

Klasifikasi Jenis Lebah Madu Menggunakan Metode Convolutional Neural Network

Syifa Fhadhillah Chairunissa

Program Studi Informatika, Fakultas Ilmu Komputer dan Rekayasa, Universitas MDP, Indonesia
e-mail: syifachaini99@mhs.mdp.ac.id

Abstrak

Penelitian ini bertujuan untuk mengembangkan suatu sistem klasifikasi jenis lebah menggunakan metode *Convolutional Neural Network* (CNN). CNN adalah salah satu metode pengolahan citra yang dapat mempelajari fitur-fitur pada gambar secara otomatis, sehingga sangat efektif untuk digunakan dalam tugas klasifikasi jenis lebah madu. Data yang digunakan dalam penelitian ini adalah gambar lebah dari tiga jenis yaitu *Carnolian Honey Bee*, *Italian Honey Bee*, dan *Russian Honey Bee*. Pengolahan data dilakukan dengan melakukan *preprocessing* pada gambar seperti *resizing* dan normalisasi intensitas piksel. Selanjutnya, model CNN dilatih pada dataset menggunakan *optimizer* Adam. Akurasi model diukur menggunakan metode ini dengan hasil akurasi sebesar 97,8%. Dalam penelitian ini juga dilakukan analisis terhadap fitur-fitur yang dipelajari oleh model CNN dalam mengklasifikasikan jenis lebah. Hasil penelitian menunjukkan bahwa metode CNN dapat digunakan secara efektif untuk klasifikasi jenis lebah dengan akurasi yang tinggi, serta mampu mempelajari fitur-fitur penting pada gambar lebah yang digunakan sebagai dataset.

Kata kunci: Pengolahan Citra Digital, Lebah Madu, *Convolutional Neural Network*, Klasifikasi Jenis Lebah Madu

Abstract

This research aims to develop a classification system for honey bee species using the Convolutional Neural Network (CNN) method. CNN is one of the image processing methods that can automatically learn features in images, making it effective for classifying honey bee species. The data used in this study consists of images of three honey bee species, namely Carnolian Honey Bee, Italian Honey Bee, and Russian Honey Bee. Data processing was carried out by pre-processing the images, such as resizing and normalizing pixel intensity. Furthermore, the CNN model was trained on the dataset using the Adam optimizer. The accuracy of the model was measured using this method with an accuracy of 97,8%. In this study, an analysis of the features learned by the CNN model in classifying honey bee species was also conducted. The results showed that the CNN method can be effectively used for the classification of honey bee species with high accuracy and can learn important features in the honey bee images used as the dataset.

Keywords: Digital Image Processing, Honey Bees, Convolutional Neural Network, Honey Bee Species Classification

1. Pendahuluan

Lebah madu (*Apis mellifera*) merupakan salah satu organisme yang paling tersebar luas di dunia. Menurut Crane (2013), manusia telah melakukan penyebaran lebah madu ini dari daerah asalnya di Eropa, Timur Tengah, dan Afrika ke setiap benua kecuali Antartika. Lebah ini banyak dibudidayakan karena produksinya tinggi dan daya adaptasinya tinggi. Selain itu, spesies ini dapat dibuat galur baru di daerah yang berbeda dengan habitat aslinya. Lebah madu cenderung tidak agresif dan kurang suka bermigrasi ke daerah beriklim dingin, atau berlerefansi tinggi.

Menurut Widiastuti dan Adalina (2008), Lebah madu ini tergolong sebagai lebah hutan yang dibudidayakan hampir diseluruh bagian Indonesia. Lebah ini biasanya dibudidayakan pada petani golongan menengah keatas dikarenakan perlu disiapkan truk untuk mengangkut perlengkapan dan fasilitas pendukung lainnya. *Apis mellifera* merupakan lebah unggul yang populer dan paling banyak ditenakan di Indonesia karena memiliki produktivitas madu yang tinggi yaitu 25-35 kg/koloni setiap tahunnya serta sifatnya yang jinak, tidak mudah kabur dan mudah beradaptasi terhadap lingkungan baru. Spesies lebah ini tergolong rakus dengan nektar (makanan). Karena itu, tidak heran budidaya ini dilakukan dengan cara menggolongkan (dipindah dari satu tempat ketempat yang lain) ditempat yang banyak sumber pakannya. Lebah madu (*Apis mellifera*) memiliki subspecies yang banyak. Tiap subspecies dari jenis lebah ini juga memiliki tingkat keunggulan atau kualitas madu yang berbeda. Manfaat dari penggunaan metode CNN untuk penelitian ini adalah mempermudah para peternak ataupun masyarakat awam yang tertarik dengan lebah madu dalam melakukan pengklasifikasian jenis lebah ini dengan mendapatkan hasil kualitas madu yang terbaik.

Adapun penelitian yang dilakukan oleh Ariyadi (2020) membahas mengenai klasifikasi spesies lebah berbasis data citra dengan metode *Support Vector Machine* dengan mempelajari fitur-fitur dari 70% data citra yang ada dan disertai dengan label. Untuk memaksimalkan proses komputasi, fitur data telah diolah dengan Histogram of Oriented Gradient (HOG) dan Principal Component Analysis (PCA). Hasil penelitian menunjukkan bahwa model yang telah dibangun dengan metode SVM menghasilkan tingkat akurasi sebesar 74%.

2. Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah sebagai berikut.

1.1. Identifikasi Masalah

Pada tahap ini dilakukan pengumpulan masalah yang ingin diteliti serta menentukan batasan masalah yang ingin dilaksanakan.

1.2. Studi Literatur

Tahap ini ditujukan untuk mengumpulkan referensi yang relevan dengan fokus literatur yang dipersempit, sehingga dapat memberikan kontribusi yang baru dan memiliki dasar informasi yang kuat.

1.3. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data untuk dilatih dan diuji.

1.4. Pemilihan Citra

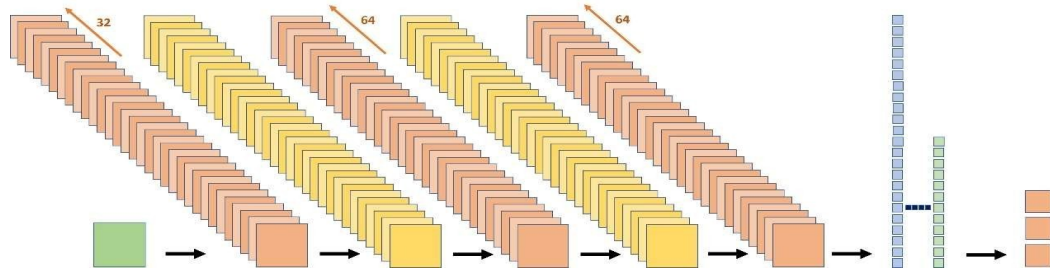
Pada tahapan ini, data akan diseleksi untuk diambil foto dengan kualitas gambar yang baik.

1.5. Pre-Processing

Pada tahap ini, peneliti akan melakukan tahapan seperti pengelompokan data train dan data test, resizing citra, dan labeling citra

1.6. Menyusun CNN

Pada tahap ini akan dilakukan penyusunan sebuah model yang akan digunakan dalam proses pelatihan data.



Gambar 1 Arsitektur CNN

1.7. Pelatihan Citra

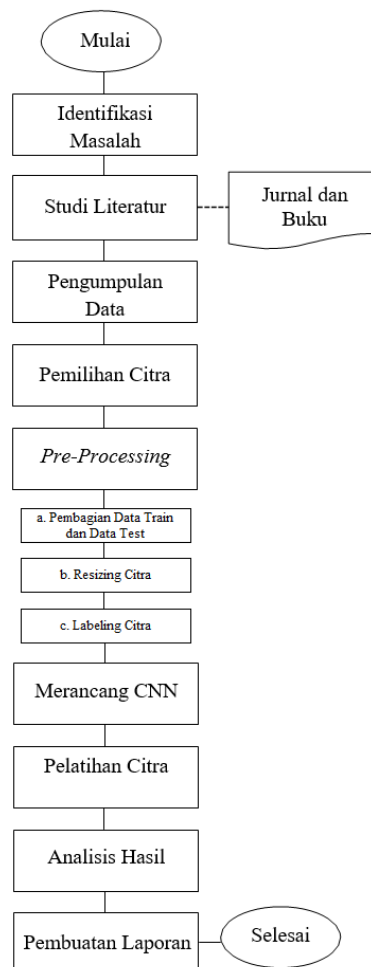
Pada tahap ini, peneliti akan melakukan tahapan pembangunan sistem agar mampu untuk mengklasifikasikan jenis lebah madu (*Apis Mellifera*) untuk menghasilkan *output* yang dapat digunakan pada tahap pengujian.

1.8. Analisis Hasil

Pada tahap ini dilakukan analisis berdasarkan hasil pengujian program yang telah dilakukan sebelumnya.

1.9. Pembuatan Laporan

Pada tahap ini, penulis akan membuat laporan dari hasil penelitian yang telah dilakukan.



Gambar 2 Metodologi Penelitian

3. Hasil dan Pembahasan

Pada bagian ini, disajikan hasil penelitian yang diperoleh melalui serangkaian implementasi pemrograman. Semua metode dari program yang telah dirancang, diimplementasikan menggunakan Google Colab.

3.1.1 *Resize Citra*

Resize citra merupakan proses untuk mengubah ukuran citra menjadi lebih kecil jika ukuran dimensi citra terlalu besar. Citra akan *dilakukan resize* menjadi 60x60 pixel. Kode program *resize citra* dapat dilihat pada gambar 3.

```
for i in range(0, 1500):
    data[i] = np.array(Image.fromarray(data[i]).resize((60, 60)))
```

Gambar 3 Kode *Resize Citra*

3.1.2 *Perancangan Model CNN*

Setelah dilakukannya *resize citra*, selanjutnya dilakukanlah perancangan model CNN dengan menggunakan Google Colab menggunakan konvolusi 2D dengan jumlah *dense layers* sebanyak 64 *layer*. *Output* dari klasifikasi ini dibagi menjadi 3 kelas yaitu *Carnolian Honey Bee*, *Italian Honey Bee*, dan *Russian Honey Bee*. Kode pemrogramannya dapat dilihat pada Gambar 4.

```
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(60, 60, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))

model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(3)) #3 kelas
```

Gambar 4 Perancangan model CNN

3.1.3 *Output Pemrograman dan Hasil*

Jika perancangan model CNN sudah selesai, dilakukanlah pembuatan baris kode untuk menampilkan output dan akurasi dari jenis lebah yang telah diklasifikasikan. Optimizer yang digunakan adalah Adam dan DSG sebagai pembandingan akurasi mana yang lebih tinggi. Dilakukan juga pembagian data train dan data latih dengan perbandingan 70:30. Kode program yang digunakan dalam penelitian ini dapat dilihat di Gambar 5.

```
model.compile(optimizer='Adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

history = model.fit(X_train, y_train, epochs=4, validation_data=(X_test, y_test))
```

Gambar 5 Perancangan model CNN (Output)

Adapun akurasi yang didapatkan dengan menggunakan metode ini adalah 97,8% seperti yang ditunjukkan pada Gambar 6.

```
Epoch 1/4
33/33 [=====] - 7s 164ms/step - loss: 14.8680 - accuracy: 0.6210 - val_loss: 0.2218 - val_accuracy: 0.9311
Epoch 2/4
33/33 [=====] - 5s 153ms/step - loss: 0.1746 - accuracy: 0.9410 - val_loss: 0.2030 - val_accuracy: 0.9289
Epoch 3/4
33/33 [=====] - 4s 130ms/step - loss: 0.1284 - accuracy: 0.9571 - val_loss: 0.0752 - val_accuracy: 0.9689
Epoch 4/4
33/33 [=====] - 6s 175ms/step - loss: 0.0560 - accuracy: 0.9781 - val_loss: 0.0612 - val_accuracy: 0.9844
```

Gambar 6 Hasil Akurasi

Disini saya melakukan empat kali pengujian terhadap dataset yang saya miliki untuk mendapatkan kesimpulan akurasi mana yang terbaik. Pertama saya menggunakan optimizer Adam dengan total data test sebesar 30% dan dilakukannya dua kali training dengan empat kali pengujian. Hasilnya menunjukkan tingkat akurasi sebesar 98%. Sedangkan untuk data test sebesar 70% dengan dilakukannya dua kali training dengan empat kali pengujian, hasilnya menunjukkan 76%. Lalu untuk percobaan kedua, saya menambahkan agar dilakukannya empat kali training dengan empat kali pengujian pada data tersebut. Hasil menunjukkan bahwa untuk data test 30% mendapatkan tingkat akurasi sebesar 97% dan untuk data test 70% mendapatkan akurasi sebesar 96%. Selanjutnya, saya menggunakan optimizer SGD sebagai perbandingan mana optimizer yang dapat memberikan tingkat akurasi yang lebih baik. Namun sayangnya, untuk optimizer SGD, memiliki tingkat akurasi yang sangat rendah. Pada data test 30% dengan total train sebanyak dua dan empat kali dengan empat kali pengujian, didapatkan akurasi sebesar 32%. Sedangkan untuk total data test 70% dengan total train sebanyak dua dan empat kali, didapatkan akurasi sebesar 32%.

Optimizer	Epoch	Train Accuracy	Test Accuracy	Average Accuracy I	Average Accuracy II
Adam 30% (2x Training)	1	0.7838	0.9600	0.9678	0.9820
	2	0.9352	0.9756		
	1	0.8190	0.9689	0.9789	
	2	0.9638	0.9889		
	1	0.8762	0.9889	0.9900	
	2	0.9800	0.9911		
	1	0.9924	0.9867	0.9878	
	2	0.9857	0.9889		
	1	0.9448	0.9822	0.9856	
	2	0.9905	0.9889		
Adam 30% (4x Training)	1	0.8181	0.8533	0.9356	0.9726
	2	0.9552	0.9444		
	3	0.9933	0.9667		
	4	0.9467	0.9778		
	1	0.9524	0.9889	0.9761	
	2	0.9952	0.9889		
	3	0.9810	0.9733		
	4	0.9810	0.9533		
	1	0.9657	0.9800	0.9795	
	2	0.9848	0.9867		
	3	0.9933	0.9667		
	4	0.9914	0.9844		
	1	0.9638	0.9844	0.9906	
	2	0.9952	0.9844		
	3	0.9952	0.9956		
	4	1.0000	0.9978		
	1	0.9724	0.9511	0.9811	
	2	0.9857	0.9933		
	3	0.9971	0.9889		
	4	1.0000	0.9911		
Adam 70% (2x Training)	1	0.3911	0.6400	0.7724	0.7653
	2	0.7044	0.9048		
	1	0.4222	0.6905	0.6881	
	2	0.7467	0.6857		
	1	0.6867	0.6829	0.6572	
	2	0.7578	0.6314		
	1	0.8222	0.8124	0.8900	
	2	0.8889	0.9676		
	1	0.7489	0.9381	0.8191	
	2	0.9644	0.7000		

Adam 70% (4x Training)	1	0.7467	0.9571	0.9243	0.9621
	2	0.9733	0.9514		
	3	0.9800	0.9238		
	4	0.9067	0.8648		
	1	0.9289	0.9781	0.9679	
	2	0.9800	0.9790		
	3	0.9911	0.9448		
	4	0.9356	0.9695		
	1	0.8889	0.9752	0.9764	
	2	0.9867	0.9724		
	3	0.9889	0.9790		
	4	0.9956	0.9790		
	1	0.9311	0.9733	0.9686	
	2	0.9844	0.9771		
	3	0.9667	0.9400		
	4	0.9889	0.9838		
1	0.9400	0.9581	0.9733		
2	0.9800	0.9790			
3	0.9867	0.9762			
4	0.9911	0.9800			

Tabel 1 Akurasi Adam Optimizer

Optimizer	Epoch	Train Accuracy	Test Accuracy	Average Accuracy I	Average Accuracy II
SGD 30% (2x Training)	1	0.3381	0.3289	0.3289	0.3289
	2	0.3352	0.3289		
	1	0.3352	0.3289	0.3289	
	2	0.3352	0.3289		
	1	0.3352	0.3289	0.3289	
	2	0.3352	0.3289		
	1	0.3352	0.3289	0.3289	
	2	0.3352	0.3289		
SGD 30% (4x Training)	1	0.3390	0.3289	0.3289	0.3289
	2	0.3352	0.3289		
	3	0.3352	0.3289		
	4	0.3352	0.3289		
	1	0.3352	0.3289	0.3289	
	2	0.3352	0.3289		
	3	0.3352	0.3289		
	4	0.3352	0.3289		
	1	0.3352	0.3289	0.3289	
	2	0.3352	0.3289		
	3	0.3352	0.3289		
	4	0.3352	0.3289		
	1	0.3352	0.3289	0.3289	
	2	0.3352	0.3289		
	3	0.3352	0.3289		
	4	0.3352	0.3289		
SGD 70% (2x Training)	1	0.3244	0.3324	0.3267	0.3269
	2	0.3667	0.3210	0.3262	
	1	0.3311	0.3238		
	2	0.3000	0.3286	0.3310	
	1	0.3156	0.3305		
	2	0.3356	0.3314	0.3248	
	1	0.3089	0.3210		
	2	0.3622	0.3286	0.3257	
1	0.3422	0.3295			
2	0.3356	0.3219			

SGD 70% (4x Training)	1	0.3356	0.3219	0.3230	0.3219
	2	0.3200	0.3129		
	3	0.3333	0.3267		
	4	0.3622	0.3305		
	1	0.3622	0.3219	0.3214	
	2	0.3489	0.3133		
	3	0.3511	0.3286		
	4	0.2978	0.3219		
	1	0.3644	0.3114	0.3212	
	2	0.3356	0.3295		
	3	0.3444	0.3219		
	4	0.3622	0.3219		
	1	0.3622	0.3219	0.3219	
	2	0.3622	0.3219		
	3	0.3622	0.3219		
	4	0.3622	0.3219		
1	0.3622	0.3219	0.3219		
2	0.3622	0.3219			
3	0.3622	0.3219			
4	0.3622	0.3219			

Tabel 2 Akurasi SGD Optimizer

4. Kesimpulan

1. Berdasarkan hasil penelitian, dapat disimpulkan bahwa metode *Convolutional Neural Network* (CNN) dapat digunakan secara efektif untuk klasifikasi jenis lebah madu. Dalam penelitian ini, dilakukan preprocessing pada gambar seperti *resizing* dan normalisasi intensitas piksel sebelum dilatih dengan model CNN menggunakan *optimizer* Adam. Hasil evaluasi menunjukkan bahwa model CNN dengan *optimizer* Adam memiliki akurasi sebesar 98% dalam mengklasifikasikan gambar lebah dari tiga jenis yang digunakan sebagai dataset.
2. Analisis fitur-fitur yang dipelajari oleh model CNN menunjukkan bahwa model CNN mampu mempelajari fitur-fitur penting pada gambar lebah untuk melakukan klasifikasi jenis lebah madu.

5. Saran

1. Berdasarkan hasil penelitian, disarankan untuk memperluas dataset dengan menambahkan gambar lebah dari jenis lain agar model CNN dapat memiliki performa yang lebih baik dalam mengklasifikasikan jenis lebah madu.
2. Disarankan bahwa dapat dilakukannya eksperimen dengan metode *preprocessing* yang berbeda untuk melihat pengaruhnya terhadap akurasi model.

6. Daftar Pustaka

- [1] Ariyadi. (2020). Klasifikasi spesies lebah berbasis data citra dengan metode Support Vector Machine. Diambil dari <https://stp-mataram.e-journal.id/JIP/article/view/204>
- [2] Kevin dkk. (2019). Klasifikasi citra genus panthera menggunakan metode Convolutional Neural Network (CNN). Diambil dari <https://ejournal.gunadarma.ac.id/index.php/infokom/article/view/2364>
- [3] Kusmaningrum, Tutut Furi. (2019). Implementasi Convolutional Neural Network (CNN) untuk klasifikasi jamur konsumsi menggunakan Keras. Diambil dari <https://dspace.uui.ac.id/handle/123456789/7781>

- [4] Sandi, Rahayu Kia. (2018). Convolutional Neural Network untuk klasifikasi gambar pada tanaman anggrek. Diambil dari <https://dspace.uui.ac.id/handle/123456789/7449>
- [5] Dhika, Harry dkk. (2020). Model prediksi jenis hewan dengan metode Convolutional Neural Network. Diambil dari <https://publikasi.mercubuana.ac.id/index.php/format/article/view/8462>
- [6] Setiawan, Nanang Agus. (2018). Identifikasi citra jenis lebah madu dengan metode klasifikasi Monkowski Distance. Diambil dari <http://simki.unpkediri.ac.id/detail/13.1.03.02.0177>
- [7] Rodriguez dkk. (2018). Deteksi lebah madu dan estimasi pose menggunakan Convolutional Neural Network. Diambil dari https://www.researchgate.net/publication/336278306_Honeybee_Detection_and_Pose_Estimation_using_Convolutional_Neural_Networks
- [8] Putra, Ilham Rizaldy Widy. (2018). Deteksi jenis buahbuahan menggunakan metode Convolutional Neural Network. Diambil dari <http://repository.dinamika.ac.id/id/eprint/5392/1/17410200032-2020-UNIVERSITASDINAMIKA.pdf>
- [9] Adiyat, Roihan. (2015). Klasifikasi sayap lebah Apis cerana dan Apis koschevnikovi menggunakan Conditional Inference Tree. Diambil dari <https://repository.ipb.ac.id/handle/123456789/76746>
- [10] Alfahmi, Gusti dan Desti Rimirasih. (2019). Klasifikasi citra genus panthera menggunakan metode Convolutional Neural Network (CNN). Diambil dari <https://ejournal.gunadarma.ac.id/index.php/infokom/article/view/2364>
-