksi resolusi citra sehingga pada pengulangan proses diblok selanjutnya dapat diekstrak fitur pada resolusi yang lebih rendah.

- Berikutnya, lapisan *fully-connected* akan menggabungkan seluruh sinyal yang kemudian akan diukur probabilitas label kategori citra menggunakan fungsi softmax..

3. METODOLOGI PENELITIAN

3.1. Pengumpulan Data

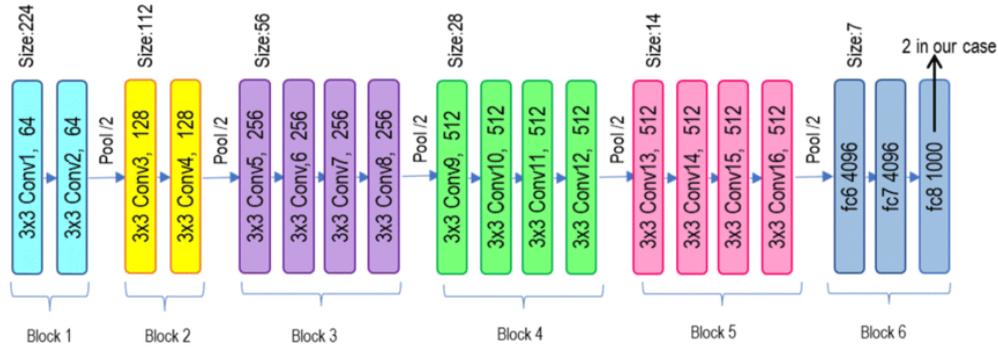
Dataset yang digunakan berasal dari dataset citra banjir perkotaan dan jalanan publik yaitu Roadway Flooding Image Dataset [24] dan Road Vehicle Images Dataset [25]. Pada dataset pertama, kami menggunakan seluruh dataset sejumlah 441 citra banjir. Sedangkan pada dataset kedua, terdapat 2704 citra jalanan perkotaan. Namun, untuk menyeimbangkan label data dari dataset citra banjir, kami hanya menggunakan 441 citra jalanan perkotaan saja. Contoh sampel citra banjir dan tanpa banjir dapat dilihat pada Gambar 4.



Gambar 4. Contoh sampel citra banjir dan tanpa banjir.

3.2. Model VGG19

Citra digital akan diklasifikasi menggunakan model *convolutional neural network*. Disini, kami menggunakan sebuah varian dari model *visual geometry group* (VGG) yaitu VGG19. Angka 19 menunjukkan varian ini terdiri atas 19 lapisan konvolusi (16 lapisan konvolusi yang dibagi menjadi 5 blok [21]). Setiap blok dilengkapi dengan lapisan MaxPool, kemudian ada dan 3 lapisan *fully-connected* yang masing-masing memiliki 4096, 4096, 1000 saraf. Pada lapisan output digunakan 1 lapisan SoftMax. Arsitektur model VGG 19 yang digunakan pada penelitian ini dapat dilihat pada gambar 5 [26].



Gambar 5. Arsitektur model *convolutional neural network* VGG19 [26]

Untuk menjaga performa CNN, kami akan melakukan optimasi selama proses training model usulan. Terdapat 4 parameter yang yang dioptimasi selama proses *training* model VGG19 yaitu *Batch Size*, *Learning Rate*, *Dropout* dan *Epoch* (pengulangan *training*). Nilai eksperimen yang digunakan untuk menemukan hasil optimal diberikan sebagaimana berikut ini:

- *Batch Size* mencakup 10, 20, 30, 40, 50, 100
- *Learning Rate* mencakup 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1
- *Dropout* mencakup 10, 20, 30, 40, 50, 90
- *Epoch* mencakup 10, 20, 30, 40, 50, 100, 150, 200, 300

3.3. Evaluasi Performa Model Usulan

Performa VGG19 dievaluasi berdasarkan akurasi yang diperoleh dari matriks *confusion* sebagaimana yang ditampilkan pada tabel 1.

Tabel 1. Matriks Confusion untuk Label Citra Banjir atau Normal

Citra banjir	Hasil Prediksi	
	positif	negatif
positif	<i>tp</i>	<i>fn</i>
negatif	<i>fp</i>	<i>tn</i>

Variabel *tp* adalah total *true positif* yang artinya hasil prediksi adalah positif (banjir) sesuai dengan foto aslinya yaitu citra banjir. Demikian pula untuk *tn* atau *true negatif* yaitu hasil prediksi citra tidak banjir sesuai kondisi foto aslinya yaitu citra normal. Adapun *fp* dan *fn* menandakan *false positif* dan *false negatif* yang artinya hasil prediksi tidak sesuai dengan foto aslinya dimana *fp* menghasilkan prediksi banjir padahal citra foto normal dan sebaliknya. Dari 4 keadaan ini, dapat diturunkan 4 metrik yang digunakan akurasi, presisi, recall dan *F1-score* yang diukur dengan rumus (1)-(4).

$$Akurasi = \frac{tp + tn}{tp + fp + tn + fn} \quad (1)$$

$$Presisi = \frac{tp}{tp + fp} \quad (2)$$

$$Recall = \frac{tp}{tp + fn} \quad (3)$$

$$F1 \text{ score} = 2 \frac{Presisi \times Recall}{Presisi + Recall} \quad (4)$$

Eksperimen dilakukan pada Google Colab yang dapat diakses secara Cloud melalui internet browser. Bahasa yang digunakan adalah Python dan implementasi VGG19 diadaptasi berdasarkan [27]. Library yang digunakan mencakup pandas, numpy, keras, sklearn, matplotlib, math, datetime, time, tensorflow, dan os.

4. HASIL DAN PEMBAHASAN

Seperti yang telah dijelaskan sebelumnya, optimasi dilakukan terhadap 4 parameter training yaitu *Batch Size*, *Learning Rate*, *Dropout* dan *Epoch* (pengulangan *update* bobot saraf). Dari total 441 dataset, dilakukan pembagian menjadi 3 sub-dataset yaitu data *training*, *validation*, dan *testing*. Resolusi citra yang digunakan adalah 224 x 224 piksel. Pada awal, eksperimen dilakukan dengan menetapkan nilai parameter yaitu *learning rate* sebesar $1e-4$ (0.0001), 50% *dropout*, dan 100 *epoch*. Sebagai catatan, pembulatan nilai persentase untuk metrik-metrik performa dilakukan berdasarkan ketentuan sebelumnya [27].

Hasil eksperimen pada parameter *batch size* ditunjukkan pada tabel 2. Pada nilai 10, akurasi VGG19 hanya mencapai 95.12% sedangkan pada nilai yang tinggi (50 dan 100), akurasi menurun menjadi sekitar 92-93%. Nilai akurasi terbaik dicapai pada nilai 20, 30, dan 40. Namun untuk metrik presisi 97% dicapai hanya pada nilai 20. Adapun waktu training yang dibutuhkan tidak banyak berbeda antara satu dengan lainnya. Nilai *batch size* terbaik yaitu 20 ini kemudian digunakan untuk optimasi parameter berikutnya.

Tabel 2. Eksperimen parameter *Batch Size*

No	<i>Batch Size</i>	Akurasi (%)	Presisi (%)	Recall (%)	F1-score (%)	CPU (detik)
1	10	95.12	95	95	95	6.00
2	20	96.34	97	96	96	5.84
3	30	96.34	96	96	96	5.77
4	40	96.34	97	96	96	5.74
5	50	92.68	94	93	93	5.76
6	100	93.90	95	94	94	5.76

Untuk hasil eksperimen pada parameter *learning rate* ditunjukkan pada tabel 3. Semakin tinggi nilai *learning rate* menghasilkan performa yang semakin rendah. Pada nilai 1, akurasi hanya mencapai 54.88%, dan diikuti oleh nilai presisi, recall dan F1-score masing-masing 76%, 55%, 55%. Adapun hasil terbaik diberikan oleh nilai *learning rate* $1e-5$ yang mencapai akurasi 98.78%. Sedangkan untuk ketiga metrik performa lain yaitu presisi, recall dan F1-score mencapai 99% dan berhasil dicapai oleh waktu *training* selama 8.92 detik. Menariknya Ketika nilai diperkecil ($1e-6$), performa VGG19 mengalami penurunan disertai dengan penambahan waktu komputasi hampir 3x.

Tabel 3. Pencarian *learning rate* terbaik (batch size = 20)

No	<i>Learning Rate</i>	Akurasi (%)	Presisi (%)	Recall (%)	F1-score (%)	CPU (detik)
1	$1e-6$	89.02	89	89	89	21.59
2	$1e-5$	98.78	99	99	99	8.92
3	$1e-4$	97.56	98	98	98	3.76
4	$1e-3$	96.34	96	96	96	3.72
5	$1e-2$	96.34	97	96	96	10.88
6	$1e-1$	91.46	93	91	91	5.79
7	1	54.88	76	55	55	5.83

Eksperimen berikutnya ditunjukkan oleh tabel 4 yaitu optimasi parameter *dropout*. Dari keseluruhan percobaan, performa VGG19 tidak ada yang mencapai 96% untuk metrik apapun. Adapun waktu komputasi rata-rata diatas 19 detik, kecuali untuk 90% dropout (7.56 detik). Hasil eksperimen menampakkan bahwasanya parameter dropout tidak memiliki

pengaruh signifikan atas performa VGG19.

Tabel 4. Pencarian parameter dropout (batch size = 20 dan learning rate = 1e-5)

No	Dropout	Akurasi (%)	Presisi (%)	Recall (%)	F1-score (%)	CPU (detik)
1	10	95.12	95	95	95	21.30
2	20	95.12	95	95	95	19.36
3	30	91.46	92	91	91	21.17
4	40	95.12	95	95	95	21.17
5	50	92.68	93	93	93	21.53
6	90	53.66	59	54	45	7.56

Pada percobaan terakhir, kami ingin melihat sensitifitas parameter *epoch* terhadap performa VGG19 seperti yang ditunjukkan pada tabel 5. Nilai yang digunakan memiliki rentang yang cukup lebar yaitu antara 10 hingga 300. Hasil terbaik dicapai pada nilai 150 epoch yaitu akurasi 97.56% dan 98% untuk ketiga metrik lainnya. Namun demikian percobaan epoch juga belum memberikan signifikansi dibandingkan percobaan pada parameter *learning rate*.

Tabel 5. Eksperimen parameter *epoch* (batch size = 20, learning rate = 1e-5, dan dropout = 20%)

No	Epoch	Akurasi (%)	Presisi (%)	Recall (%)	F1-score (%)	CPU (detik)
1	10	95.12	95	95	95	3.37
2	20	93.90	94	94	94	5.79
3	30	96.34	96	96	96	5.92
4	40	96.34	96	96	96	7.29
5	50	97.56	98	98	98	9.64
6	100	96.34	96	96	96	9.87
7	150	97.56	98	98	98	9.57
8	200	97.56	98	98	98	10.95
9	300	96.34	96	96	96	7.54

Berdasarkan eksperimen-eksperimen tersebut, didapati hasil terbaik akurasi VGG19 untuk mendeteksi citra banjir adalah 98.78% dan 99% untuk ketiga metrik lainnya. Hasil ini dihasilkan oleh dengan parameter *Batch Size* sebesar 20, *Learning Rate* sebesar 1e-5 (0.00001), 50% *Dropout* dan 100 *Epoch*. Jika dibandingkan dengan penelitian yang dilakukan oleh [22] yang hanya mencapai akurasi 73%, model usulan berhasil mencapai akurasi yang lebih baik sebesar 25.78%. Hal ini menunjukkan model VGG19 dapat memprediksi citra banjir dengan lebih baik dibandingkan model AlexNet.

5. KESIMPULAN DAN SARAN

Untuk pengembangan sistem deteksi banjir, penelitian ini telah mencapai performa yang cukup baik yaitu akurasi sebesar 98.78% dan 99% untuk presisi, *recall*, dan *F1-score*. Nilai capaian dari keempat metrik ini dapat dianggap cukup sehingga model VGG19 berpeluang untuk dikembangkan dan disematkan kedalam sistem *monitoring* atau aplikasi deteksi banjir untuk memonitoring kondisi banjir perkotaan. Model VGG19 juga telah terbukti memiliki capaian yang lebih baik dibandingkan model CNN AlexNet. Percobaan yang dilakukan juga telah dilakukan terhadap data publik yang dapat diuji oleh peneliti lainnya. Dapat disimpulkan model VGG19 menghasilkan kinerja yang lebih signifikan untuk memprediksi citra banjir dibandingkan dengan model AlexNet. Namun demikian, masih terdapat model *convolutional neural network* lainnya seperti ResNet, GoogleNet, ataupun MobileNet yang belum diujicoba untuk memprediksi citra banjir. Untuk studi kedepannya, kami merekomendasikan untuk diadakan eksplorasi terhadap model-model tersebut.

DAFTAR PUSTAKA

- [1] B. D. B. Adwil, "Pentingnya Status Pulau Bagi Gazetir Republik Indonesia," *Direktorat Jenderal Bina Administrasi Kewilayahan Kementerian Dalam Negeri Republik Indonesia*, 2022. <https://ditjenbinaadwil.kemendagri.go.id/berita/detail/pentingnya-status-pulau-bagi-gazetir-republik-indonesia>
- [2] A. DLH, "Kerusakan Sungai dan Daerah Aliran Sungai di Indonesia," *Dinas Lingkungan Hidup Kabupaten Grobogan*, 2016. <https://dlh.grobogan.go.id/index.php/info-lh/berita/36-kerusakan-sungai-dan-daerah-aliran-sungai-di-indonesia>
- [3] A. Lidwina, "Berapa Jumlah Kejadian Banjir dalam Lima Tahun Terakhir?," *Databoks*, 2020. <https://databoks.katadata.co.id/datapublish/2020/06/08/berapa-jumlah-kejadian-banjir-dalam-lima-tahun-terakhir>
- [4] V. A. Dihni, "BNPB: Kejadian Bencana Alam Indonesia Capai 3.058 Sepanjang 2021," *Databoks*, 2021. <https://databoks.katadata.co.id/datapublish/2021/12/29/bnpb-kejadian-bencana-alam-indonesia-capai-3058-sepanjang-2021>
- [5] P. N. Rahardjo, "7 Penyebab Banjir di Wilayah Perkotaan yang Padat Penduduknya," *J. Air Indones.*, vol. 7, no. 2, 2018.
- [6] L. Sebastian, "Pendekatan pencegahan dan penanggulangan banjir," 2008.
- [7] M. C. Ulum, "Governance dan capacity building dalam manajemen bencana banjir di Indonesia," *J. Dialog Penanggulangan Bencana*, vol. 4, no. 2, pp. 69–76, 2013.
- [8] C. Angrelia, R. Prihastha, A. C. Mubarak, and W. K. Utami, "Peranan Pemerintah Kota Tangerang dalam Penanggulangan dan Pencegahan Banjir Tahun 2020," *J. Agreg. Aksi Reformasi Gov. dalam Demokr.*, vol. 8, no. 1, 2020.
- [9] A. Findayani, "Kesiap siagaan masyarakat dalam penanggulangan banjir di Kota Semarang," *J. Geogr. Media Inf. Pengemb. Dan Profesi Kegeografian*, vol. 12, no. 1, pp. 102–114, 2018.
- [10] A. Y. S. Siahaan and others, "Implementasi Program Pembangunan Drainase dalam upaya Penanggulangan Banjir di Kota Medan," 2019.
- [11] A. N. Aurora and others, "Studi Komparasi Kebijakan Penanggulangan Banjir di Wilayah DKI Jakarta pada Era Kepemimpinan Basuki Tjahja Purnama dan Anies Baswedan," *J. Polit. Gov. Stud.*, vol. 10, no. 3, pp. 138–161, 2021.
- [12] R. Sulistyowati, H. A. Sujono, and A. K. Musthofa, "Sistem pendeteksi banjir berbasis sensor ultrasonik dan mikrokontroler dengan media komunikasi sms gate way," *Skripsi. Fak. Teknol. Ind. Inst. Teknol. Adhi Tama Surabaya*, 2015.
- [13] A. A. a'ziziyah, I. I. Nugroho, R. Sabillillah, B. A. S. Aji, and K. Amiroh, "Perbandingan Sistem Deteksi Banjir Menggunakan Algoritma Naive Bayes Dan K-NN Berbasis IOT," *IJCIT (Indonesian J. Comput. Inf. Technol.)*, vol. 7, no. 1, 2022.
- [14] M. JL, "A logical calculus of ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, pp. 115–133, 1943.
- [15] H. Akbar, S. Dewi, Y. A. Rozali, L. P. Lunanta, N. Anwar, and D. Anwar, "Exploiting Facial Action Unit in Video for Recognizing Depression using Metaheuristic and Neural Networks," in *2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI)*, 2021, vol. 1, pp. 438–443.
- [16] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*, Springer, 1982, pp. 267–285.
- [17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 1097–1105, 2012.
- [19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [20] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Prepr. arXiv1409.1556*, 2014.
- [22] M. A. Hidayat, N. L. Husni, and F. Damsi, "Pendeteksi Banjir Dengan Image Processing Berbasis Convolutional Neural Network (CNN) pada Kamera Pengawas: Image Processing Based Flood

- Detector Using Convolutional Neural Network (CNN) Within Surveillance Camera,” *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 2, no. 2, pp. 10–18, 2022.
- [23] A. Khvostikov, K. Aderghal, J. Benois-Pineau, A. Krylov, and G. Catheline, “3D CNN-based classification using sMRI and MD-DTI images for Alzheimer disease studies,” *arXiv Prepr. arXiv1801.05968*, 2018.
- [24] C. Sazara, M. Cetin, and K. Iftekharuddinbh, “Image Dataset for Roadway Flooding,” v1, 2019. doi: 10.17632/t395bwcvbw.1.
- [25] A. Yeafi, “Road Vehicle Images Dataset,” 2021. <https://www.kaggle.com/datasets/ashfakyeafi/road-vehicle-images-dataset?resource=download>
- [26] A. Khattar and S. M. K. Quadri, “Generalization of convolutional network to domain adaptation network for classification of disaster images on twitter,” *Multimed. Tools Appl.*, pp. 1–28, 2022.
- [27] A. Jovandy, “Image Classification using VGG19 and resnet50,” *Github*, 2020. <https://github.com/Adithia88/Image-Classification-using-VGG19-and-Resnet>