Progresif: Jurnal Ilmiah Komputer
Jl. Ahmad Yani, K.M. 33,5 - Kampus STMIK Banjarbaru
Loktabat – Banjarbaru (Tlp. 0511 4782881), e-mail: puslit.stmikbjb@gmail.com
e-ISSN: 2685-0877
p-ISSN: 0216-3284

Klasifikasi Rumput Liar Menggunakan Algoritme *Deep Learning* Dengan *Dense Convolutional Neural Network*

Yusril Nurhadi AS[1*], Dadang Iskandar Mulyana[2], Yuma Akbar[3]
Program Studi Teknik Informatika, Sekolah Tnggi Ilmu Komputer Cipta Karya Informatika, Jakarta, Indonesia
*e-mail *Corresponding Author:* yusrilnurhadi63@gmail.com

Abstract

Weed control research using robots increases productivity in agriculture. Most of the work focused on developing robotics for farmland, ignoring the weed management issues facing pasture farmers. The biggest obstacle to the expansion of robotic weed control is the robust classification of weed species in their natural environment. This work contributes to the method of classifying weed species using Deep Learning with Dense Convolutional Neural Network. The wild grass images used are Chinee apple, Snake weed, Lantana, Prickly acacia, Siam weed, Parthenium, Rubber vine and Parkinsonia. This image recognition implementation is done by using Resnet50 on Tensorflow at Google Collaboratory. The dataset used in the test is the DeepWeeds dataset which consists of 17,509 images labeled with 10,505 training data and 3,502 test data used to produce evaluation values with 78% precision, 78% recall, 78% f1-score, 77.73% accuracy and loss. 0.6676.

Kata kunci: *ResNet50; Convolutional Neural Network; Image Classification.*

Abstrak

Penelitian kontrol gulma menggunakan robot meningkatkan produktivitas di bidang agrikultur. Sebagian besar pekerjaan fokus pada pengembangan robotika untuk lahan pertanian, mengabaikan masalah pengelolaan gulma yang dihadapi peternak padang rumput. Kendala terbesar untuk peluasan pengendalian gulma dengan robot adalah klasifikasi kuat *species* gulma di lingkungan alami mereka. Artikel ini berkontribusi pada metode pengklasifikasian *species* gulma menggunakan *Deep Learning* dengan *Dense Convolutional Neural Network*. Citra tanaman rumput liar yang digunakan adalah *Chinee apple, Snake weed, Lantana, Prickly acacia, Siam weed, Parthenium, Rubber vine* dan *Parkinsonia*. Implementasi pengenalan citra ini dilakukan dengan memanfaatkan *Resnet50* pada *Tensorflow* di *Google Collaboratory*. Dataset yang digunakan dalam pengujian adalah dataset *DeepWeeds* yang terdiri dari 17.509 gambar berlabel sebanyak 10.505 data training dan 3.502 data test yang digunakan menghasilkan nilai evaluasi dengan nilai *precision* 78%, *recall* 78%, f1-score 78%, akurasi 77,73% dan *loss* 0.6676.
Kata kunci: *ResNet50; Convolutional Neural Network; Image Classification.*

1. Pendahuluan

Agrikultur (*Agriculuture*) adalah sebuah seni dan ilmu mengolah tanah, menanam tanaman dan memelihara ternak. Ini mencakup penyediaan produk tumbuhan dan hewan untuk digunakan masyarakat dan distribusinya ke pasar. *Agriculuture* menyediakan sebagian besar makanan dan kain dunia. Kapas, wol dan kulit semuanya adalah produk *agriculture*. *Agriculture* juga menyediakan kayu untuk konstruksi dan produk kertas. Produk-produk ini, serta metode *agriculture* yang digunakan bisa bervariasi dari satu bagian dunia ke bagian dunia lainnya [1].

Pertanian Republik Indonesia lebih banyak *agriculture* yang menghasilkan produk-produk tanaman seperti karet, sawit, kakao dan kopi, namun di lain sektor ada udang yang menjadi produk unggulan nomor 1 [2]. Sedangkan produk hewani yang berasal dari sapi yang menghasilkan daging ataupun domba dengan wolnya bahkan belum bisa menempati 10 produk potensial Indonesia [3].

Berlokasi di Jonggol, Kabupaten Bogor, Jabar terdapat banyak masyarakat lokal yang telah menerapkan agriculture untuk menghasilkan produk dari sapi ataupun domba. Dalam kasus ini kami menganggap para peternak di Jonggol masih belum mengeluarkan potensi maksimal dari agriculture yang mereka terapkan karena terlihat sapi-sapi yang diternak di padang penggembalaan sana terlihat kurus dibandingkan dengan sapi yang diternak di kandang.

Dengan informasi yang kami dapatkan setelah melihat area sekitar padang penggembalaan, faktor kualitas rumput yang menjadi asupan utama nutrisi untuk sapi/domba di Jonggol tidak dapat mencukupi kebutuhan nutrisi sapi/domba yang digembalakan di sana. Karena kebanyakan peternak di Jonggol lebih memilih menggunakan rumput lapang sebagai hijauan pakan. Di sisi lain lahan penanaman rumput baik di tegalan maupun di lahan kurang produktif banyak tersedia di daerah Jonggol. Rumput lapang sangat beragam jenis dan jumlahnya, sehingga memperbesar peluang terkonsumsinya tanaman beracun yang dapat mengakibatkan keracunan dan penyakit pada ternak sapi dan domba. Sebagai contoh, daun keladi yang jika dikonsumsi lendirnya akan menghambat saluran pencernaan atas ternak ataupun bayam-bayaman jika jumlahnya tidak terukur kandungan oksalat yang tinggi dapat menyebabkan hipokalsemia berat [4].

Penelitian kontrol gulma menggunakan Teknologi Informasi akhir-akhir ini mengalami peningkatan dengan potensinya untuk meningkatkan produktivitas di bidang agrikultur [5-7]. Sebagian besar pekerjaan fokus pada pengembangan robotika untuk lahan pertanian [8-10], mengabaikan masalah pengelolaan gulma yang dihadapi peternak padang rumput. Kendala terbesar untuk peluasan pengendalian gulma dengan robot adalah klasifikasi kuat species gulma di lingkungan alami mereka.

Dengan berkembangnya teknologi dan informasi yang sangat cepat pada saat ini terdapat banyak pilihan model komputasi yang dapat digunakan untuk membantu mengatasi kendala terbesar dalam dunia pertanian, sebagai contohnya adalah penggunaan algoritme *Deep Learning* [11, 12] yang terinspirasi dari sistem biologis manusia untuk mengenali citra yang diamati dalam *Computer Vision*. Dari analisa tersebut, kami menduga bahwa bahwa teknologi *Machine Learning* dapat digunakan dalam pengklasifikasi gulma, yang diterapkan pada robot pengendali gulma untuk meningkatkan kualitas produk agriculture khususnya di daerah Jonggol.

Penelitian kami mendesain dan mensimulasikan model sistem pakar untuk mengklasifikasi Rumput Liar Menggunakan algoritme *Deep Learning* dengan *Dense Convolutional Neural Network*, dengan harapan dapat mengembangkan robotnya di masa mendatang.

2. Tinjauan Pustaka

Sudah ada beberapa penelitian yang telah dilakukan terkait topik identifikasi rumput liar/gulma seperti *Weed Growth Stage Estimator Using Deep Convolutional Neural Networks* oleh Nima Teimouri, Mads Dyrmann, Per Rydahl Nielsen, Solvejg Kopp Mathiassen, Gayle J. Somerville & Rasmus Nyholm Jørgensen [13] dengan menggunakan Convolutional Neural Network pada kinerja komputer dalam mengevaluasi 2516 gambar yang bervariasi dalam hal tanaman, jenis tanah, resolusi gambar dan kondisi cahaya menghasilkan model yang dapat mencapai akurasi maksimal 78% untuk mengidentifikasi Polygonum spp. dan akurasi minimal 46% untuk mengidentifikasi blackgrass. Selain itu, diperoleh tingkat akurasi rata-rata 70% dalam memperkirakan jumlah daun dan akurasi 96% saat menerima penyimpangan dua daun.

Selain penelitian yang terkait topik rumput liar, ada juga yang terkait dengan Convolutional Neural Network yang menggunakan ResNet50 seperti Pengenalan Jamur yang Dapat Dikonsumsi Menggunakan Metode Transfer Learning Pada Convolutional Neural Network oleh Elok Iedfitra Haksoro dan Abas Setiawan [14] dengan menggunakan 4 base model yaitu MobileNets, MobileNetV2, ResNet50 dan VGG19 menghasilkan akurasi lebih dari 86% dengan hasil terbaik ialah 92,19% dari base model MobileNetsV2.

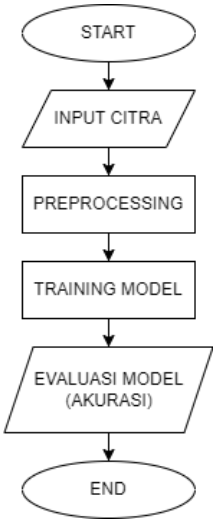
Fitriyah dan Maulana [15] mengembangkan model teknologi Deteksi Gulma Berdasarkan Warna HSV dan Fitur Bentuk Menggunakan Jaringan Syaraf Tiruan. Mereka membuat sistem yang dapat mendeteksi gulma secara otomatis di antara tanaman pada lahan pertanian riil. Sistem ini menggunakan gambar lahan pertanian riil dimana tanaman tampak utuh (daun dapat lebih dari satu) yang diambil menggunakan kamera dengan posisi vertikal menghadap ke bawah. Algortme yang dibuat menggunakan segmentasi berdasarkan warna hijau dalam ruang warna

HSV untuk mendeteksi daun, baik gulma maupun tanaman pada beragam pencahayaan. Sebanyak tiga fitur bentuk domain spasial digunakan untuk membedakan gulma dengan tanaman yang memiliki karakteristik bentuk daun yang berbeda. Fitur bentuk yang digunakan adalah Rectangularity, Edge-to-Center distances function, dan Distance Transform function. Klasifikasi gulma dan tanaman menggunakan metode Jaringan syaraf tiruan pada penelitian tersebut memiliki tingkat akurasi sebesar 95.46%.

Peneliti kami menggunakan *Convolutional Neural Network* dengan *ResNet50* untuk menguji/evaluasi model rumput liar. Citra tanaman rumput liar yang digunakan adalah *Chinee apple, Snake weed, Lantana, Prickly acacia, Siam weed, Parthenium, Rubber vine* dan *Parkinsonia*.

3. Metodologi

Penelitian ini dilakukan menggunakan Convolutional Neural Network dengan model *ResNet50* dengan data-data citra rumput liar bersumber dari dataset *DeepWeeds* yang diambil dari *Kaggle*. Selanjutnya citra dilakukan preprocessing dengan *me-resize* citra, data *augmentation* dan *feature extraction pre-trained* model *ResNet50*. Setelahnya data akan melalui tahap klasifikasi data *train* dengan menggunakan model *ResNet50* kemudian hasil akan dinilai sehingga dapat disimpulkan dan dievaluasi. Berikut langkah-langkah pada penelitian ini ialah sebagai berikut:



Gambar 1. Tahapan Metode Penelitian

3.1 Dataset Pengujian

Dataset yang digunakan dalam penelitian ini adalah DeepWeeds yang terdiri dari 17.509 gambar berlabel dari delapan spesies gulma yang signifikan tumbuh alami di padang rumput Australia Utara bersumber dari Kaggle. Sedangkan untuk sampel yang digunakan dalam penelitian ini terdiri dari 8 jenis weeds dengan total sampel sebanyak 10.505 citra untuk data training dan 3.502 citra untuk data testing. Berikut adalah 8 jenis *weeds* yang digunakan dalam penelitian ini beserta penjelasannya:

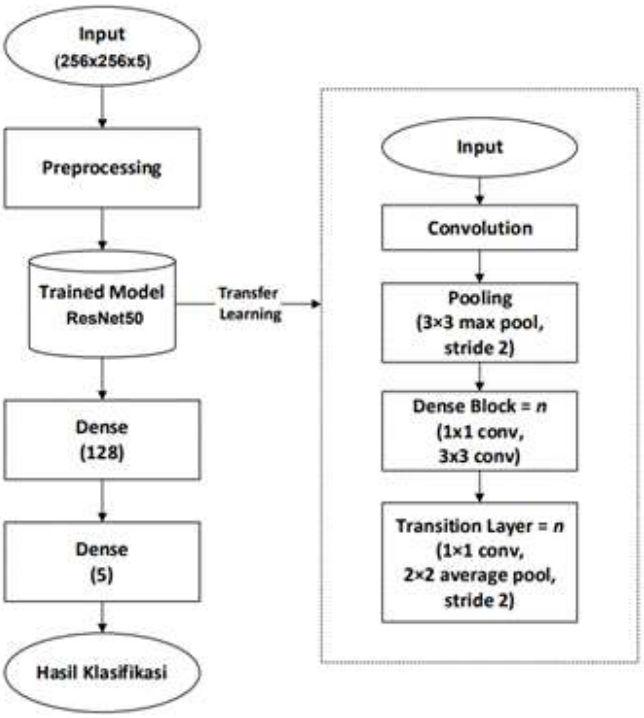
Table 1 Dataset Pengujian

No.	Variable	Train	Test
1	Citra tanaman Chinee apple	1125	226
2	Citra tanaman Snake weed	1016	204
3	Citra tanaman Lantana	1064	213
4	Citra tanaman Prickly acacia	1062	213
5	Citra tanaman Siam weed	1074	215
6	Citra tanaman Parthenium	1022	205
7	Citra tanaman Rubber vine	1009	202
8	Citra tanaman Parkinsonia	1031	207

3.2 Rancangan Pengujian

Gambar 2 merupakan rancangan model pengujian yang digunakan dalam penelitian ini. Dari rancangan tersebut dapat dilihat bahwa citra input yang dimasukan berukuran 256x256x3 piksel kemudian dilakukan preprocessing citra. Tahap preprocessing yang dilakukan kali ini yaitu menggunakan bantuan *ImageDataGenerator* dengan melakukan manipulasi terhadap citra seperti *horizontal_flip=True, vertical_flip=True, rotation_range=360, width_shift_range=0.2, height_shift_range=0.2, shear_range=0.1* dan *fill_mode="nearest"*.

Setelah data augmentasi dilakukan selanjutnya kita training dengan *pre-trained model ResNet50* untuk mendapatkan *feature extraction* dengan melakukan *convolusi* terhadap citra input sebelumnya dan dilakukan *max* pooling berukuran 3x3 dengan 2 *stride*. Kemudian dilakukan *dense block* dengan proses *convolusi* berukuran 1x1 dan 3x3 sebanyak *n*, dimana *n* merupakan banyaknya *dense blok* yang dilakukan secara berulang.



Gambar 2 Rancangan Pengujian

Kemudian ditambahkan *Transition layer* dengan ukuran konvolusi sebesar 1x1 dan average pooling berukuran 2x2 dengan 2 stride sebanyak n. *Transition layer* juga dilakukan berulang sebanyak n sehingga didapatkan extraksi fitur. Setelah kita dapatkan ekstraksi fitur didapat kemudian kita gunakan dua dense layer, dengan layer pertama berfungsi sebagai *activation ReLu (rectified linear unit)* berukuran 128 neuron dan layer kedua *Softmax* sejumlah 5 neuron sesuai dengan jumlah kelas data yang diambil dari *dataset*.

4. Hasil dan Pembahasan

Dataset *DeepWeeds* diinput ke dalam projek di *notebook* pada *Google Colaboratory* menggunakan *Tensorflow*. Untuk melakukannya dijalankan kode blok berikut yang dituliskan menggunakan bahasa pemrograman *python* di *notebook*.

4.1 Pemuatan Dataset

```
import tensorflow as tf
import tensorflow_datasets as tfds

data = tfds.load("deep_weeds", with_info = True)
```

Gambar 3. Kode Blok Untuk Mendownload Dataset DeepWeeds

Library tensorflow_datasets adalah kumpulan *dataset* yang siap digunakan dengan *framework Tensorflow*. Selanjutnya *dataset* yang telah didownload perlu dipisahkan dan disimpan menjadi 3 variable sekaligus menentukan jumlah data yang akan digunakan untuk *data_train*, *data_valid*, *data_test*. Untuk melakukannya dapat menjalankan blok kode berikut.

Split Data :

```
data_train, data = tfds.load("deep_weeds", with_info=True, split='train[:20%]')
data_valid , val_inf  = tfds.load("deep_weeds", with_info=True, split='train[80%:90%]')
data_test , test_inf  = tfds.load("deep_weeds", with_info=True, split='train[90%:100%]')

print(len(data_train))
print(len(data_valid))
print(len(data_test))
```

```
3502
1751
1751
```

Gambar 4. Kode Blok Untuk Split data_train, data_valid, data_test

Menggunakan *tfds.load*, *dataset* dapat dimuat sesuai nama pada *argument* pertama ke dalam *tf.data.Dataset* kemudian *argument split* akan memberikan jumlah data yang akan digunakan berdasarkan irisan dari total data yang ada sesuai angka yang dicantumkan. Data yang digunakan untuk *data_train* kali ini adalah 60% dari 17.509 data yaitu 10.500, untuk *data_validation* adalah 20% yaitu 3.502, untuk *data_test* adalah 20% yaitu 3.502.

4.2 Data *Preprocessing*

Proses selanjutnya penulis melakukan data preprocessing menggunakan *library math & tensorflow_addons*. Augmentasi yang dilakukan adalah *random_brightness(max_delta=0.5)* yaitu menyesuaikan kecerahan gambar secara acak dari rentang 0 hingga 0.5, *random_flip_left_right* yaitu membalikkan citra secara acak ke arah *horizontal*, *random_flip_up_down* yaitu membalikkan citra secara acak ke arah *vertical*, *random_hue* yaitu menyesuaikan rona gambar RGB secara acak dari rentang 0 hingga 0.1, *random_contrast* yaitu menyesuaikan kontras gambar secara acak dari rentang 0.7 hingga 1.3, *random_uniform* dengan *math.radius(-360) & math.radius(360)* yaitu menghasilkan nilai acak dari rentang -360° hingga 360° dalam nilai kembalian yang sudah diconvert menjadi *float* yang nantinya akan digunakan sebagai *argument* di *tfa.image.rotate* yaitu memutar gambar berlawanan arah jarum jam dengan sudut ditentukan dalam satuan radian. Semua ini dilakukan dengan menjalankan kode blok berikut.

Data Preprocessing :

```
import math
!pip install tensorflow_addons
import tensorflow_addons as tfa

batch_size = 16
def preprocess_weeds(dic):
  image = dic['image']
  # Scale to between 0 and 1
  preprocessed_image = tf.image.resize(image,[224,224])/255
  label = dic['label']
  return preprocessed_image, label

def augment(image,label):
  # Random brightness
  image = tf.image.random_brightness(image, max_delta=0.5)
  image = tf.image.random_flip_left_right(image)
  image = tf.image.random_flip_up_down(image)
  image = tf.image.random_hue(image, max_delta=0.1)
  # Adjust contrast
  image = tf.image.random_contrast(image, lower=0.7, upper=1.3)
  delta = tf.random.uniform([], math.radians(-360),math.radians(360))
  image = tfa.image.rotate(image, delta)
  return image, label
```

Gambar 5. Kode Blok Proses Data *Preprocessing*

4.3 Pembuatan Model

Agar data memperoleh akurasi yang baik dan mengurangi waktu komputasi, maka dilakukan metode *transfer_learning* dari *pre-trained* model ResNet50. Dari model terlatih tersebut kita akan mencari ekstraksi fiturnya saja yang nantinya akan saya gunakan sebagai dasar pembuatan model final, berikut merupakan proses yang dilakukan dalam mendapatkan ekstraksi fitur model ResNet50.

```
[17] from tensorflow.keras.layers import Input, Dense, Flatten, AveragePooling2D
     image_input = Input((224,224,3))
     resnet = tf.keras.applications.ResNet50(include_top=False, weights="imagenet")
     NUM_CLASSES=9
     model = tf.keras.Sequential(
         [image_input,
          resnet,
          AveragePooling2D(),
          tf.keras.layers.Flatten(),
          Dense(NUM_CLASSES, activation="softmax")])
     model.compile(optimizer="adam",loss="sparse_categorical_crossentropy",metrics="accuracy")

     Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/re
     94773248/94765736 [==============================] - 1s 0us/step
     94781440/94765736 [==============================] - 1s 0us/step
```

Gambar 6. Kode Blok Pembuatan Model *ResNet50*

Pada gambar diatas merupakan kode untuk mendapatkan ekstraksi fitur dari model terlatih *ResNet50*. Dataset yang kita miliki sebelumnya akan di *training* terlebih dahulu oleh *Imagenet* agar model dapat mengenali objek citra yang akan kita gunakan, kemudian citra *input* yang digunakan adalah berukuran 224x224x3 piksel dan *pooling* yang digunakan adalah *AveragePooling2D*. Setelah ekstraksi fitur diperoleh maka itulah model akhir kita selanjutnya yaitu melakukan penambahan *accelerator GPU* untuk mempercepat proses pelatihan model klasisfikasi. Jalankan blok kode berikut untuk memeriksa ketersediaan *accelerator GPU* pada *notebook*/projek.

```
print("Num GPUs Available: ", len(tf.config.experimental.list_physical_devices('GPU')))

Num GPUs Available:  1

tf.test.is_gpu_available(cuda_only=False, min_cuda_compute_capability=None)

WARNING:tensorflow:From <ipython-input-11-78f884b5c1a9>:1: is_gpu_available (from tensorf
Instructions for updating:
Use `tf.config.list_physical_devices('GPU')` instead.
True

tf.config.list_physical_devices('GPU')

[PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
```

Gambar 7. Kode Blok Untuk Menerapkan *GPU* pada *Notebook*/Projek

4.4 Hasil Pengujian Model

Pengujian dilakukan untuk mendapatkan hasil akurasi dari setiap model uji yang dibuat sebelumnya. Data citra yang digunakan dalam penelitian ini adalah data 8 rumput liar dengan jumlah data latih (*train*) sebanyak 17.509 yang dibagi menjadi 60% data *training* sebanyak 10.505 dan 20% validasi sebanyak 3.502 serta data uji (*test*) sebanyak 3.502 citra rumput liar.

Setelah model sebelumnya selesai dibuat, selanjutnya melakukan pelatihan data rumput liar dengan melakukan melakukan *fit model*. Dalam melakukan *fit model* akan digunakan epoch sebanyak 50 kali, batch_size = 8 dan test_size = 60% training dan 20% validasi. *Epoch* berarti berapa kali jaringan akan melihat seluruh kumpulan data, sedangkan *batch_size* adalah jumlah contoh pelatihan dalam satu *forward/backward pass*. Semakin tinggi nilai *batch_size* maka akan semakin banyak memori yang dibutuhkan.

```
from tensorflow.keras.callbacks import Callback

# create custom callback
class CustomCallback(Callback):
  def on_epoch_end(self, epoch, logs={}):
    if logs.get('accuracy') > 0.90:
      print('Accuracy reach above 85% -- Stop Training')
      self.model.stop_training = True # Stop model training

my_callback = CustomCallback()

model.fit(data_train_gen,
          batch_size=batch_size,
          epochs=50,
          validation_data=data_valid_gen,
          verbose=1,
          #history callback plot keras
          callbacks=[my_callback]
          )
```

Gambar 8 Blok Kode Pengujian Model *ResNet50*

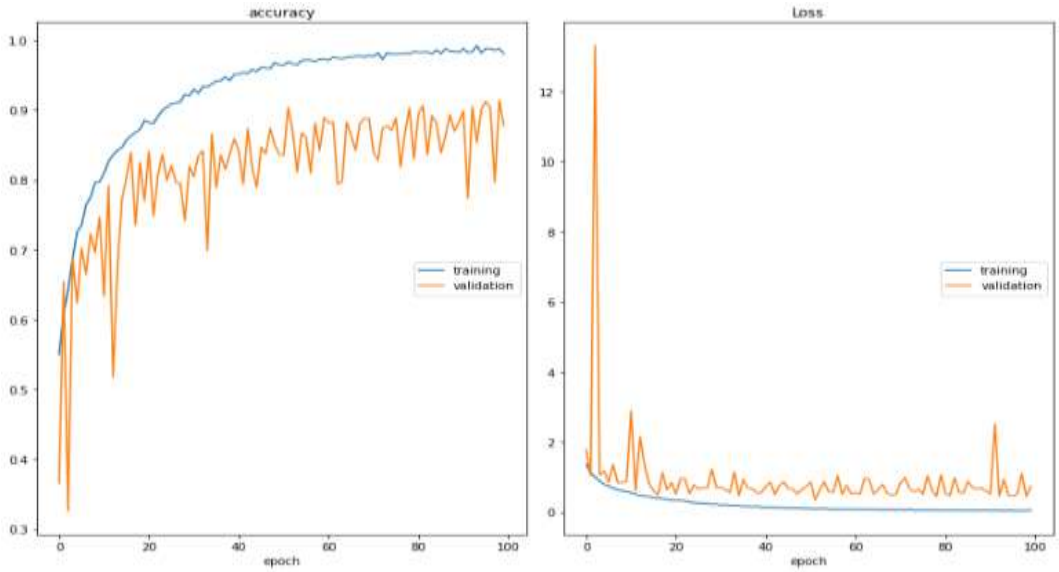
Table 2. Hasil Fit Model

epoch	Data *Train*		Data Validation	
	loss	accuracy	val_loss	val_accuracy
1	1.5711	0.5188	1.3715	0.5443
2	1.3805	0.5356	1.2184	0.5637
3	1.2992	0.5484	1.2085	0.5745
4	1.3000	0.5524	1.4301	0.5277
5	1.2319	0.5711	1.3450	0.5594

epoch	Data *Train*		Data Validation	
	loss	accuracy	val_loss	val_accuracy
6	1.1878	0.5829	1.1857	0.5720
7	1.1596	0.5884	1.1948	0.5762
8	1.1596	0.5966	1.3491	0.5551
9	1.0986	0.6062	1.2523	0.5540
10	1.0688	0.6154	1.3707	0.5605
11	1.0451	0.6183	1.3300	0.5660
12	1.0003	0.6413	0.9614	0.6565
13	0.9777	0.6488	1.0681	0.6428
14	0.9419	0.6604	1.0333	0.6391
15	0.9536	0.6657	0.9643	0.6750
16	0.8919	0.6781	0.9449	0.6633
17	0.8707	0.6925	1.0613	0.6334
18	0.8522	0.6943	0.8703	0.7002
19	0.8402	0.6987	0.9437	0.6676
20	0.8192	0.7075	1.1708	0.6396
21	0.7934	0.7139	0.8232	0.7133
22	0.7757	0.7178	0.7754	0.7253
23	0.7666	0.7282	0.7792	0.7250
24	0.7557	0.7266	1.0406	0.6645
25	0.7262	0.7420	0.7656	0.7493
26	0.7056	0.7485	0.8922	0.6962
27	0.7022	0.7022	0.8620	0.7087
28	0.6809	0.7556	0.9195	0.7019
29	0.7023	0.7578	0.7628	0.7464
30	0.6633	0.7661	1.2207	0.7150
31	0.6367	0.7699	0.7141	0.7584
32	0.6315	0.7767	0.8314	0.7461
33	0.6170	0.7812	0.7155	0.7630
34	0.6127	0.7862	0.7996	0.7544
35	0.6098	0.7829	0.7271	0.7776
36	0.5882	0.7904	0.8163	0.7430
37	0.5826	0.7917	0.9203	0.6976
38	0.5716	0.7954	0.9142	0.7359
39	0.5588	0.8031	0.7870	0.7470
40	0.5521	0.8070	0.7090	0.8010
41	0.5304	0.8142	1.0109	0.7427
42	0.5332	0.8119	0.8041	0.7433
43	0.5220	0.8150	0.6876	0.7853
44	0.5090	0.8256	0.6939	0.7790
45	0.5019	0.8248	0.6845	0.7984
46	0.4909	0.8288	0.8581	0.7499
47	0.4816	0.8339	0.7539	0.7539
48	0.4703	0.8354	0.7443	0.7684
49	0.4630	0.8394	**0.6454**	**0.8078**
50	**0.4529**	**0.8432**	0.8913	0.7864

Tabel 2 merupakan hasil dari pelatihan data *train* dan data *test* dengan menggunakan *epoch* sebanyak 50 kali. Dapat diketahui bahwa iterasi menghasilkan nilai *accuracy* dan nilai *loss* dari data *train* dan data *validasi*. Nilai *accuracy* merupakan nilai yang dapat digunakan sebagai acuan dalam mengetahui tingkat keberhasilan/kelayakan model yang telah dibuat dan nilai *loss* merupakan ukuran dari kegagalan/error yang dibuat *networks* yang bertujuan untuk meminimalisirnya. Pada data *train* diperoleh nilai akurasi tertinggi sebesar 0,8432 pada epoch ke-50 sedangkan nilai *loss* terendah 0,4529 di epoch ke-50, kemudian pada data *validasi* diperoleh nilai akurasi tertinggi sebesar 0,8078 pada epoch ke-49 sedangkan nilai *loss* terendah

0,6454 di epoch ke-49. Dari hasil pelatihan data *train* dan data *validasi* tersebut dapat kita visualisasikan kedalam plot/grafik sebagai berikut.



Gambar 9. Plot *Accuracy & Loss* Data *Train vs* Data *Val* Model

Dari gambar diatas dapat diketahui bahwa hubungan antara nilai *accuracy* dan nilai *loss* pada data *train* dan data *validasi* dengan jumlah *epoch/iterasi*. Korelasi/hubungan yang terjadi pada nilai akurasi menunjukan korelasi positif yang memiliki hubungan searah dengan ketentuan semakin banyak jumlah *epoch* yang digunakan maka nilai *accuracy* data *train* dan data *validation* semakin tinggi. Berbanding terbalik dengan nilai *accuracy*, hubungan antara banyaknya *epoch* dengan nilai *loss* merupakan korelasi negatif dimana banyaknya jumlah epoch yang digunakan akan mempengaruhi nilai *loss* yang dihasilkan pada pelatihan data semakin kecil. Berdasarkan hasil tersebut, maka dapat disimpulkan bahwa untuk memperkecil nilai *loss* yang diharapkan maka dapat dilakukan dengan cara memperbanyak jumlah *epoch* pada proses *training*. Setelah kita melakukan training *model*, selanjutnya kita lakukan evaluasi kinerja model pada set pengujian. Evaluasi dilakukan guna melihat kemungkinan kegagalan objek citra yang dibaca dalam proses klasifikasi, juga akan didapatkan nilai akurasi dan nilai loss dengan probabilitas tertinggi yang akan didapat dari keseluruhan model pengujian.

```
loss, accuracy = reconstructed_model.evaluate(data_test_gen)
print(f"Loss: {loss}")
print(f"Test accuracy: {round(accuracy * 100, 2)}%")

438/438 [==============================] - 15s 32ms/step - loss: 0.7819 - accuracy: 0.7773
Loss: 0.7818941473960876
Test accuracy: 77.73%
```

Gambar 10 Evaluasi Pengujian Model

Gambar diatas merupakan kode yang digunakan untuk melihat evaluasi kinerja model, dari gambar tersebut dapat dilihat evaluasi akurasi yang dihasilkan dari data *test* dengan nilai akurasi sebesar 77,73% dan nilai *loss* sebesar 0,781. Dari evaluasi tersebut maka didapat data classification report pada tabel berikut.

Table 3. *Classification Report*

Label	Precision	Recall	F1-Score	Support
Chinee apple	0.73	0.55	0.63	234
Lantana	0.86	0.64	0.74	211
Parkinsonia	0.73	0.62	0.67	213
Parthenium	0.90	0.59	0.71	203

Prickly acacia	0.65	0.72	0.68	217
Rubber vine	0.67	0.79	0.72	200
Siam weed	0.79	0.86	0.83	218
Snake weed	0.58	0.74	0.65	235
Negatives	0.83	0.86	0.85	1771
micro avg	0.78	0.78	0.78	3502
macro avg	0.75	0.71	0.72	3502
weighted avg	0.78	0.78	0.78	3502
samples avg	0.78	0.78	0.78	3502

4.5 Tes Akurasi Model

Pada tahap ini saya akan mencoba melakukan prediksi citra pada model yang sudah dibuat sebelumnya dengan memasukan salah satu citra rumput liar dan kemudian model akan membaca citra tersebut dengan hasil prediksinya. Tahap awal kita akan memilih citra Lantana sebagai sampel dengan memasukan kode berikut.

Gambar 11. Contoh Citra *Lantana*

```
from tensorflow.keras.preprocessing import image

img = image.load_img("20170126-095404-0.jpg", target_size=(224, 224, 3))

x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)

images = np.vstack([x])

predictions = reconstructed_model.predict(images)

classe = np.argmax(predictions, axis=1)
labels = weeds_label = ["Chinee apple", "Lantana", "Parkinsonia", "Parthenium", "Prickly acacia", "Rubber vine", "Siam weed", "Snake weed", "Negatives"]

print(predictions[0])
print(predictions[0][int(classe[0])])
print(list(labels)[int(classe[0])])
```

```
1/1 [==============================] - 0s 23ms/step
[0. 0. 0. 0. 0. 0. 0. 0. 1.]
1.0
Negatives
```

Gambar 12 Hasil Prediksi Citra *Lantana*

Dengan melihat gambar diatas bahwa akurasi yang didapat dari model sebelumnya dalam mengklasifikasikan sampel citra yang masih kurang sesuai. Dari hasil akurasi tersebut bisa dikatakan bahwa model yang dibuat sebelumnya belum cukup layak dan belum bisa dikatakan berhasil dalam mengklasifikasikan citra rumput liar.

Evaluasi dari hasil pengujian model dengan metode transfer learning dari model yang sudah terlatih ResNet50 yang dilatih menggunakan data train sebanyak 10.505 citra diperoleh

nilai evaluasi yang dihasilkan dari data test sebanyak 3.502 citra dengan melakukan pengujian sebanyak 50 kali, batch_size=8, test_split=0.2 didapat nilai *precision, recall* dan nilai *f1-score* sebagai berikut.

Table 4 Hasil Akhir *Classification Report*

Classification Report		
Precision	**Recall**	**F1-Score**
78%	78%	78%

5. Simpulan

Berdasarkan penelitian dan hasil penerapan Convolutional Neural Network (CNN) dengan model *pre-Trained* dari ResNet50 dalam mengklasifikasikan 8 rumput liar, dapat diambil kesimpulan bahwa dalam penelitian ini citra dapat melewati proses preprocessing dengan baik yaitu dengan mengubah ukuran dimensi citra menjadi 224x224x3 piksel dan kemudian dilakukan data augementasi berupa *horizontal_flip*, *vertical_flip*, *rotation_range*, *brightness* & *contrast.* Selain itu akurasi yang akan dicapai semakin baik apabila digunakan data train yang digunakan semakin besar. Hal ini dibuktikan dengan pengujian pada jumlah data training sebanyak 60% dan data validasi sebanyak 20% pada setiap model pengujian. Terakhir, memanfaatkan fitur ekstraksi dari model terlatih ResNet50 dan melakukan evaluasi pengujian model final pada epoch sebanyak 50 kali, batch_size=8, dan validation_split=60% training dan 20% validasi diperoleh nilai precision 78%, recall 78% dan f1-score 78%. Beberapa saran bagi penelitian selanjutnya dari peneliti berdasarkan penelitian ini khususnya yang menggunakan metode Convolutional Neural Network menggunakan Google Collaboratory disarankan untuk menggunakan internet yang cepat dan stabil agar mempercepat proses pelatihan CNN, guna mempercepat pemrosesan pelatihan CNN dibutuhkan perangkat Graphical Processing Unit (GPU) yang cukup besar dan mengembangkan program identifikasi citra rumput liar dengan model dan metode CNN yang lain seperti MobileNetV2 atau VGG19 dan mampu mengembangkan perolehan model CNN menjadi semakin dapat menghasilkan output yang bervariatif, handal dan representatif dalam mengenali objek.

Daftar Referensi

[1] M. McDaniel, S. Teng, E. Sprout, H. Costa, H. Hall, J. Hunt, D. Boudreau, T. Ramroop, K. Rutledge, "*The Art and Science of Agriculture*," National Geographic, 28 Juli 2022. [Online]. https://education.nationalgeographic.org/resource/ agriculture [Diakses: 30 Mei 2022].

[2] Kementerian Perdagangan Republik Indonesia, "*Produk Unggulan Indonesia*," Kementerian Perdagangan Republik Indonesia, 2022. [Online]. http://ppei.kemendag.go.id/produk-unggulan-indonesia/ [Diakses: 30 Mei 2022].

[3] Kementerian Pertanian Republik Indonesia, "*Empat dari Sepuluh Produk Ekspor Andalan Indonesia Adalah Komoditas Pertanian*," Kementerian Pertanian Republik Indonesia, 2022. [Online]. https://www.pertanian.go.id/home/?show= news&act=view&id=3813#:~:text= Dari%20sekian%20banyak%20produk%20yang,%2C%20sawit%2C%20kakao%20dan%20kopi. [Diakses: 30 Mei 2022].

[4] R.A. Sari, "*Rumput Lapang Sebagai Hijauan Pakan Ternak*," Dinas Perikanan Dan Peternakan Kabupaten Bogor, 26 Oktober 2020. [Online]. https://diskanak. bogorkab.go.id/rumput-lapang-sebagai-hijauan-pakan-ternak/ [Diakses: 30 Mei 2022].

[5] I.N. Widiarta, "Sistem Pendukung Pengambilan Keputusan Pengendalian Hama Terpadu Pada Tanaman Padi Berbasis Teknologi Informasi". *Jurnal Penelitian dan Pengembangan Pertanian Vol*, vol. 40, no. 1, pp. 9-20, 2021.

[6] M. Rizki, Z. Zaini, M.A. Nikhaldo, T.U. Septiyeni, & T. Yuliswar, "Inovasi Teknologi Machine Learning Bagi Masyarakat Di Ekowisata Sungkai Green Park Nagari Lambung Bukit Kecamatan Pauh Padang. *Jurnal Hilirisasi IPTEKS*, vol. 5, no. 1, pp. 48-59, 2022.

[7] A. Izzuddin, & M.R. Wahyudi, "Pengenalan pola daun untuk membedakan tanaman padi dan gulma menggunakan metode Principal Components Analysis (PCA) dan Extreme Learning Machine (ELM)". *ALINIER: Journal of Artificial Intelligence & Applications*, vol. 1, no. 1, pp. 44-51, 2020.

[8] A.A. Nugraha, S. Panjaitan, D. Joachim, & S. Lubis, “Modernisasi Pertanian Dengan Pengembangan Robot Tani Dalam Rangka Penyamarataan Otomatisasi Pada Industri Agriculture”. *PROSIDING SERINA*, vol. 2, no. 1, pp. 617-626, 2022.
[9] J.E. Hutagalung, & D. Dahriansah, “Desain Robot Pemotong Rumput Dengan Perintah Suara Berbasis Android”. *Sains dan Teknologi Informasi*, *5*(1), vol. 5, no. 1, pp. 102-109, 20219.
[10] W. Cahya, M. Febriansyah, F. Angellia, & T.W. Widyaningsih, “Implementasi Arm Robot pada Smart Farming Berbasis Internet of Things”. *Techno. Com*, vol. 21, no. 4, pp. 927-934, 2022.
[11] L.M. Azizah, S.F. Umayah, & F. Fajar, “Deteksi Kecacatan Permukaan Buah Manggis Menggunakan Metode Deep Learning dengan Konvolusi Multilayer”. *Semesta Teknika*, vol. 21, no. 2, pp. 230-236, 2018.
[12] G.A.W. Satia, E. Firmansyah, & A. Umami, “Perancangan sistem identifikasi penyakit pada daun kelapa sawit (Elaeis guineensis Jacq.) dengan algortme deep learning convolutional neural networks”. *Jurnal Ilmiah Pertanian*, vol. 19, no. 1, pp. 1-10, 2022.
[13] N. Teimouri, M. Dyrmann, P.R. Nielsen, S.K. Mathiassen, Somerville GJ and Jørgensen RN. “Weed Growth Stage Estimator Using Deep Convolutional Neural Networks,” *Sensors*, vol. 18, no. 5, pp. 580, 2018 doi: 10.3390/s18051580.
[14] E. I. Haksoro and A. Setiawan, “Pengenalan Jamur Yang Dapat Dikonsumsi Menggunakan Metode Transfer Learning Pada Convolutional Neural Network,” J. ELTIKOM, vol. 5, no. 2, pp. 81–91, 2021, doi: 10.31961/eltikom.v5i2.428.
[15] F. Hurriyatul, M. Rizal, Deteksi Gulma Berdasarkan Warna HSV dan Fitur Bentuk Menggunakan Jaringan Syaraf Tiruan. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 8, no. 5, pp. 929-938, 2021.