

## Klasifikasi 10 Spesies Monyet Berdasarkan Citra Menggunakan Convolutional Neural Network

Amien Aziz<sup>1</sup>, Rafdi Reyhan Zhafari<sup>2</sup>, Mayanda Mega Santoni<sup>3</sup>  
Program Studi Informatika / Fakultas Ilmu Komputer  
Universitas Pembangunan Nasional Veteran Jakarta  
Jl. RS. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia  
amienaziz55@gmail.com<sup>1</sup>, rafdirz@upnvj.ac.id<sup>2</sup>, megasantoni@upnvj.ac.id<sup>3</sup>

**Abstrak.** Dengan perkembangan pesat kecerdasan buatan, sekarang memungkinkan untuk melakukan klasifikasi citra secara otomatis. Klasifikasi spesies pada hewan kerap kali dilakukan secara manual sehingga membutuhkan sumber daya yang banyak. Pada penelitian ini dilakukan klasifikasi citra monyet dengan menggunakan deep learning. Klasifikasi dilakukan dengan menggunakan data citra atau gambar dari 10 spesies monyet menggunakan Convolutional Neural Network (CNN). Jumlah data yang digunakan pada penelitian ini sebanyak 1370 gambar. Untuk meningkatkan variasi data, maka dilakukan proses Augmentasi citra. Pembagian data latih dan uji sebesar 1098 data latih dan 272 data uji. Hasil yang diperoleh pada penelitian ini adalah model CNN dapat mengklasifikasikan jenis monyet dengan tingkat akurasi 78%.

**Kata Kunci:** CNN, gambar, klasifikasi, monyet.

### 1 Pendahuluan

Monyet merupakan bagian dari primata yang ada di bumi, saat ini jumlah spesies monyet di bumi terdapat lebih dari 260 spesies[5]. Spesies monyet dapat dibedakan dari warna rambut dan ukuran dari masing-masing spesies [11], di antara spesies monyet ada yang berukuran kecil dengan dominan warna abu-abu dan ada pula monyet berambut lebat dengan warna dominan coklat dsb. Satu hal yang menjadi ciri khusus dari monyet adalah biasanya terdapat ekor pada bagian belakang tubuhnya[5].

Salah satu cara yang dapat digunakan untuk menentukan spesies monyet adalah dengan data citra dari tiap-tiap spesies. Menurut Pulung dalam buku “Pengolahan Citra Digital” Citra adalah suatu gambaran atau kemiripan dari suatu objek[2], dari pengertian tersebut dapat disimpulkan bahwa citra yang didapatkan dari suatu objek dapat digunakan untuk mewakili gambaran suatu objek. Agar citra yang kita dapatkan dapat diproses maka dibutuhkan untuk merubah citra tersebut dari citra analog ke citra digital[2].

Citra digital artinya citra yang dapat diproses dan diolah oleh komputer[2], dengan demikian dapat memanfaatkan cara-cara atau algoritma-algoritma untuk pemrosesan data citra seperti mendapatkan fitur yang diinginkan, merubah sudut pandang citra sampai merubah ukuran dari citra tersebut. Salah satu cara untuk memanfaatkan data citra adalah dengan melakukan klasifikasi data citra dengan menggunakan algoritma CNN (Convolutional Neural Network).

Convolutional Neural Network atau yang disingkat dengan CNN merupakan salah satu teknik atau algoritma yang terkenal pada deep learning, CNN sering digunakan pada masalah-masalah yang membutuhkan pendekatan terhadap machine learning[9], CNN merupakan salah satu algoritma terbaik untuk penggunaan terhadap klasifikasi gambar. Teknik atau algoritma yang digunakan pada CNN adalah dengan adanya convolutional layer, pooling layer, dan softmax layer[4]. Dengan algoritma tersebut CNN memungkinkan pengklasifikasian gambar dengan mempelajari gambar bagian per bagian pada tiap layer nya[9].

## 2 Landasan Teori

### 2.2 Augmentasi Gambar

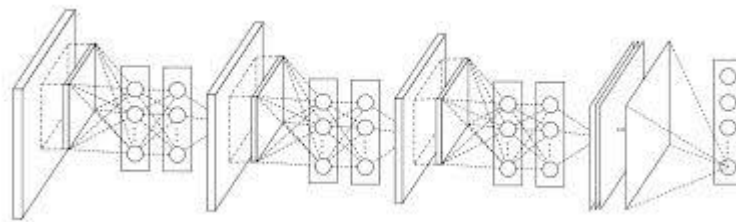
Augmentasi Gambar adalah proses dimana gambar yang kita punya akan dimanipulasi. Gambar dari dataset yang sudah dipisah menjadi latih dan validasi akan kemudian dilakukan rotasi, pergeseran vertikal dan horizontal, pembesaran/pegecilan, dan flip, yang kemudian disimpan kedalam kelas dan tipe (latih dan validasi) masing-masing. Semua gambar kemudian akan dimasukkan ke dalam model klasifikasi untuk ekstraksi fitur.

Augmentasi gambar dilakukan untuk menambahkan varietas dari tiap-tiap gambar yang ada sehingga menambahkan tingkat ketepatan untuk gambar yang memiliki resolusi rendah, *not-centered*, *flipped*, dan lain-lain atau bisa kita bilang gambar 'jelek' saat klasifikasi.

### 2.2 CNN

Convolutional Neural Network (CNN) adalah kelas deep feed-forward artificial neural network yang biasa digunakan dalam masalah computer vision seperti klasifikasi citra. Perbedaan CNN dari jaringan "plain" multilayer perceptron (MLP) adalah penggunaan lapisan convolutional, pooling, dan non-linearitas seperti tanh, sigmoid, dan ReLU.

Pertama adalah lapisan konvolusi. Lapisan konvolusi menggunakan filter untuk melakukan operasi konvolusi pada data input. Satu filter dapat dilihat sebagai matriks. Selama operasi konvolusi, filter bergeser secara horizontal (dengan ukuran langkah tertentu yang disebut steps), kemudian bergerak secara vertikal (juga menggunakan steps yang ditentukan) untuk baris berikutnya, hingga seluruh gambar telah dipindai. Himpunan keluaran filter membentuk matriks baru yang disebut peta fitur.



**Gambar. 1.** *CNN Architecture*

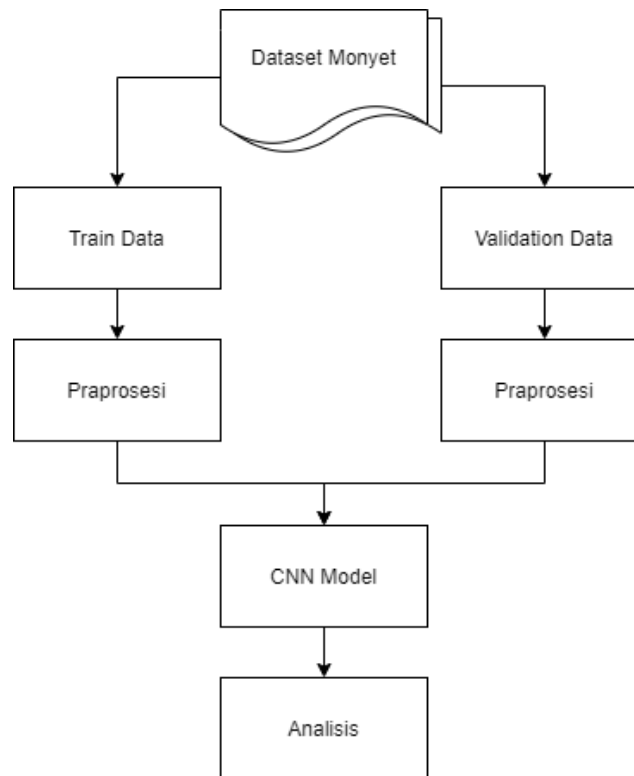
Lapisan Pooling memiliki komponen umum yang sama dengan lapisan konvolusi kecuali: 1) filter disebut kernel tidak memiliki nilai; 2) output dari kernel adalah nilai maksimal atau rata-rata dari area yang dihentikannya; dan 3) ukuran spasial data input tidak diubah melalui pooling layer. Layer pooling hanya mengembalikan nilai. Oleh karena itu, parameter dari pooling layer adalah ukuran kernel, ukuran langkah (steps), dan tipe pooling yang digunakan. Selain itu, lapisan yang terhubung penuh biasanya dimasukkan ke dalam ekor CNN.

Terakhir, fungsi aktivasi digunakan untuk memperkenalkan non-linearitas dalam perhitungan. Tanpa itu, model hanya akan mempelajari pemetaan secara linier. Fungsi aktivasi yang umum digunakan saat ini adalah fungsi ReLU. ReLU umumnya digunakan lebih dari tanh dan sigmoid karena ditemukan bahwa itu sangat mempercepat konvergensi penurunan gradien stokastik dibandingkan dua fungsi lainnya. Selanjutnya, dibandingkan dengan komputasi ekstensif yang dibutuhkan oleh tanh dan sigmoid, ReLU diimplementasikan hanya dengan melakukan thresholding nilai matriks pada nol.

### 3 Metodologi

#### 3.1 Tahapan Penelitian

Tahapan pada penelitian ini terdiri dari lima tahapan dengan input berupa dataset, dataset dibagi menjadi dua bagian, satu bagian untuk data latih dan bagian lainnya untuk data uji.



**Gambar. 2.** Tahapan Penelitian

#### 3.2 Dataset

Dataset yang digunakan pada penelitian ini didapatkan dari kaggle dengan nama “10 Monkey Species” dan bersifat *open source* atau dapat digunakan secara terbuka oleh siapapun. Dataset yang digunakan ini merupakan versi kedua dari dataset yang direvisi pada tahun 2019. Atribut yang digunakan pada dataset ini berupa gambar berwarna RGB yang terdiri dari 3 dimensi dengan 10 kelas sesuai dengan jumlah spesies yang diklasifikasikan.

#### 3.3 CNN Classifier

Model ini bertujuan untuk membuat matriks gambar input menjadi matriks 1 dimensi. Pada awalnya dilakukan banyak dimension reduction untuk memenuhi kebutuhan, yaitu dengan menggunakan konvolusi. Penggunaan konvolusi pasti akan mengurangi dimensi panjang dan tinggi selama kernel yang dipakai bukan (1,1). Pada penelitian ini juga digunakan layer pooling untuk mengurangi dimensi input.


Pada layer terakhir sangat penting untuk model untuk hanya memiliki matriks 1 dimensi yang memiliki 10 anggota. 1 dimensi karena CNN hanya menerima array 1 dimensi dan 10 anggota karena pada dataset ini kita memiliki 10 kelas spesies monyet.

## 4 Hasil dan Pembahasan

### 4.1 Dataset

Penelitian ini dilakukan menggunakan dataset sebanyak 1370 data entri yang terbagi menjadi dua bagian yaitu 1098 data latih dan 272 untuk validasi dengan 10 kelas berbeda. Proporsinya adalah 20% data latih dan 80% data validasi.

Label	Latin Name	Common Name	Train Images	Validation Images
0	n0 alouatta palliata	mantled_howler	131	26
1	n1 erythrocebus patas	patas_monkey	139	28
2	n2 cacajao calvus	bald_uakari	137	27
3	n3 macaca fuscata	japanese_macaque	152	30
4	n4 cebuella pygmaea	pygmy_marmoset	131	26
5	n5 cebus capucinus	white_headed_capuchin	141	28
6	n6 mico argentatus	silvery_marmoset	132	26
7	n7 saimiri sciureus	common_squirrel_monkey	142	28
8	n8 aotus nigriceps	black_headed_night_monkey	133	27
9	n9 trachypitecus johnii	nilgiri_langur	132	26

**Gambar. 3.** Tahapan Penelitian

### 4.2 Preprocessing

pada tahap ini kita melakukan augmentasi gambar pada gambar latih dan validasi. dilakukan augmentasi gambar pada dataset karena menurut penelitian yang dilakukan oleh [3] bahwa akurasi tidak jatuh terlalu jauh saat gambar yang ingin di klasifikasi 'jelek' atau out-of-focus, not-centered, diperbesar/diperkecil, atau resolusi rendah. penelitian disini adalah perbandingan nilai akurasi terhadap data latih yang dilakukan augmentasi gambar dan yang tidak. saat gambar input masih 'bagus' nilai akurasi sama persis, tapi semua itu berubah saat gambar input 'memburuk', nilai akurasi pada classifier yang dilakukan augmentasi terhadap gambar latih lebih tinggi daripada yang tidak dilakukan augmentasi gambar.



**Gambar. 4.** Dari kiri ke kanan: gambar asli, rotasi, geser, diperkecil, flip.

### 4.3 CNN

Tahap ini menggunakan input gambar yang di limitasi menjadi 220X220 pixel, RGB. Dilakukan dimension reduction pada beberapa hidden layer. pada layer polling digunakan max pooling karena dari percobaan ini, akurasi lebih tinggi saat menggunakan max pooling. Peneliti juga lakukan dropout pada model CNN untuk: 1) mengurangi waktu latih dari model. 2) membuat model yang lebih general[2].

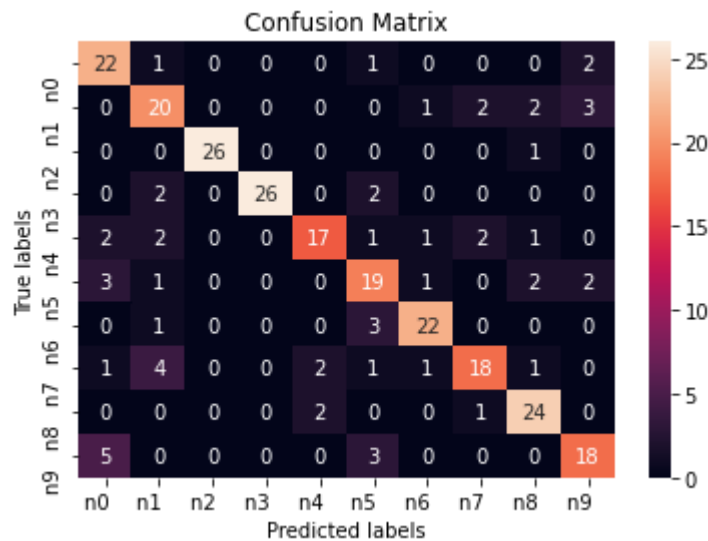
```

Model: "sequential"
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)             (None, 72, 72, 32)         2432
activation (Activation)     (None, 72, 72, 32)         0
batch_normalization (Batch Normalization) (None, 72, 72, 32)         128
conv2d_1 (Conv2D)           (None, 34, 34, 64)         51264
activation_1 (Activation)   (None, 34, 34, 64)         0
batch_normalization_1 (Batch Normalization) (None, 34, 34, 64)         256
conv2d_2 (Conv2D)           (None, 16, 16, 64)         36928
activation_2 (Activation)   (None, 16, 16, 64)         0
batch_normalization_2 (Batch Normalization) (None, 16, 16, 64)         256
conv2d_3 (Conv2D)           (None, 7, 7, 128)          73856
activation_3 (Activation)   (None, 7, 7, 128)         0
batch_normalization_3 (Batch Normalization) (None, 7, 7, 128)         512
conv2d_4 (Conv2D)           (None, 3, 3, 128)          147584
activation_4 (Activation)   (None, 3, 3, 128)         0
dropout (Dropout)           (None, 3, 3, 128)         0
conv2d_5 (Conv2D)           (None, 3, 3, 512)          66048
max_pooling2d (MaxPooling2D) (None, 1, 1, 512)         0
conv2d_6 (Conv2D)           (None, 1, 1, 10)           5130
global_average_pooling2d (Global Average Pooling) (None, 10)                 0
dropout_1 (Dropout)         (None, 10)                 0
activation_5 (Activation)   (None, 10)                 0
-----
Total params: 384,394
Trainable params: 383,818
Non-trainable params: 576
  
```

**Gambar. 5.** *Our CNN Model*

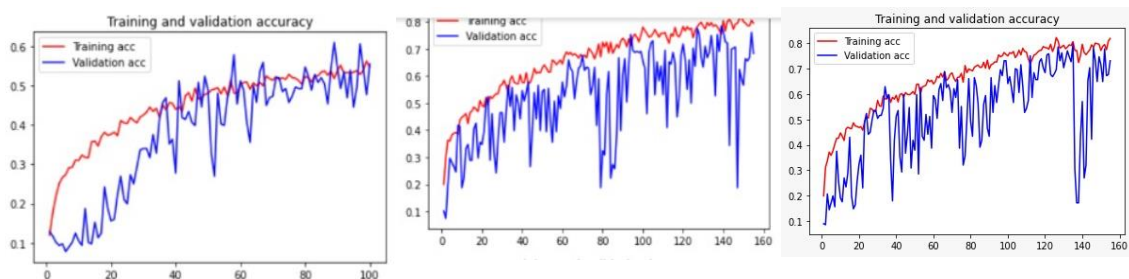
Gambar 4 adalah summary dari model yang telah dibuat. Banyak sekali fungsi konvolusi yang digunakan pada model ini dan sengaja dibatasi karena keterbatasan waktu dan sumber daya. Peneliti juga menaikkan nilai filter secara gradual pada layer ini.

Bisa dilihat bahwa hanya dengan menggunakan layer konvolusi matriks tidak akan bisa mencapai 1 dimensi. Tapi sebelum itu matriks harus memiliki nilai perkalian total sebanyak kelas yaitu 10. Peneliti melakukan ini dengan cara melakukan konvolusi dengan kernel (1,1) dan filter 10, dimana hasilnya dari (1,1,512) menjadi (1,1,10). setelah itu dilakukan Global average pooling untuk mendapat matriks sesuai dengan yang kita butuhkan[8]. Setelah matriks sesuai dengan kebutuhan, maka bisa dilakukan fungsi aktivasi softmax untuk mendapat akurasi terbaik[1].



**Gambar. 6.** Our CNN Classifier's Confusion Matrix

Melihat daripada akurasi yang kurang stabil karena adanya kesalahan pada saat pengklasifikasian, namun didapatkan hasil yang cukup baik secara keseluruhan dimana dominan dari data validasi atau data *testing* sudah diprediksi tepat pada sasaranannya sebagaimana yang tergambar pada confusion matrix pada Gambar 5.



**Gambar. 7.** Grafik Sejarah Pelatihan Model

Gambar 6 menunjukkan sejarah dari beberapa percobaan model yang peneliti buat. Sumbu x pada grafik menunjukkan *epoch*-nya sedangkan sumbu y menunjukkan nilai akurasi. Garis biru menunjukkan akurasi data validasi dan garis merah menunjukkan akurasi data latih. Bisa kita lihat pada grafik, Garis yang menunjukkan data latih lebih stabil daripada garis data validasi. Hal ini terjadi karena model di fit pada data latih, data latih juga akan hampir selalu memiliki akurasi yang lebih tinggi. Hal yang muncul di ketiga percobaan adalah bahwa trend akurasi masih menaik, sehingga sangat mungkin untuk nilai akurasi final akan lebih tinggi jika *epoch* kita tambah. Banyaknya *epoch* tidak menjamin akurasi klasifikasi akan menjadi lebih tinggi, tapi dengan melihat trend pada graph sejarah latih maka kita bisa mengambil deduksi berbobot tentang apa yang akan terjadi jika kita tambahkan nilai *epoch*.

Penelitian terkait pengklasifikasian 10 spesies monyet pernah dilakukan sebelumnya. Penelitian pertama juga menggunakan CNN yang mendapatkan nilai akurasi validasi sampai 95%[5]. Perbedaan hasil antara penelitian ini dan sebelumnya terjadi karena keterbatasan sumber daya. Pertama pada bagian model CNN, bisa dilihat pada gambar 4, *trainable params* kami hanya 380 ribu, setiap *epoch* pada penelitian ini menghabiskan waktu sekitar 65 detik, sehingga memakan waktu sekitar 2 jam 40 menit untuk setiap percobaan, sedangkan penelitian pertama memiliki *trainable params* lebih dari 9 juta. Kedua pada nilai *epoch*, penelitian ini hanya 155 *epoch* dan sudah mencapai 79% sedangkan penelitian sebelumnya menggunakan 200 *epoch*.

## 5 Penutup

### 5.1 Kesimpulan

Salah satu cara untuk melakukan klasifikasi terhadap dataset gambar yaitu dengan menggunakan CNN. CNN sendiri merupakan salah satu contoh dari pengaplikasian deep learning dimana terdapat input layer, hidden layer dan output layer pada arsitekturnya.

Pada penelitian ini dataset yang digunakan adalah dataset gambar dari 10 spesies monyet yang ada di bumi, terdapat 1370 entri data yang dibagi dengan proporsi 80% untuk data latih dan 20% untuk validasi, dilakukan generasi pada gambar sehingga menciptakan sudut pandang baru. Setelah dilakukan pengklasifikasian menggunakan CNN didapatkan hasil 78% akurasi dari 1370 entri data.

### 5.2 Saran

Dari hasil penelitian dan simpulan yang didapat, maka disarankan untuk melakukan pengklasifikasian selanjutnya dapat dilakukan penambahan data gambar untuk memperkaya pembelajaran mesin terhadap dataset tersebut.

Dari kesimpulan Gambar 6 juga bisa kita tangkap bahwa pada model ini kita masih bisa menaikkan nilai epoch untuk mendapatkan akurasi yang lebih tinggi. Penelitian ini menggunakan 155 epoch karena keterbatasan sumber daya dan waktu. Tapi melihat trend yang masih naik maka hampir pasti bahwa nilai akurasi bisa naik sampai lebih dari 80% jika nilai epoch kita tambahkan.

## Referensi

- [1] Agarp, A. F. (2017). An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification. CoRR, <https://dblp.org/rec/journals/corr/abs-1712-03541.bib>
- [2] Andono, P. N., Sutojo, T., & Mulijono. (2017). Pengolahan Citra Digital. Yogyakarta: Penerbit ANDI.
- [3] Inoue, H. (2019). Multi-Sample Dropout for Accelerated Training and Better Generalization. CoRR, abs/1905.09788.
- [4] Kannoja, S. P., & Jaiswal, G. (2018). Effects of Varying Resolution on Performance of CNN based Image Classification An Experimental Study. International Journal of Computer Sciences and Engineering, 6(9), 451–456. <https://doi.org/10.26438/ijcse/v6i9.451456>.
- [5] Maduwuba, E., Dharanikota R. K., & Kamaljeet S. M. (2020). Performing Image Classification for 10 Different Monkey Species using CNN. ECE 9309B: Machine Learning: From Theory to Applications. Western University of Ontario.
- [6] Monyet. (2021, Februari 28). Di Wikipedia, Ensiklopedia Bebas. Diakses pada 12:05, Februari 28, 2021, dari <https://id.wikipedia.org/w/index.php?title=Monyet&oldid=18048333>
- [7] Mubarak, H., Sylviana M., & Mayanda M. S. "Penerapan Algoritma K-Nearest Neighbor Untuk Klasifikasi Tingkat Kematangan Buah Tomat Berdasarkan Fitur Warna". SENAMIKA 2.1 (2021).
- [8] Razzak, M. I., Naz, S., & Zaib, A. (2017). Deep Learning for Medical Image Processing: Overview, Challenges and the Future. Lecture Notes in Computational Vision and Biomechanics, 323–350. [https://doi.org/10.1007/978-3-319-65981-7\\_12](https://doi.org/10.1007/978-3-319-65981-7_12).
- [9] Rossi, M. (2019). Synthetic image generator for defocusing and astigmatic PIV/PTV. Measurement Science and Technology, 31(1), 017003. <https://doi.org/10.1088/1361-6501/ab42bb>
- [10] Sun, Y., Xue, B., Zhang, M., Yen, G. G., & Lv, J. (2020). Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification. IEEE Transactions on Cybernetics, 50(9), 3840–3854. <https://doi.org/10.1109/tycb.2020.2983860>.
- [11] V. Christlein, L. Spranger, M. Seuret, A. Nicolaou, P. Král and A. Maier, "Deep Generalized Max Pooling," 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019, pp. 1090-1096, doi: 10.1109/ICDAR.2019.00177.
- [12] White, R. (2017). CHEC: a Compact High Energy Camera for the Cherenkov Telescope Array. Journal of Instrumentation, 12(12), C12059. <https://doi.org/10.1088/1748-0221/12/12/c12059>

- [13] Yohannes, Y., Sari, Y., & Feristyani, I. (2019). Klasifikasi Wajah Hewan Mamalia Tampak Depan Menggunakan k-Nearest Neighbor Dengan Ekstraksi Fitur HOG. *Jurnal Teknik Informatika Dan Sistem Informasi*, 5(1). <https://doi.org/10.28932/jutisi.v5i1.1584>.