

Sistem Hitung Dan Klasifikasi Objek Dengan Metode Convolutional Neural Network

M. Hamsy Romario
Fakultas Teknik/Teknik Elektro
Universitas Mercubuana
Jakarta, Indonesia
mhamsyromario@gmail.com

Trie Maya Kadarina
Fakultas Teknik/Teknik Elektro
Universitas Mercubuana
Jakarta, Indonesia
trie.maya@mercubuana.ac.id

Eko Ihsanto
Fakultas Teknik/Teknik Elektro
Universitas Mercubuana
Jakarta, Indonesia
Eko.ihsanto@mercubuana.ac.id

Abstrak— Sistem klasifikasi objek ini di rancang untuk melakukan klasifikasi dan perhitungan jumlah objek terdeteksi pada suatu gambar. menggunakan metode Convolutional Neural Network yang telah dilatih, Metode CNN merupakan salah satu metode deep learning yang mampu melakukan proses pembelajaran mandiri untuk pengenalan objek, ekstraksi objek dan klasifikasi serta dapat diterapkan pada citra resolusi tinggi yang memiliki model distribusi nonparametrik. Kemudian gambar yang telah diterima dijalankan menggunakan Bahasa pemrograman python pada laptop operasional menggunakan platform open source spyder3. Input system ini adalah citra 2 dimensi dengan skala minimal 400 x 400 pixel dan skala maksimal 1600 x 1600 pixel. Setelah program dijalankan maka outputnya adalah sebuah citra yang dengan keterangan jumlah wajah terdeteksi dan keterangan framing terhadap pola wajah pada gambar output. Penelitian menggunakan tiga kelompok gambar percobaan, gambar kelompok pertama berisikan gambar dengan objek manusia, kelompok gambar kedua berisikan objek manusia asli yang di campur dengan karikatur, kelompok gambar ketiga berisi gambar kartun berperawakan manusia. Pada percobaan kelompok pertama hasil deteksi mencapai 80%, Dan kelompok kedua di dapat hasil deteksi mencapai 75%. Dan pada percobaan gambar kelompok ketiga system tidak mendeteksi adanya pola wajah manusia. Hasil penelitian ini menunjukkan bahwa penggunaan metode CNN berpotensi untuk pendekatan pengenalan objek secara otomatis dalam membedakan jenis pola wajah manusia sebagai bahan

pertimbangan interpreter dalam menentukan objek pada citra.

Kata Kunci — *Convolutional neural network, Python, Spyder 3.*

I. PENDAHULUAN

Rumusan masalah yang penulis ambil diawali pada suatu ketika penulis dalam sebuah perjalanan jarak jauh antar kota menggunakan Bus sebagai sarana transportasi darat, saya sebagai penulis kemudian memperhatikan ada satu aktifitas di mana di tengah perjalanan menuju suatu kota, Bus yang saya tumpangi menepi masuk ke rest area. Kemudian terlihat dimana ada petugas yang masuk kedalam bus untuk melakukan penghitungan jumlah penumpang, setelah di hitung petugas mengambil gambar menggunakan media handphone. Saya berfikir aktifitas tadi adalah aktifitas rutin untuk keperluan pendataan penumpang bus.

Karena melihat adanya efisiensi akan waktu dan aktifitas sebagai tujuan, yang mendorong penulis untuk membuat Penelitian ini, maka kemudian dirancang sebuah sistem yang dapat mengklasifikasi objek manusia dan menghitungnya dalam satu waktu menjadi batasan masalah dalam Penelitian saya sebagai penulis. Dan untuk menjalankan sistem tersebut penulis menggunakan beberapa Library seperti openCV dan Numpy dengan pilihan bahasa pemrograman Python pada sebuah platform spyder. Program python di rancang sedemikian simpel agar dapat mudah dipahami dalam pengaplikasiannya. Dengan memasukkan sebuah citra gambar sebagai bagian input kemudian program dijalankan maka akan

dihasilkan outputan berupa gambar lengkap dengan keterangan dan kalkulasi objek manusia pada gambar tersebut.

II. PENELITIAN TERKAIT

A. Grayscale

Citra grayscale adalah citra yang hanya memiliki satu buah kanal sehingga yang ditampilkan hanyalah nilai intensitas atau dikenal juga dengan istilah derajat keabuan. Grayscale adalah mengkonversi citra berwarna yang memiliki tiga kanal yaitu RGB (Red, Green, Blue) menjadi citra grayscale yang memiliki satu buah kanal. Convolutional Neural Network (CNN).

B. Object Detection

Pendeteksian objek adalah salah satu bagian ilmu dari computer vision dalam memproses gambar yang bertujuan dapat mendeteksi objek yang ada pada suatu gambar. Objek yang dimaksud pada topik ini adalah manusia. Setiap objek memiliki fitur khusus yang membedakan objek pada suatu citra. Ada banyak bermacam teknik untuk mendapatkan fitur pada suatu citra sesuai kebutuhan objek yang akan dideteksi.

C. Histograms of Oriented Gradients (HOG)

Adalah deskriptor fitur yang digunakan dalam pengolahan citra untuk tujuan deteksi objek. Teknik ini menghitung kemunculan orientasi gradien pada bagian gambar yang dilokalisasi. Fitur yang dihasilkan dari metode ini berupa vektor yang kemudian vektor ini digunakan untuk mengklasifikasikan objek menggunakan algoritma klasifikasi Support Vector Machine (SVM) Tahap awal dari metode ini adalah menghitung gradien dari setiap piksel citra, yaitu dengan menghitung perubahan pada nilai pixel di sepanjang sumbu-x untuk horisontal dan sumbu-y untuk vertikal pada setiap pixel. Untuk menghitung gradien dapat digunakan kernel sebagai berikut.

D. Neural Network

Convolutional Neural Network (CNN) termasuk dalam jenis deep learning karena kedalaman jaringannya. Deep learning adalah cabang dari machine learning yang dapat mengajarkan komputer untuk melakukan pekerjaan selayaknya manusia, seperti komputer dapat belajar dari proses training (Deng & Yu, 2013). CNN merupakan operasi konvolusi yang menggabungkan beberapa lapisan pemrosesan, menggunakan beberapa elemen yang beroperasi secara paralel dan terinspirasi oleh sistem saraf biologis [1].

Pada CNN setiap neuron dipresentasikan dalam bentuk 2 dimensi, sehingga metode ini cocok untuk pemrosesan dengan input berupa citra [2]. Arsitektur jaringan dengan menggunakan CNN ditunjukkan pada Gambar 1. Struktur CNN terdiri dari input, proses ekstraksi fitur, proses klasifikasi dan output. Proses ekstraksi dalam CNN terdiri dari

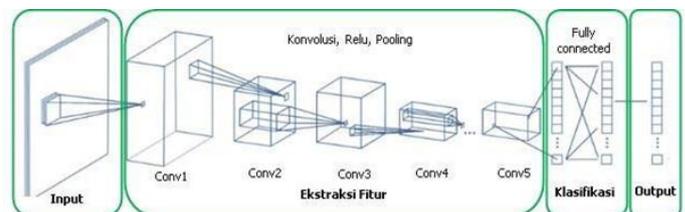
beberapa lapisan tersembunyi atau hidden layer, yaitu lapisan konvolusi, fungsi aktivasi (ReLU), dan pooling. CNN bekerja secara hierarki, sehingga output pada lapisan konvolusi pertama digunakan sebagai input pada lapisan konvolusi selanjutnya. Pada proses klasifikasi terdiri dari fully-connected dan fungsi aktivasi (softmax) yang outputnya berupa hasil klasifikasi [3].

E. Haar Cascade

Haar Cascade merupakan CNN Architecture Salah satu algoritma yang dapat digunakan dalam mendeteksi wajah adalah Haar Cascade Classifier. Haar Cascade Classifier atau Haar Like Feature atau algoritma Viola-Jones merupakan suatu algoritma yang biasa digunakan untuk mendeteksi objek pada citra digital, salah satunya adalah wajah. Haar like feature adalah suatu fitur berbentuk persegi (rectangular features) yang memberikan indikasi secara spesifik pada sebuah citra. Nama haar berasal dari suatu fungsi matematika yaitu Haar Wavelet. Pada awalnya, algoritma ini digunakan untuk mengolah citra yang dilihat dari nilai RGB pada setiap pikselnya, namun tidak efektif. Kemudian Viola-Jones mengembangkannya sehingga terbentuk nama haar like feature.

F. Lapisan Konvolusi

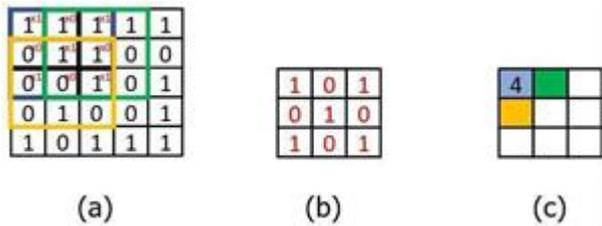
Lapisan konvolusi menggunakan filter untuk mengekstraksi objek dari citra input. Filter ini berisi bobot yang digunakan untuk mendeteksi karakter dari objek seperti tepi, kurva, atau warna. Konvolusi akan menghasilkan transformasi linear dari citra input yang sesuai dengan informasi spasial pada data. Filter diaplikasikan secara berulang sehingga menghasilkan serangkaian bidang receptive. Terdapat parameter yang dapat diubah untuk memodifikasi sifat tiap lapisan, yaitu



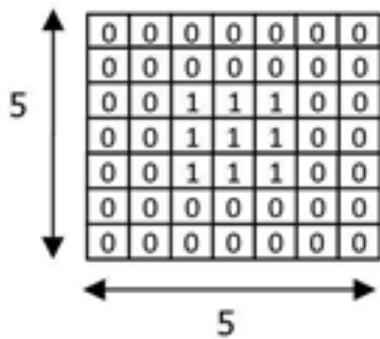
Gambar 1. Arsitektur CNN

Ukuran filter, stride dan padding. Stride mengontrol bagaimana filter diterapkan pada data input dengan bergerak sepanjang ukuran piksel yang telah ditentukan. Padding adalah penambahan ukuran piksel dengan nilai tertentu disekitar data input agar hasil dari bidang receptive tidak terlalu kecil sehingga tidak banyak informasi yang hilang. Nilai ini biasanya nol sehingga disebut dengan zero padding. Hasil dari bidang receptive berupa data tunggal. Output dari proses konvolusi ini dijadikan sebagai input untuk lapisan konvolusi

selanjutnya [4]. Ilustrasi proses stride dan padding ditampilkan pada Gambar 2 dan Gambar 3.



Gambar 2. Operasi konvolusi dengan stride 1 (a) Input data 5x5 (b) filter 3x3 (c) bidang receptive 3x3



Gambar 3. Operasi zero padding 2 pada data 3x3

G. Fungsi Aktivasi ReLU

ReLU (Rectification Linear Unit) merupakan operasi untuk mengenalkan nonlinearitas dan meningkatkan representasi dari model. Fungsi aktivasi ReLU adalah $f(x) = \max(0, x)$ [5]. Nilai output dari neuron bisa dinyatakan sebagai 0 jika inputnya adalah negatif. Jika nilai input adalah positif, maka output dari neuron adalah nilai input aktivasi itu sendiri [6].

H. Pooling

Pooling atau subsampling adalah pengurangan ukuran matriks. Terdapat dua macam pooling yang sering digunakan yaitu average pooling dan max pooling [7]. Nilai yang diambil pada average pooling adalah nilai rata-rata sedangkan pada max pooling adalah nilai maksimal [8].

I. Fully Connected Layer

Lapisan fully connected layer merupakan kumpulan dari proses konvolusi [9]. Lapisan ini mendapatkan input dari proses sebelumnya untuk menentukan fitur mana yang paling berkorelasi dengan kelas tertentu. Fungsi dari lapisan ini adalah untuk menyatukan semua node menjadi satu dimensi [10].

J. Fungsi Aktivasi Softmax

Fungsi aktivasi softmax digunakan untuk mendapatkan hasil klasifikasi. Fungsi aktivasi menghasilkan nilai yang

diinterpretasi sebagai probabilitas yang belum dinormalisasi untuk tiap kelas. Nilai kelas dihitung dengan menggunakan fungsi softmax [11], yang ditunjukkan oleh Persamaan 1.

$$y_{ijk} = \frac{e^{x_{ijk}}}{\sum_{t=1}^D e^{x_{ijt}}} \dots\dots\dots(1)$$

Dimana:

- y_{ijk} = vektor yang berisi nilai antara 0 dan 1.
- X = vektor yang berisi nilai yang didapatkan dari lapisan fully-connected terakhir.

Fungsi kesalahan klasifikasi dihitung dengan Persamaan 2.

$$l(x, c) = -\log \frac{e^{x_c}}{\sum_{k=1}^C e^{x_k}} = -x_c + \log \sum_{k=1}^C e^{x_k} \dots\dots\dots(2)$$

Dimana:

- $l(x,c)$ = membandingkan prediksi (x) dan label (c).
- x = vektor dari probabilitas akhir. $p(k) = x_k, k=1..C$
- C = banyak kelas.

Untuk mengontrol overfitting, pooling layer digunakan untuk mengurangi representasi ukuran spasial dan mengurangi jumlah parameter. Lapisan dropout memberikan aturan untuk menghilangkan atau menjaga neuron dengan beberapa nilai probabilitas p yang bernilai antara 0 dan 1 [12]. Lapisan dropout berguna untuk memudahkan penggolongan kelasnya.

III. METODOLOGI PENELITIAN

A. Hardware

Personal Computer (PC)/Laptop

Berfungsi sebagai tempat untuk menyimpan basis data dan beberapa program pendukung. Pada Penelitian ini digunakan laptop dengan spesifikasi harddisk 500GB, memory 2GB, Software Operation Windows10.

Kamera

Pada sistem ini, kamera yang digunakan adalah kamera web, atau yang sering disebut webcam. Untuk pemilihan kamera, sebaiknya kita memilih webcam yang memiliki kecepatan serta resolusi yang bagus dan juga harus memiliki kemampuan yang baik dalam adaptasi terhadap cahaya di sekelilingnya. Kamera web yang dipakai pada Penelitian ini memiliki resolusi 2 Mega piksel.

Dalam Metode ini menggunakan citra RGB dengan minimum pixel 400 x 400 dan maksimal pixel 1600 x1600. Kapasitas citra atau gambar input tidak dapat terlalu besar maupun terlalu kecil sebab menyesuaikan dengan skala setingan yang program tetapkan untuk menjamah detail pada citra.



(a)



(b)



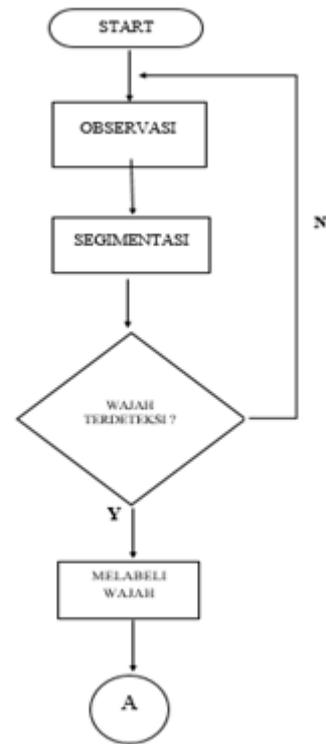
(c)

Gambar 4. Kelompok Gambar: (a) manusia, (b) Manusia dan Karikatur, (c) Kartun

B. Diagram Alir

Diagram Alir pada penelitian ini dimulai dari Start, program dijalankan. Observasi pada gambar input. Object segmentation, program memeriksa setiap frame image yang masuk untuk diperiksa dan dicari objek yang dimaksud. Program memeriksa apakah pada frame terdeteksi wajah. Bila tidak terdeteksi, program kembali ke awal. Dan apabila pada gambar terdeteksi wajah, program dilanjutkan ke tahap berikutnya. Kemudian program akan memberi label

keterangan penumpang atau orang yang terdeteksi dengan membuat frame pada wajah yang terdeteksi. Kemudian reset kernel untuk kembali ke kondisi awal. Program kembali ke awal.



Gambar 5. Diagram Alir

C. Program

```

@author: mr.hamsy import numpy as np import cv2 as cv

wajah=cv.CascadeClassifier('haarcascade_frontalface_default.xml')
img=cv.imread('GAMBAR/mas.jpg')
img_gray=cv.cvtColor(img,cv.COLOR_BGR2GRAY)

deteksi_wajah=wajah.detectMultiScale(img_gray,1.1,5)
font=cv.FONT_HERSHEY_SIMPLEX
jumlah=0

for(x,y,w,h) in deteksi_wajah:
    jumlah=jumlah+1
    cv.putText(img,"peserta",(x,y-10),font,0.75,(0,0,255),2,cv.LINE_AA)
    cv.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
    
```

```

roi_gray=img_gray[y:y+h,x:x+w]
roi_color=img[y:y+h,x:x+w]
cv.putText(img, "Data Peserta M hamsy
romario",(10,60),font,1,(1,1,1),3,cv.LINE_AA)
cv.putText(img,"Jumlah Terdeteksi : "+str(jumlah)+ "
Orang",(10,30),font,1,(0,0,0),2,cv.LINE_AA)
cv.imshow('img',img)
cv.waitKey(0)
cv.destroyAllWindows()
    
```

“import numpy as np” dan “import cv2 as cv” untuk menjalankan perintah Library agar bisa kemudian di proses oleh platform.

“wajah=cv.CascadeClassifier('haarcascade_frontalface_default.xml')” di gunakan untuk memberi tahu bahwa yang akan proses adalah deteksi wajah.

“img=cv.imread('GAMBAR/mas.jpg)’” perintah untuk membaca mendeteksi gambar mas pada folder GAMBAR.

“img_gray=cv.cvtColor(img,cv.COLOR_BGR2GRAY)” ini perintah untuk menseleksi pewarnaan pada objek dalam gambar sehingga program dapat menentukan apakah itu adalah objek yang di maksud.

“deteksi_wajah=wajah.detectMultiScale(img_gray,1.1,5)” skala ketentuan yang di ditetapkan untuk mendeteksi gambar dimana y=1.1 dan x = 5.

“font=cv.FONT_HERSHEY_SIMPLEX” adalah perintah penggunaan font.

“cv.putText(img,"peserta",(x,y-10),font,0.75,(0,0,255),2, cv.LINE_AA)”

cv.putText merupakan perintah untuk membuat keterangan berupa text dengan ketebalan huruf dan mengatur posisinya pada output gambar.

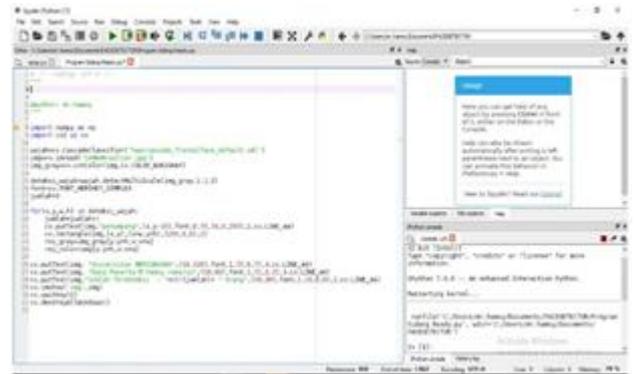
IV. HASIL DAN ANALISA

Implementasi CNN

Terdapat tiga tahap dalam mengimplementasikan CNN, yaitu training, validasi dan tes. Tahap training adalah tahap utama untuk melatih jaringan mempelajari data input. Kemudian jaringan tersebut diuji pada data validasi. Apabila memberikan hasil yang baik, maka jaringan tersebut dapat digunakan untuk melakukan klasifikasi data dengan data tes.

Implementasi Software

Menggunakan aplikasi platform spyder kita dapat menjalankan sebuah program dengan bahasa Python, Berikut adalah tampilan bahasa program yang digunakan.



Gambar 6. Implementasi Software

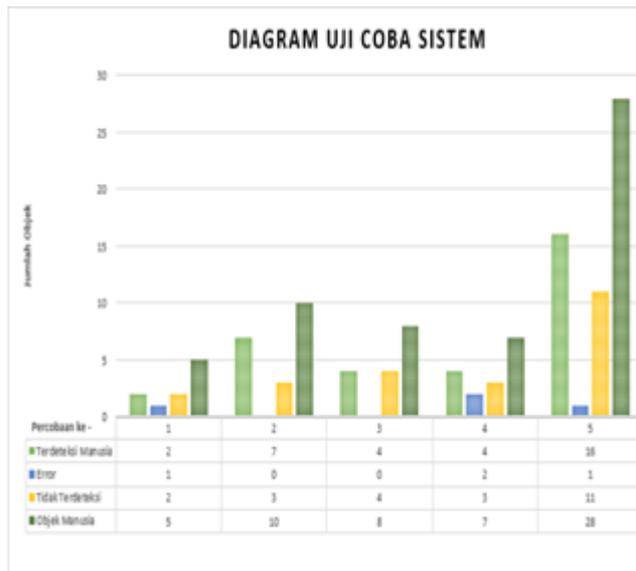
Dengan bahasa program pada gambar 6 kita dapat mendeteksi dan menjumlahkan objek manusia dalam satu gambar. Program ini di gunakan secara berulang ulang untuk mendeteksi gambar satu persatu.

Hasil Percobaan

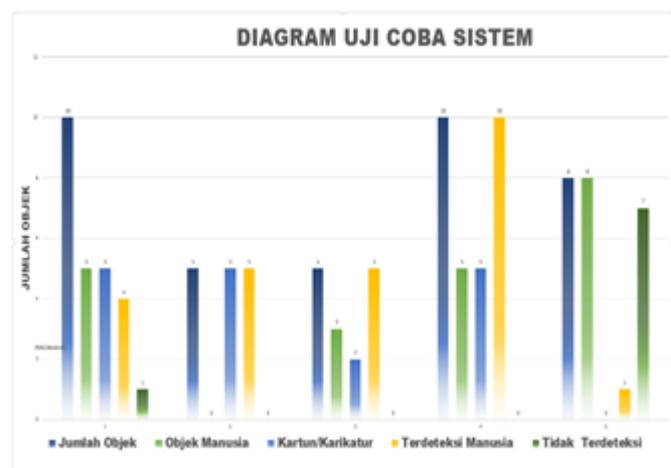
Tabel 1. Hasil Percobaan

No	Jumlah Objek Manusia	Jumlah Objek Mirip Manusia	Jumlah detek	Jumlah Error	T1 (s)	T2 (s)
1	5	0	2	1	3	2
2	10	0	7	0	2	2
3	8	0	4	0	3	2
4	7	0	4	2	4	3
5	28	0	16	1	3	4
6	10	5	4	0	5	4
7	5	5	5	0	4	3
8	5	2	5	0	3	3
9	10	5	10	0	3	2
10	8	0	1	1	3	3
Jml	96	17	58	5		

Dari hasil percobaan pada tabel 1, terdapat jumlah objek yang terbagi menjadi 2 jenis yaitu manusia dan menyerupai/bukan manusia. Dari hasil 10 kali pengujian dengan 113 pendeteksian, sistem dapat mendeteksi sekitar 63% dengan error 0.06%.



(a)



(b)

Gambar 7. (a) Grafik hasil pengujian, (b) Bentuk objek yang diuji

V. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan dengan melalui beberapa tahap perencanaan, perancangan, pembuatan serta pengujian maka dapat diambil kesimpulan sebagai berikut:

- Pada setiap percobaan untuk menjalankan program skala gambar sebagai input memiliki skala batas minimal dimensi dengan size 400 x 400, dan skala batas maksimal dimensi 1600 x 1600, maka program dapat memproses dengan baik.

- Masih terdapat error, dimana error adalah (mendeteksi pola wajah manusia yang pada gambar aktualnya tidak terdapat gambar wajah manusia).
- Dari hasil percobaan sampel di dapat hasil, dimana program berhasil membedakan objek manusia dan bukan manusia dengan terdapat sedikitnya terjadi error sebanyak satu pada tiap dua kali percobaan dengan skala keberhasilan 85% (Persen).
- Berhasil mendeteksi detail wajah manusia dengan baik, di buktikan ketika di minta mendeteksi karikatur berupa manusia, maka program berhasil mendeteksi gambar dengan pola wajah manusia identik
- Program ini telah di latih untuk mengenal pola spesifik pada wajah di mana dapat membaca posisi letak dua mata dan posisi hidung dan mulut. Maka semua hal terlihat sangat menyerupai pola itu akan di data sebagai objek wajah.
- Ketika diminta mendeteksi gambar kartun SpongeBob SquarePants, hasil output deteksi tidak menampilkan frame objek manusia. Artinya program ini berhasil membedakan dengan detail yang baik terhadap pola wajah manusia, dimana tingkat keberhasilan mencapai 95% (Persen).

UCAPAN TERIMA KASIH

Terima kasih kami ucapkan kepada semua pihak yang telah membantu menyelesaikan penelitian ini. Serta terima kasih kepada universitas mercu buana dan tim editorial Jurnal Teknologi Elektro atas dipublikasikannya penelitian ini.

DAFTAR PUSTAKA

- [1] Hu, F., Xia, G. S., Hu, J., & Zhang, L. (2015). Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing*, 7(11), 14680– 14707. <https://doi.org/10.3390/rs71114680>
- [2] Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2016). Convolutional Neural Networks for Large- Scale Remote-Sensing Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2), 645–657. <https://doi.org/10.1109/TGRS.2016.2612821>
- [3] Katole, A. L., Yellapragada, K. P., Bedi, A. K., Kalra, S. S., & Siva Chaitanya, M. (2015). Hierarchical Deep Learning Architecture for 10K Objects Classification. *Computer Science & Information Technology (CS & IT)*, (September), 77–93. <https://doi.org/10.5121/csit.2015.51408>
- [4] Castelluccio, M., Poggi, G., Sansone, C., Verdoliva, L. (2015). Land Use Classification in Remote Sensing Images by Convolutional Neural Networks. Diambil dari

- <https://arxiv.org/pdf/1508.00092.pdf>
<https://doi.org/10.1080/19475705.2017.1315619>
- [5] Heaton, J. (2015). Artificial Intelligence for Humans: Deep learning and neural networks of Artificial Intelligence for Humans Series. CreatespaceIndependent Publishing Platform.
- [6] Kim, J., Sangjun, O., Kim, Y., & Lee, M. (2016). Convolutional Neural Network with Biologically Inspired Retinal Structure. *Procedia Computer Science*, 88, 145–154. <https://doi.org/10.1016/j.procs.2016.07.418>
- [7] Bejiga, M. B., Zeggada, A., Nouffidj, A., & Melgani, F. (2017). A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery. *Remote Sensing*, 9(2). <https://doi.org/10.3390/rs9020100>
- [8] Zhi, T., Duan, L. Y., Wang, Y., & Huang, T. (2016). Two- stage pooling of deep convolutional features for image retrieval. In 2016 IEEE International Conference on Image Processing (ICIP) (hal. 2465– 2469). <https://doi.org/10.1109/ICIP.2016.7532802>
- [9] Hijazi, S., Kumar, R., & Rowen, C. (2015). Image Recognition Using Convolutional Neural Networks. *Cadence Whitepaper*, 1–12
- [10] Albelwi, S., & Mahmood, A. (2017). A Framework for Designing the Architectures of Deep Convolutional Neural Networks. *Entropy*, 19, 242.
- [11] Vedaldi, A., & Lenc, K. (2015). MatConvNet: Convolutional Neural Networks for MATLAB. In *Proceedings of the 23rd ACM International Conference on Multimedia* (hal. 689–692). New York, NY, USA: ACM. <https://doi.org/10.1145/2733373.2807412>
- [12] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15, 1929– 1958. <https://doi.org/10.1214/12-AOS1000>.