

Penerapan Algoritma Convolutional Neural Network (CNN) Untuk Identifikasi Lahan Kosong Di Kota Tegal Berdasarkan Citra Google Earth

¹Mohammad Amin Triwinanto, ²Aang Alim Murtopo*, ³Syefudin, ⁴Gunawan Gunawan
^{1, 2, 3, 4}STMIK YMI Tegal, Indonesia.

¹20185008@mhs.stmik-tegal.ac.id, ²aang.alim@stmik-tegal.ac.id, ³syefudin@stmik-tegal.ac.id,
⁴gunawan@stmik-tegal.ac.id

ABSTRAK

Lahan kosong memiliki berbagai macam jenis. Setiap jenis lahan kosong memiliki macam-macam tertentu dengan model yang beragam. Dalam menentukan jenis lahan kosong maka perlu dilakukan sebuah klasifikasi dengan menggunakan metode *Convolutional Neural Network* (CNN). Dengan penggunaan CNN dapat dilakukan ekstraksi sebuah fitur kemudian fitur-fitur tersebut akan menjadi data dalam menentukan klasifikasi jenis lahan kosong. Data gambar lahan kosong yang dikumpulkan dari data augmentasi adalah sebanyak 120 gambar dengan jenis lahan kosong tambak, rawa, pemukiman, dan sawah. Keempat kelas jenis lahan kosong tersebut memiliki perbandingan data latih 70% dan data uji 30%. Masing-masing kelas menggunakan empat *convolutional layer* dengan filter 32, 32, 64, dan 64 dan menggunakan *pool size* sebesar 2x2 dengan neuron (*hidden layer*) sebanyak 512. Pengujian *website image classification* dengan menggunakan metode *confusion matrix* didapatkan akurasi sebesar 80,5% dari pengujian yang dilakukan pada data uji.

Kata Kunci: lahan kosong, klasifikasi, Convolutional Neural Network, *image classification*, *confusion matrix*

PENDAHULUAN

Informasi lahan kosong telah banyak digunakan berbagai pihak seperti pemerintah, peneliti, dan lain-lain. Permukaan bumi tidak homogen sama sekali, melainkan mengandung variasi seperti air, tanah kosong, rumput, aspal, dan lain-lain. Lahan kosong sangat sering kita jumpai setiap hari. Salah satu problem dalam visi komputer yang telah lama dicari solusinya adalah klasifikasi objek pada citra secara umum. Bagaimana menduplikasi kemampuan manusia dalam memahami informasi citra, agar komputer dapat mengenali objek pada citra selayaknya manusia (Zufar et al., 2016). Proses feature engineering yang digunakan pada umumnya sangat terbatas dimana hanya dapat berlaku pada dataset tertentu saja tanpa kemampuan generalisasi pada jenis citra apapun. Hal tersebut dikarenakan berbagai perbedaan antar citra antara lain perbedaan sudut pandang, perbedaan skala, perbedaan kondisi pencahayaan, deformasi objek, dan sebagainya (Budi et al., 2023). *Convolutional Neural Network* (CNN) merupakan *neural network* yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *deep neural network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada ekspresi manusia dengan akurasi 80%. CNN memiliki akurasi yang tinggi sebesar 99% dan dapat memberikan hasil yang baik dalam mengenali sebuah objek pada sebuah pengenalan citra. CNN memiliki fungsi untuk mengklasifikasikan sebuah citra. Hasil penelitian yang diperoleh untuk klasifikasi kualitas kayu kelapa menunjukkan bahwa arsitektur GoogleEarth memiliki performansi klasifikasi dengan rata-rata akurasi 84,89% pada setiap lapisan (Winiarti et al., 2021).

TINJAUAN PUSTAKA

Pada penelitian ini membahas tentang salah satu pengklasifikasian *image* tekstur lahan pada tutupan lahan merujuk pada unsur-unsur fisik yang terlihat di permukaan bumi, termasuk berbagai material seperti rumput, aspal, pepohonan, tanah terbuka, air, dan material fisik lainnya dengan *convolutional neural network*. *Image classification* digunakan untuk menentukan jenis lahan dari sebuah tekstur lahan kosong. Penggunaan lahan mengacu pada bagaimana manusia memanfaatkan lahan, baik di wilayah perkotaan maupun pedesaan. Klasifikasi menggunakan deep learning merupakan metode yang akurat dan bertujuan untuk membantu mendapatkan informasi mengenai tutupan lahan di suatu wilayah diharapkan dapat membantu pengguna dalam mengenali jenis lahan dari sebuah lahan kosong (Ichwan & Hadi, 2023).

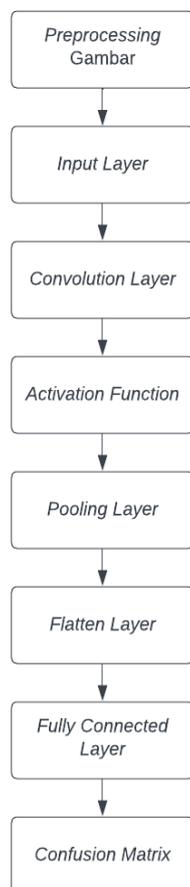
Penelitian ini membahas tentang Algoritma *Deep Learning* (DL) semakin banyak digunakan dalam aplikasi penginderaan jauh (Assidhiqi, 2021), salah satunya *Convolutional Neural Networks* (CNN). CNN adalah arsitektur hierarki yang dapat dilatih pada kumpulan data skala besar untuk melakukan pengenalan dan deteksi objek (Ayu et al., 2021). Setiap proses pemahaman gambar yaitu klasifikasi gambar dan objek menggunakan model CNN memerlukan suatu input data pelatihan. Penelitian ini menggunakan algoritma Mask R-CNN yaitu salah satu model yang dikembangkan dari model Faster-RCNN yaitu *region-based convolutional neural networks* yang dimana mampu mendeteksi objek dalam gambar sekaligus menghasilkan *segmentasi instance* dimana bertujuan untuk mendeteksi kelas objek bersama dengan prediksi kotak pembatas dalam gambar. Adapun memiliki suatu tujuan untuk menggunakan potensi CNN untuk membantu melakukan fitur klasifikasi otomatis yang menghindari digitasi secara manual untuk mengatasi tantangan ekstraksi bangunan pada citra resolusi tinggi pada daerah Desa Campurejo, Kabupaten Gresik. Algoritma Mask-RCNN digunakan untuk mengatasi tantangan proses segmentasi bangunan dimana algoritma ini dapat membantu dalam melakukan pendeteksian batas tidak hanya berupa *semantic segmentation* namun juga *instance segmentation*. Wilayah Desa Campurejo dipilih karena memiliki tingkat kompleksitas yang tinggi. Dimana wilayah ini memiliki sebuah eksklave yang membagi wilayahnya menjadi dua wilayah. Hal ini membuat Desa Campurejo memiliki kondisi dan bentang alam yang berbeda-beda sehingga berpengaruh pula terhadap persebaran dan bentuk geometri bangunannya (Frasetya et al., 2018).

METODE PENELITIAN

CNN merupakan salah satu dari algoritme *deep learning*, perkembangan dari *multilayer perceptron* (MLP) yang digunakan untuk mengolah sebuah data citra dalam bentuk 2 dimensi. Algoritma ini dapat mengklasifikasikan sebuah data yang telah diberikan label. Label tersebut untuk mengklasifikasikan sebuah data baru dengan data yang sudah diuji (Tilasefana & Putra, 2023). Model CNN merupakan sebuah jaringan *neuron multilayer* yang terdiri atas 2 bagian, yaitu ekstraksi fitur dan pelatihan data. Pelatihan data dilakukan untuk masuk ke dalam kelas. Ekstraksi fitur terdiri atas lapisan *feature map* dan bagian pengambilan fitur yang berbeda dari gambar awal, yaitu: konvolusi (Ulfah Nur Oktaviana et al., 2021). CNN terdiri dari terdiri atas *input layer*, *convolution layer*, *activation function*, *pooling layer*, *flatten layer*, dan *fully connected layer* (Rasywir et al., 2020). Tahap yang dilalui pada penelitian ini terdiri atas pengumpulan data kemudian dilanjutkan tahap *preprocessing* gambar. Gambar yang sudah melalui tahap *preprocessing* dikelompokkan dengan menggunakan arsitektur CNN. Arsitektur CNN meniru otak manusia terutama dalam karakterja korteks dan sistem sarafnya. Adapun tahapan dalam arsitektur CNN terdiri dari *input layer*, *convolution layer*, *activation function*, *pooling layer*, *flatten layer*, dan *fully connected layer*. Metode yang digunakan untuk proses pengujian yaitu *confusion matrix*.

A. Preprocessing Gambar

Pengambilan data gambar dari *Google Image* berjumlah 120 gambar berupa *image* tekstur jenis lahan kosong yang dibagi menjadi empat kategori, yaitu: lahan kosong tambak, lahan kosong rawa, lahan kosong sawah, dan lahan kosong pemukiman kemudian dimasukkan ke dalam *dataset* (Fikri et al., 2022). Dilakukan perbandingan data sebesar 70% untuk data *training*, yaitu pembelajaran data oleh komputer, dan 30% data *test*, yaitu pengujian data yang belum pernah dilihat oleh komputer. Semua gambar dari data *training* akan dilakukan beberapa proses, seperti *rescale* yang berfungsi untuk mengubah skala *grayscale* (sesuai warna aslinya), rotasi, dan fungsinya untuk melakukan perputaran pada gambar (Ayudhawara et al., 2023). Gambar awal memiliki warna sesuai dengan jenis lahan kosong dan posisi. *Horizontal flip* dapat membuat gambar yang tadinya ada di posisi kiri berpindah ke posisi kanan/sebaliknya. *Fill_mode* dengan *nearest* yang dapat berfungsi untuk mengisi ruang kosong dari setiap piksel dengan mengambil nilai terdekatnya. Untuk validasi data hanya dilakukan proses *rescale* karena komputer hanya mempelajari data *train*. Agar komputer makin banyak belajar maka gambar diperbanyak dengan menggunakan *preprocessing* (augmentasi) ini. Data augmentasi adalah sebuah cara yang sering digunakan untuk memperkecil *overfitting* dalam sebuah model. Dengan melakukan augmentasi data, manfaat yang dimiliki dapat menghasilkan sebuah data baru yang menggunakan sebuah perubahan-perubahan yang ada pada data asli. Augmentasi data memungkinkan untuk meningkatkan generalisasi data atau model dalam sebuah citra (Hansel et al., 2021). Augmentasi dapat menghasilkan data *training* baru dari data aslinya. Tahapan proses yang dilakukan dalam penelitian ini ditunjukkan pada Gambar 1. sebagai berikut:



Gambar 1. Tahapan proses yang dilakukan penelitian

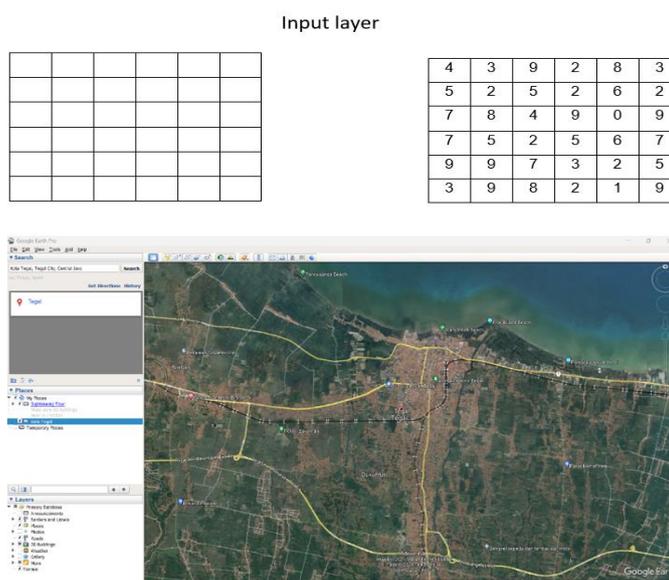
B. Input Layer

Layer ini merupakan tempat awal mula gambar dimasukkan sebelum menjalankan sebuah proses klasifikasi gambar. *Layer* ini merupakan *layer* pertama dalam arsitektur CNN. Pada bagian ini nantinya semua gambar dapat dimasukkan dan diproses ke *layer* berikutnya. Mata manusia melihat sebuah gambar, namun sebuah komputer jika melihat sebuah gambar akan direpresentasikan sebagai kumpulan sebuah piksel yang nantinya *input layer* memiliki ukuran gambar sebesar 128 x 128 piksel.

Pada simulasi ini, *input layer* memiliki ukuran 6 x 6 piksel sehingga nantinya seluruh gambar yang masuk ke dalam arsitektur untuk diproses ukuran dikonversi menjadi ukuran 6x 6 piksel, seperti yang ditunjukkan pada Gambar 2. Semua ukuran dibuat sama sehingga tidak ada data yang memiliki piksel di luar piksel yang ditentukan. Jika terjadi perbedaan ukuran, nantinya nilai piksel Gambar 1. dengan gambar yang lainnya tidak dapat dibandingkan.

C. Convolution Layer

Pada *layer* ini dilakukan sebuah proses untuk memperoleh sebuah piksel yang didasarkan nilai piksel sendiri dan nilai piksel tetangga dengan melibatkan sebuah matriks yang biasa dibidang *kernel* dalam menampilkan nilai bobotnya (Fauzi et al., 2019). Dengan adanya *convolution layer* ini, gambar dapat diekstrak fiturnya untuk diambil pola bentuk dan warna. Dengan demikian, komputer dapat membedakan antara lahan kosong tambak, lahan kosong rawa, lahan kosong sawah, dan lahan kosong pemukiman dengan melihat warna dari tiap lahan kosong pada *layer* sebelumnya.



Gambar 2. Simulasi input layer

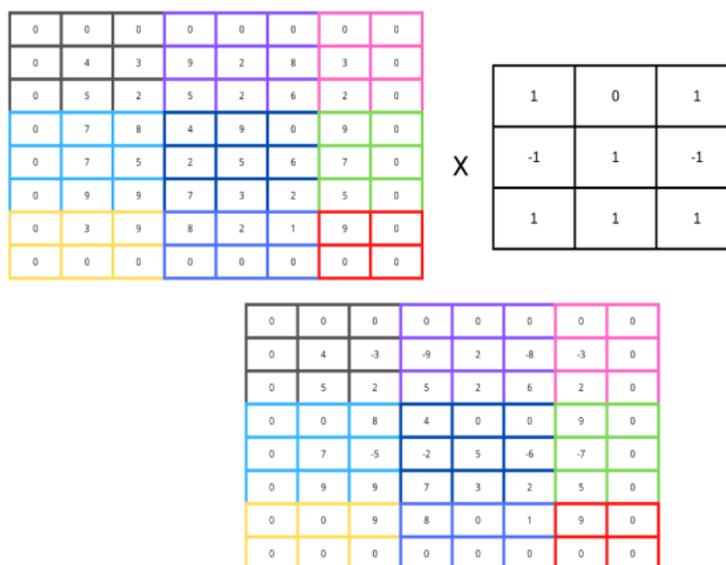
Filter size yang digunakan adalah 3 x 3 karena *input layer* adalah 128 x 128 piksel. Jumlah *convolution layer* terdiri dari 4 buah. Jika menggunakan kurang atau lebih dari empat *layer*, akan mengurangi nilai akurasi untuk jumlah filter baru dengan nilai 32, 32, 64, 64. Pemilihan angka ini didasari dengan pengembangan dari penelitian pendahulu. Jika menggunakan sesuai dengan layer, nilai akurasinya menjadi kurang baik.

Penggunaan *padding same* dilakukan agar data yang terbuang dikurangi. Angka filter tidak ditentukan sebelumnya. Karena tidak ditentukan di awal, isi dari semua piksel pada *filter* otomatis sesuai dengan fitur yang akan diekstrak pada *fully connected layer*, seperti deteksi

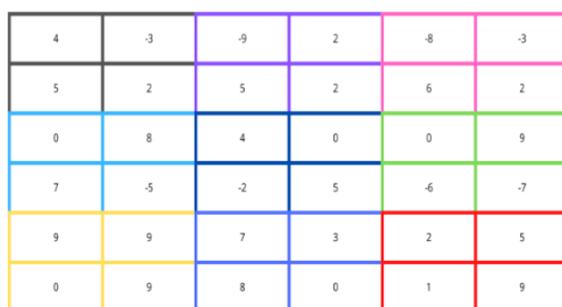
tepi dan lain-lain. Contoh simulasi perhitungan *convolution* dengan *layer input* dan *filter* berukuran 3x3 piksel. Karena *padding same*, maka *image input* akan menjadi 8 x 8 piksel dengan *padding 0* dan *image 6x 6* piksel menjadi 8 x 8 piksel lalu dikalikan dengan filter baru, seperti yang ditunjukkan pada Gambar 3. Piksel akan kembali ke ukuran semula, yaitu berukuran 6 x 6 piksel, seperti yang ditunjukkan pada Gambar 4.

D. Activation Function

Layer ini merupakan sebuah salah satu dari fungsi aktivasi yang menerapkan *retrified linear unit* (Relu) yang berfungsi untuk membatasi nilai negatif sehingga nantinya diubah menjadi angka nol.



Gambar 3. Simulasi perhitungan filter dengan *input*



Gambar 4. Simulasi penggunaan *size* semula

Fungsi dari *activation* salah satunya, yaitu untuk membuat semua nilai yang bernilai minus menjadi positif dengan menggunakan aktivasi Relu sehingga nantinya semua nilai yang bernilai minus akan dibuat 0. Angka minus hanya akan memperlambat komputasi komputer dalam melakukan *train* sebuah model karena filter yang bernilai minus tidak dapat diekstrak nilainya sehingga akan dihapus. Fungsi *activation* ini ditunjukkan pada Gambar 5.

E. Pooling Layer

Pooling layer memiliki *filter* dan *stride* tertentu, yaitu dengan ukuran filter 2x2 dan dengan nilai *stride* berukuran 1. Fungsi dari *pooling layer* ini, yaitu mencari nilai maksimum dari sebuah *kernel* (matriks). Proses *pooling layer* ditunjukkan pada Gambar 6. Penggunaan *pooling layer* sebesar 2x2 karena jika ukuran dimensi hanya 1x1, akan tetap mempertahankan dimensinya. Namun, jika menggunakan 3x3, akan semakin banyak data yang terbuang (Trisiawan & Yuliza, 2022). Semakin banyak data yang terbuang, kemungkinan akan terdapat fitur penting yang hilang. Dengan demikian, fitur penting yang didapatkan juga akan semakin sedikit.

Pooling layer memiliki fungsi untuk mengurangi komputasi komputer sehingga hanya mengambil nilai piksel terbesar dari setiap *pool size*. Nilai yang lebih kecil kurang berguna untuk di ekstrak fitur-fiturnya sehingga hanya akan menambah komputasi komputer. *Pool size* sebesar 2 x 2 agar lebih sedikit data yang terbuang. Simulasi *pooling layer* ditunjukkan pada Gambar 6.

4	0	0	2	0	0
5	2	5	2	6	2
0	8	4	0	0	9
7	0	0	5	0	0
9	9	7	3	2	5
0	9	8	0	1	9

Gambar 5. Simulasi *activation function*

4	0	0	2	0	0
5	2	5	2	6	2
0	8	4	0	0	9
7	0	0	5	0	0
9	9	7	3	2	5
0	9	8	0	1	9

5	5	6
8	5	9
9	8	9

Gambar 6. Simulasi *pooling layer*

F. Flatten Layer

Pada layer ini dibentuk suatu *vector* satu dimensi dari nilai matriks pada tahap sebelumnya. Pada *pooling layer* gambar masih memiliki ukuran 2 dimensi. Dengan dilakukan hal ini, maka hasil *flatten layer* ini akan menjadi nilai *input* pada *fully connected layer* untuk mendapatkan hasil klasifikasi (Leidiyana, 2013).

Flatten layer memiliki fungsi menjadi masukan *layer* berikutnya untuk dikumpulkan dan diekstrak fiturnya menjadi satu bagian. Dari bentuk satu dimensi nantinya seluruh nilai yang ada di *flatten layer* akan masuk ke dalam *fully connected layer* yang juga memiliki bentuk satu dimensi. Pada *layer* sebelumnya dari 3x3 piksel akan menjadi satu dimensi yang dilakukan dari kiri atas ke bawah kanan. Simulasi *flatten layer* ditunjukkan pada Gambar 7.

G. Fully Connected Layer

Pada *layer* ini didapatkan nilai masukan (matriks) dari hasil *flatten layer* pada tahap sebelumnya. Setiap masukan akan terhubung dengan *neuron* yang berada di *hidden layer* untuk mendapatkan hasil berdasarkan nilai yang ada pada *hidden layer*. *Fully connected layer* ditunjukkan pada Gambar 8. *Feature map* terbuat dari ekstraksi fitur yang masih berupa *array* multidimensi. *Feature map* perlu diubah menjadi vektor agar dapat digunakan sebagai nilai masukan vektor dari lapisan yang terhubung sepenuhnya.

Fully connected layer merupakan lapisan *neuron* aktif pada lapisan sebelumnya dan terhubung dengan *neuron* pada lapisan berikutnya sehingga menyerupai jaringan syaraf tiruan (Rininda et al., 2023). *Fully connected layer* berisi kumpulan seluruh fitur dari sebuah gambar mulai dari *edge detection*, pola bentuk dari tiap piksel, warna, dan lain-lain. Setiap *layer* yang ada pada *fully connected layer* akan saling berhubungan satu dengan lainnya sehingga membentuk sebuah jaringan *neuron*. Keluarannya akan menghasilkan sebuah kelas lahan kosong, yaitu kelas tambak, rawa, sawah, dan pemukiman. Simulasi ini ditunjukkan pada Gambar 9.

Bagian kiri berupa piksel warna-warni berisi *flatten* yang merupakan masukan dari *layer* sebelumnya. *Fully connected layer* berisi ekstrak dari sebuah gambar yang telah dipelajari sebelumnya sehingga *flatten layer*, *fully connected layer*, dan *output layer* akan saling terhubung. Komputer dapat menyimpulkan kelas yang didapatkan dari keterhubungan antar *layer* tersebut. *Confusion matrix* merupakan sebuah tabel perhitungan akurasi yang dapat menyatakan keberhasilan uji yang benar atau salah dalam melakukan klasifikasi (Putro et al., 2020).

5
5
6
8
5
9
9
8
9

Gambar 7. Simulasi *flatten layer*

HASIL DAN PEMBAHASAN

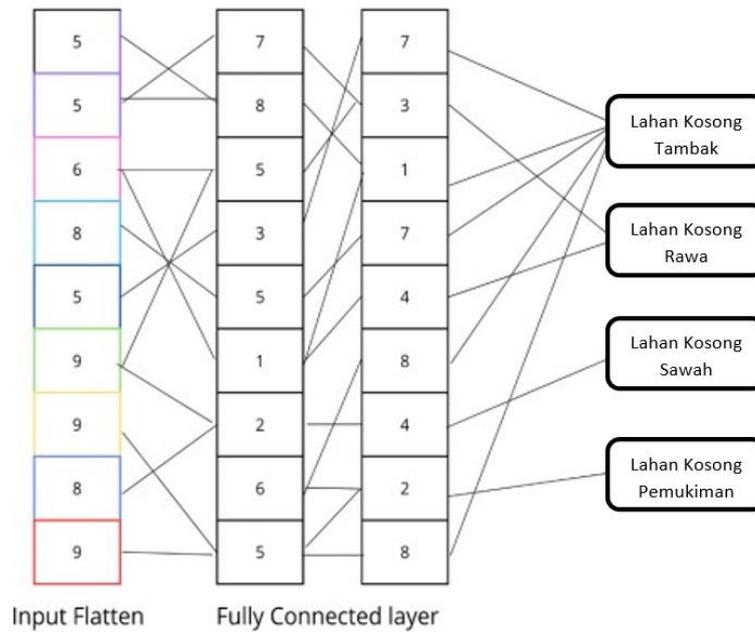
Hasil Persentase menampilkan gambar yang sebelumnya telah diunggah dan menampilkan hasil prediksi berupa jenis (*class*) lahan kosong (lahan kosong tambak, lahan kosong rawa, lahan kosong sawah, dan lahan kosong pemukiman) dan hasil persentase (*probability*). Tampilan Hasil Persentase ditunjukkan pada Gambar 9.

A. Implementasi Convolutional Neural Network

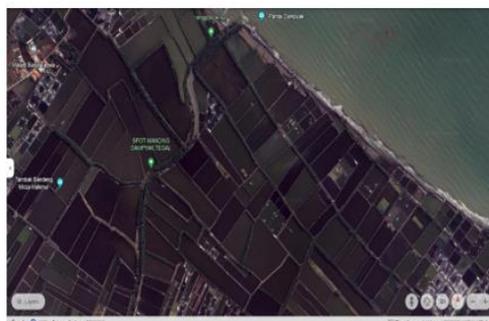
Arsitektur CNN pada *Convolution Layer* 1 terdiri atas 32 *filter* baru dengan *kernel size* sebesar 3x3 dan dengan *input shape* 128, 128, 3 (*pixel width*, *pixel height*, *image channel*).

- Padding* yang sama ditambahkan *pooling max* (*Pooling Layer* 1) dengan *pool size* sebesar 2x2 pada *Convolution Layer* 2 yang terdiri dari 32 *filter* baru dan ukuran sebesar 3x3.

b. *Padding* yang sama dilakukan *pooling max* (*Pooling Layer 2*) yang terdiri dari *pool size* sebesar 2x2 pada



Gambar 8. Simulasi *fully connected layer*



Lahan Kosong Tambak



Lahan Kosong Rawa



Lahan Kosong Sawah



Lahan Kosong Pemukiman

Gambar 9. Hasil presentasi lahan kosong

c. *Convolution Layer 3* yang terdiri dari 64 *filter* baru dan ukuran sebesar 3x3.
d. *Padding* yang sama dilakukan *pooling max* (*Pooling Convolution Layer 3* yang terdiri dari 64 *filter* baru dan ukuran sebesar 3x3. (*Pooling Layer 3*) yang terdiri dari

pool size sebesar 2×2 . Untuk *layer* konvolusi yang terakhir (*Convolution Layer 4*) terdiri dari 64 *filter* baru dan ukuran sebesar 3×3 .

- e. *Padding* yang sama dilakukan *pooling max* (*Pooling Layer 4*) yang terdiri dari *pool size* sebesar 2×2 *flatten* agar menjadi sebuah *vector* yang nantinya digunakan sebagai masukan *fully-connected layer*.

Neuron sebanyak 512 akan menggunakan Relu untuk mengubah semua piksel yang negatif menjadi positif dengan kelas sebanyak 4 buah, yaitu: lahan kosong tambak, lahan kosong rawa, lahan kosong sawah, dan lahan kosong pemukiman.

B. Pengujian Algoritma CNN

Pengujian menggunakan *confusion matrix* dilakukan secara manual. Pengujian dilakukan per satu gambar pada data *test* yang sudah dipisahkan sebelumnya. Sebelum pengujian menggunakan *confusion matrix* dapat dilakukan, harus melewati tahap pelatihan. Tahap pelatihan melibatkan pelatihan data dan pengujian dengan data validasi. *Step per epoch*, yaitu *batch* yang dieksekusi tiap *epoch*, data *train* dibagi *batch size* data *training* dengan *epoch* sebanyak 160 kali. Pada *step validation*, jumlah data *validation* dibagi *batch size* data *validation*. Dilakukan pembagian *batch* ini untuk menghemat waktu pelatihan, dengan catatan *epoch* setidaknya sesuai dengan *batch data train* atau lebih tinggi agar semua data dapat dilatih. Fungsi tahap *train*, yaitu untuk melihat hasil model CNN dari pelatihan data.

Sebelum melakukan pemasukan pada tabel *confusion matrix*, maka semua data *test* harus diuji satu per satu. Pada kelas lahan kosong tambak terdapat 3 kesalahan, pada kelas lahan kosong rawa terdapat 1 kesalahan, pada kelas lahan kosong sawah terdapat 2 kesalahan, dan pada kelas lahan kosong pemukiman terdapat 1 kesalahan. Semua hasil yang salah ini ditunjukkan pada Tabel 1.

Tingkat akurasi dari data *test* pada metode pengujian *confusion matrix* adalah sebesar 80,5% yang didapatkan dari jumlah gambar seluruh kelas benar dibagi seluruh gambar. Kelas yang benar akan berwarna kuning yang disebut dengan *True Positive* (TP) dan angka yang tidak memiliki warna disebut *False Positive* (FP).

$$\text{Accuracy} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \quad (1)$$

Hasil perhitungannya adalah:

$$\text{Accuracy} = \frac{6+8+7+8}{6+1+0+8+1+2+2+7+1+0+8} = \frac{29}{36} = 0,805$$

Tabel 1. Hasil Confusion Matrix

Confusion Matrix		Kelas Prediksi			
		Lahan Kosong Tambak	Lahan Kosong Rawa	Lahan Kosong Sawah	Lahan Kosong Pemukiman
Kelas Sebenarnya	Lahan Kosong Tambak	6	0	2	0
	Lahan Kosong Rawa	1	8	0	0
	Lahan Kosong Sawah	2	1	7	1
	Lahan Kosong Pemukiman	0	0	0	8

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka dapat disimpulkan perbandingan terhadap *dataset* sebesar 70% (*datatrain*) dan 30% (*data test*). Dengan arsitektur CNN yang memiliki kriteria empat *convolution layer*, maka dibuat *convolution layer* pertama dan kedua terdiri atas 32 filter dan *convolution layer* ketiga dan keempat terdiri atas 64 filter. Empat *pooling layer* memiliki *pool size* masing-masing sebesar 2x2 dan memiliki neuron sebanyak 512 untuk 4 kelas jenis lahan kosong, yaitu tambak, rawa, sawah, dan pemukiman. Pelatihan data *train* dan pengujian validasi dilakukan sebanyak 160 kali (*epoch*). Dengan arsitektur ini juga dilakukan pengujian data *test* menggunakan *confusion matrix* terhadap kriteria data *test* dari kelas lahan kosong tambak, lahan kosong rawa, lahan kosong sawah, dan lahan kosong pemukiman. Hasil persentasenya sebesar 80,5 %.

REFERENSI

- Ayu, C., Kinasih, S., & Hidayat, H. (2021). *Ekstraksi Data Bangunan Dari Data Citra Unmanned Aerial Vehicle Menggunakan Metode Convolutional Neural Networks (CNN) (Studi Kasus: Desa Campurejo, Kabupaten Gresik) Building Data Extraction of Unmanned Aerial Vehicle Image Data Using Convolutional Neural Networks (CNN) Method (Case Study: Campurejo Village, Gresik Regency)*. 17(1), 81–92.
- Ayudhawara, A., Satrio, B., Poetro, W., Qomaruddin, M., Sultan, I., & Semarang, A. (2023). Deteksi Ketersediaan Tempat Parkir menggunakan Mask R-CNN (Studi Kasus : Fakultas Teknologi Industri Universitas Islam Sultan Agung). In *Seminar Riset Mahasiswa-Computer & Electrical (SERIMA-CE (Vol. 1, Issue 1)*.

- Budi, R., Harianto, R. A., & Setyati, E. (2023). Segmentasi Citra Area Tumpukan Sampah Dengan Memanfaatkan Mask R-CNN. *Journal of Intelligent System and Computation*, 5(1), 58–64. <https://doi.org/10.52985/insyst.v5i1.305>
- Fauzi, S., Eosina, P., & Laxmi, G. F. (n.d.). *Implementasi Convolutional Neural Network Untuk Identifikasi Ikan Air Tawar*. 163–167.
- Fikri, A. A., Darmawan, A., Hilmanto, R., Banuwa, I. S., Agustiono, A., & Agustiana, L. (2022). Pemanfaatan platform Google Earth Engine dalam Pemantauan Perubahan Tutupan Lahan di Taman Hutan Raya Wan Abdul Rachman. *Journal of Forest Science Avicennia*, 5(1), 46–57. <https://doi.org/10.22219/avicennia.v5i1.19938>
- Frasetya, B., Setiati, Y., & Septianugraha, R. (2018). *E-Jurnal Agroekoteknologi Tropika Pemanfaatan Citra Landsat 8 dan Google Earth untuk Identifikasi Lahan Sawah di Kecamatan Cibiru Kota Bandung*. 7(3). <https://ojs.unud.ac.id/index.php/JAT428>
- Hansel, L., #1, G., Bunyamin, H., & Si, S. (2021). *Penggunaan Augmentasi Data pada Klasifikasi Jenis Kanker Payudara dengan Model Resnet-34* (Vol. 3). <https://cs231n.github.io/convolutional-networks/>
- Ichwan, M., & Hadi, S. (2023). MIND (Multimedia Artificial Intelligent Networking Database Kinerja Model EfficientNetV2M dalam Klasifikasi Citra Tutupan dan Penggunaan Lahan. *Journal MIND Journal / ISSN*, 8(2), 203–216. <https://doi.org/10.26760/mindjournal.v8i2.203-216>
- Leidiyana, H. (2013). PENERAPAN ALGORITMA K-NEAREST NEIGHBOR UNTUK PENENTUAN RESIKO KREDIT KEPEMILIKAN KENDARAAN BEMOTOR. In *Jurnal Penelitian Ilmu Komputer, System Embedded & Logic* (Vol. 1, Issue 1).
- Oleh, D., & Assidhiqi, F. (2021). *PENGEMBANGAN SISTEM DETEKSI HUNIAN PARKIR MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK*.
- Putro, H. F., Vulandari, R. T., & Saptomo, W. L. Y. (2020). Penerapan Metode Naive Bayes Untuk Klasifikasi Pelanggan. *Jurnal Teknologi Informasi Dan Komunikasi (TIKOMSiN)*, 8(2). <https://doi.org/10.30646/tikomsin.v8i2.500>
- Rasywir, E., Sinaga, R., Pratama, Y., Dinamika, U., & Jambi, B. (2020). *Analisis dan Implementasi Diagnosis Penyakit Sawit dengan Metode Convolutional Neural Network (CNN)*. 22(2). <https://doi.org/10.31294/p.v21i2>
- Rininda, G., Santi, I. H., & Kirom, S. (2023). PENERAPAN SVM DALAM ANALISIS SENTIMEN PADA EDLINK MENGGUNAKAN PENGUJIAN CONFUSION MATRIX. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 7, Issue 5).
- Tilasefana, R. A., & Putra, R. E. (2023). Penerapan Metode Deep Learning Menggunakan Algoritma CNN Dengan Arsitektur VGG NET Untuk Pengenalan Cuaca. *Journal of Informatics and Computer Science*, 05.

Trisiawan, I. K., & Yuliza, Y. (2022). Penerapan Multi-Label Image Classification Menggunakan Metode Convolutional Neural Network (CNN) Untuk Sortir Botol Minuman. *Jurnal Teknologi Elektro*, 13(1), 48. <https://doi.org/10.22441/jte.2022.v13i1.009>

Ulfah Nur Oktaviana, Ricky Hendrawan, Alfian Dwi Khoirul Annas, & Galih Wasis Wicaksono. (2021). Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih Resnet101. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(6), 1216–1222. <https://doi.org/10.29207/resti.v5i6.3607>

Winiarti, S., Saputro, M. Y. A., & Sunardi, S. (2021). Deep Learning dalam Mengidentifikasi Jenis Bangunan Heritage dengan Algoritma Convolutional Neural Network. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 5(3), 831. <https://doi.org/10.30865/mib.v5i3.3058>

Zufar, M., Setiyono, B., & Matematika, J. (2016). *Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time* (Vol. 5, Issue 2).