

## TRANSFER LEARNING MODEL CNN PRALATIH UNTUK KLASIFIKASI BUNGA IRIS BERBASIS CITRA

Ahmad Sabri

Universitas Gunadarma  
Jl. Margonda Raya 100, Depok Kota Depok 16424 Jawa Barat  
sabri@staff.gunadarma.ac.id

### ABSTRAK

*Klasifikasi bunga iris merupakan topik yang populer pada bidang kecerdasan buatan. Penelitian yang dilakukan untuk topik ini pada umumnya adalah model klasifikasi iris yang mengacu pada dataset iris standar yang berbentuk tabular numerik. Penelitian ini mengusulkan pendekatan alternatif, yaitu menggunakan model kecerdasan buatan yang dibangun berdasar dataset iris berbasis citra. Model yang digunakan adalah model CNN pralatih MobileNet V2, VGG 16 dan Inception V3. Transfer learning dilakukan dengan mengganti top layer dari ketiga model tersebut dengan flattened layer yang dirangkai dengan output layer berupa dense layer 3 simpul dengan fungsi aktivasi softmax. Metode transfer learning dipilih karena jumlah trainable parameter yang jauh lebih sedikit dan waktu training yang lebih singkat dibandingkan menggunakan model Convolutional Neural Network (CNN) yang dibangun dari awal. Perbandingan kinerja ketiga model ditinjau dari metrik precision, recall, F1, dan akurasi. Hasil evaluasi menunjukkan nilai metrik yang bervariasi, di mana tidak ada satu di antara ketiga model tersebut yang semua metriknya mengungguli kedua model lainnya. Dalam hal akurasi, MobileNet V2 memberikan akurasi terbaik yaitu sebesar 91%, diikuti oleh Inception V3 sebesar 87% dan terakhir VGG16 sebesar 84%.*

**Kata Kunci:** *dataset iris, klasifikasi, model pralatih, transfer learning*

## PENDAHULUAN

Klasifikasi bunga iris merupakan masalah klasik dan populer dalam bidang kecerdasan buatan. Model klasifikasi bunga iris pada umumnya mengacu pada dataset iris standar yang dibangun oleh Fisher pada tahun 1936 [1] untuk keperluan taksonomi. Dataset ini berbentuk tabular numerik dengan 4 atribut (*sepal length*, *sepal width*, *petal length*, *petal width*), yang mendefinisikan 3 spesies bunga iris (iris setosa, iris versicolor, iris virginica). dengan 50 sampel per spesiesnya. Fisher saat itu tidak menyadari bahwa dataset yang ia bangun tersebut menjadi dataset standar dalam bidang kecerdasan buatan puluhan tahun kemudian.

Cukup banyak penelitian tentang model kecerdasan buatan untuk klasifikasi bunga iris dengan menggunakan dataset iris standar. Di antaranya adalah penelitian [2] yang menggunakan model *Support Vector Machine* (SVM), regresi logistik dan *k-Nearest Neighbors* (kNN) untuk klasifikasi bunga iris, di mana ketiga model tersebut berturut-turut memberikan akurasi 96%, 91% dan 93%. Selanjutnya penelitian [3] menggunakan *Deep Neural Network* (DNN) untuk klasifikasi bunga iris dengan dataset iris standar [1] yang memberikan akurasi 100% untuk klasifikasi iris setosa dan iris versicolor, dan 87,5% untuk iris virginica. Penelitian [4] menggunakan *neural network* dengan dataset iris standar untuk klasifikasi bunga iris, dan menghasilkan akurasi 96,67%.

Beberapa penelitian klasifikasi berbasis citra untuk berbagai spesies bunga juga telah banyak dilakukan sebelum ini. Penelitian [5] menerapkan *transfer learning* pada model pralatih VGG16 untuk klasifikasi bunga pada dataset citra bunga Oxford 102 [6], dataset Oxford 17 [7], dan dataset Zou Nagy [8]. Terhadap masing-masing dataset tersebut, model mereka memberikan akurasi 99,0%, 98,5%, 97,1%.

Penelitian [9] menggunakan *Convolutional Neural Network* (CNN) untuk klasifikasi bunga pada dataset citra bunga KRD26 (26 spesies bunga yang tumbuh di Kurdistan) dan menghasilkan akurasi 94,61%. Penelitian [10] melakukan identifikasi tanaman berdasarkan daun dan

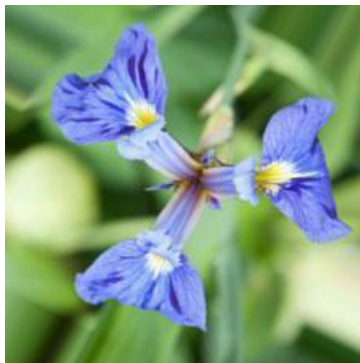
bunganya. Eksperimen yang dilakukan memberikan ketelitian berturut-turut 98%, 85% dan 74% untuk identifikasi tanaman berdasarkan daun saja, bunga saja, dan kombinasi keduanya.

Beberapa penelitian menerapkan *transfer learning* dari model CNN pralatih (*pre-trained model*), antara lain penelitian [5] yang menerapkan VGG16 [11]. *Transfer learning* lebih efisien dibandingkan menggunakan model CNN yang dibangun dari awal, karena hanya melatih layer terakhirnya saja. Keuntungan dari melakukan *transfer learning* adalah kuantitas *trainable parameter* yang relatif lebih sedikit, dan waktu pelatihan yang lebih singkat dibandingkan melatih model CNN dari awal.

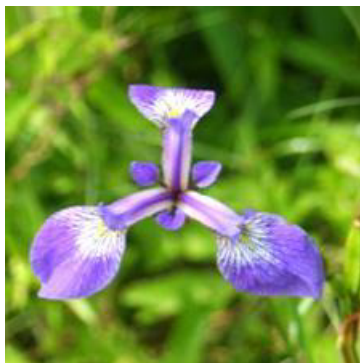
Sejauh penelusuran yang dilakukan, belum terdapat penelitian tentang klasifikasi berbasis citra untuk bunga iris dengan menggunakan model CNN, baik model CNN pralatih ataupun model yang dibangun dari awal. Berdasar latar belakang tersebut, penelitian ini mengusulkan *transfer learning* terhadap model CNN pralatih untuk melakukan klasifikasi bunga iris berbasis citra. Penelitian ini menggunakan model pralatih MobileNet V2 [12], VGG16 [11], dan Inception V3 [13], [14].

## METODE PENELITIAN

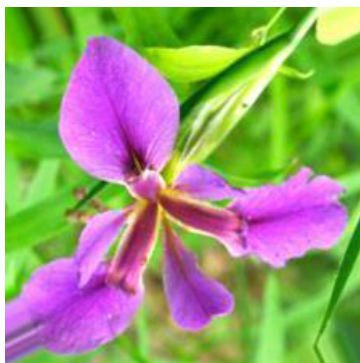
Dataset yang digunakan adalah citra RGB dari bunga iris dengan ukuran  $224 \times 224$  piksel dengan karakteristik: objek tampak dari atas secara utuh dan tunggal. Dataset pada penelitian ini diperoleh dengan menyeleksi citra yang memenuhi karakteristik tersebut dari Kaggle [15]. Dataset dasar terdiri dari 18 citra iris setosa, 42 citra iris versicolor, dan 23 citra iris virginica. Kemudian terhadap dataset tersebut dilakukan *image augmenting* dengan salah satu/kombinasi dari transformasi rotasi (*range* = 40), *shear* (*range* = 0,2), *zoom* (*range* = 0,2), *brightness* (*range* = 0,5 to 1,5), *horizontal flip* dan *vertical flip*. Dataset hasil augmentasi terdiri dari 288 citra iris setosa, 288 citra iris versicolor, dan 275 citra iris virginica. Gambar 1, 2, dan 3 menampilkan sebuah sampel citra dari masing-masing jenis iris.



**Gambar 1.** *Iris setosa.*



**Gambar 2.** *Iris versicolor.*



**Gambar 3.** *Iris virginica.*

*Transfer learning* dilakukan terhadap 3 model pralatih: MobileNet V2, VGG16 dan Inception V3. Pada *transfer learning*, pembaruan parameter terjadi pada *top layer*, sedangkan parameter internal dari ketiga model pralatih ditetapkan sebagai *non-trainable*. Pada penelitian ini, *top layer* dari ketiga model digantikan *flattened layer* dengan koefisien *dropout* 0.2, kemudian dirangkai dengan *output layer* berupa *dense layer* dengan 3 simpul dengan fungsi aktivasi *softmax*. Banyaknya parameter model (*trainable* maupun *non-trainable*) setelah dilakukan modifikasi pada masing-

masing *top layer* diberikan oleh Tabel 1.

**Tabel 1**

Parameter	MobileNet V2	VGG16	Inception V3
Trainable	3.843	75.267	55.299
Non trainable	2.257.984	14.714.688	21.802.784
Total	2.261.827	14.789.955	21.858.083

*Transfer learning* dilakukan dengan menggunakan bahasa pemrograman Python, yang berjalan pada hardware dengan CPU Intel i5 11300H (3 GHz), GPU NVIDIA RTX3050 dan RAM 16 GB. Skenario pelatihan diterapkan dengan menetapkan hiperparameter berikut:

- *train test split* = 0,8,
- *epoch* = 30,
- *batch size* = 34,
- *validation split* = 0,2.

Untuk mencegah terjadi *overfitting*, diterapkan *callback* dengan parameter *patience* = 4, *monitor* = *val loss*, *mode* = *min*. Parameter *patience* menentukan banyaknya iterasi berurutan tanpa ada perbaikan terhadap parameter yang dimonitor, dalam hal ini adalah *validation loss*. Dengan penetapan tersebut, *early stopping* dilakukan jika dalam 4 kali iterasi berurutan nilai *validation loss* tidak mengalami perbaikan.

Evaluasi model dilakukan berdasarkan matriks konfusi, yang memberikan metrik *precision*, *recall*, *F1-score*, dan akurasi.

Perbandingan kinerja model ditinjau berdasarkan mean dan variansi dari metrik masing-masing model, serta waktu training.

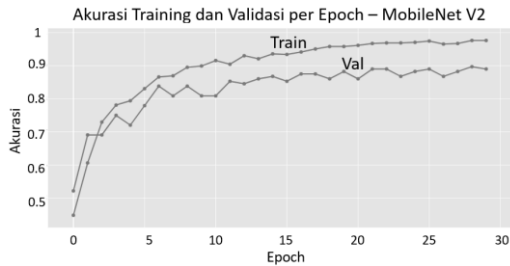
## HASIL DAN PEMBAHASAN

### Training dan Validasi

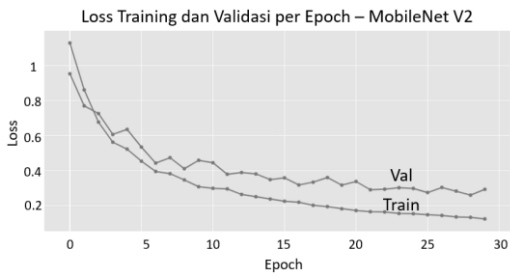
Pelatihan model dilakukan dengan menggunakan 680 citra iris, terdiri dari 235 citra iris setosa, 224 citra iris versicolor dan 221 citra iris virginica. Hasil training dan validasi dijelaskan berikut ini.

Nilai akurasi training dan akurasi validasi mengalami peningkatan secara konsisten pada setiap *epoch*-nya, yang masing-masing mencapai nilai 0,9761 dan 0,8897 di akhir training. Demikian pula dalam hal *training loss* dan *validation loss*, terjadi penurunan yang konsisten pada setiap

*epoch*-nya, dan masing-masing mencapai nilai 0,1245 dan 0,2934 di akhir training. Proses training menjalani keseluruhan 30 *epoch* tanpa *early stopping*. Grafik progres pelatihan MobileNet V2 diberikan oleh Gambar 4 (akurasi vs *epoch*) dan Gambar 5 (*loss* vs *epoch*).



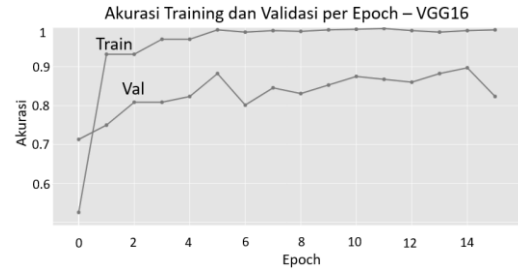
**Gambar 4.** Akurasi vs Epoch pada training dan validasi MobileNet V2.



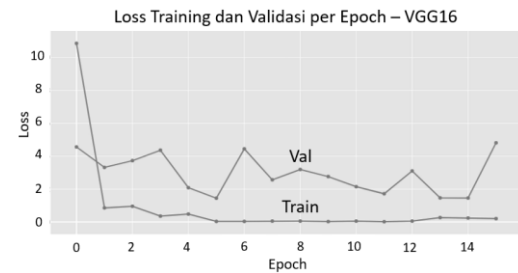
**Gambar 5.** Loss vs Epoch pada training dan validasi MobileNet V2.

Untuk pelatihan VGG16, nilai akurasi training mengalami peningkatan secara konsisten pada setiap *epoch*-nya, mencapai 0,9945 di *epoch* terakhir. Model dengan cepat mencapai akurasi melebihi 0,9 pada *epoch* ke 2. Pada akurasi validasi terjadi sedikit fluktuasi dan mencapai maksimum 0,8971 pada *epoch* ke 13.

Dalam hal *training loss*, terjadi penurunan *loss* dengan tajam pada *epoch* ke 2, dan cenderung stagnan setelah *epoch* ke 6. *Training loss* terendah dicapai pada *epoch* terakhir dengan nilai 0,2057. Pada *validation loss* terjadi fluktuasi yang diduga mengindikasikan *overfitting*. Proses training mengalami *early stopping* pada *epoch* ke 16 dari 30 *epoch* yang diskenariokan. Grafik progres pelatihan VGG16 diberikan oleh Gambar 6 (akurasi vs *epoch*) dan Gambar 7 (*loss* vs *epoch*).



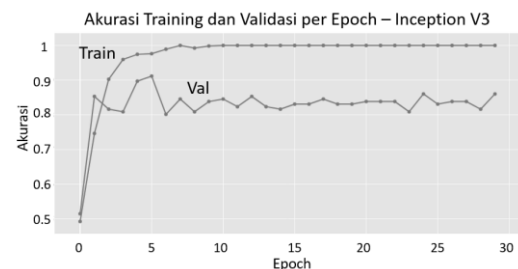
**Gambar 6.** Akurasi vs Epoch pada training dan validasi VGG16.



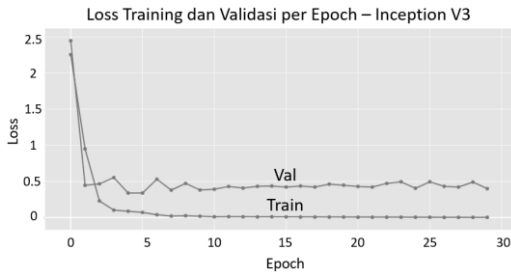
**Gambar 7.** Loss vs Epoch pada training dan validasi VGG16.

Untuk pelatihan Inception V3, nilai akurasi training mengalami peningkatan secara konsisten di *epoch* awal, untuk kemudian stagnan pada angka 1,000. Untuk akurasi validasi secara umum berkisar antara 0,8 dan 0,9.

Dalam hal *training loss*, terjadi penurunan yang tajam di *epoch* awal, untuk kemudian stagnan pada *epoch* selanjutnya di kisaran 0,003. Demikian pula halnya untuk *validation loss* stagnan pada kisaran 0,4. Keadaan ini mengindikasikan *overfitting*. Proses training menjalani keseluruhan 30 *epoch* tanpa *early stopping*. Grafik progres pelatihan VGG16 diberikan oleh Gambar 8 (akurasi vs *epoch*) dan Gambar 9 (*loss* vs *epoch*).



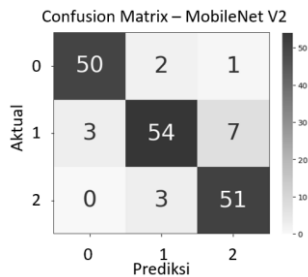
**Gambar 8.** Akurasi vs Epoch pada training dan validasi Inception V3.



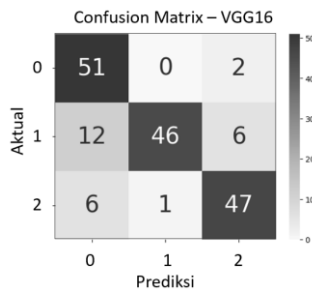
**Gambar 9.** Loss vs Epoch pada training dan validasi Inception V3.

**Testing dan Evaluasi Metrik**

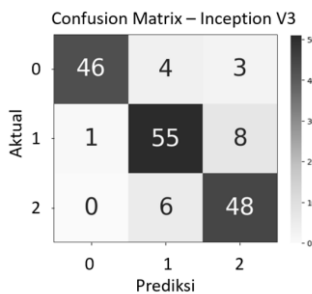
Pengujian dilakukan terhadap 171 citra iris yang terdiri dari 53 iris setosa, 64 iris versicolor dan 54 iris virginica. Matriks konfusi hasil pengujian MobileNet V2, VGG16 dan Inception V3 berturut-turut diberikan oleh Gambar 10, 11 dan 12.



**Gambar 10.** Matriks konfusi MobileNet V2.

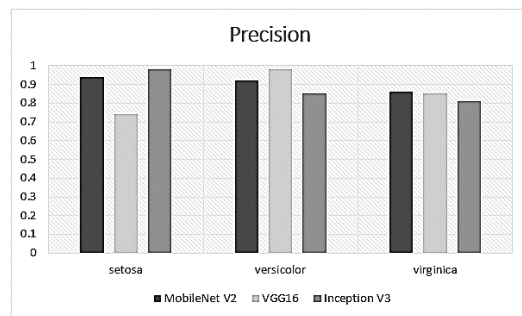


**Gambar 11.** Matriks konfusi VGG16.

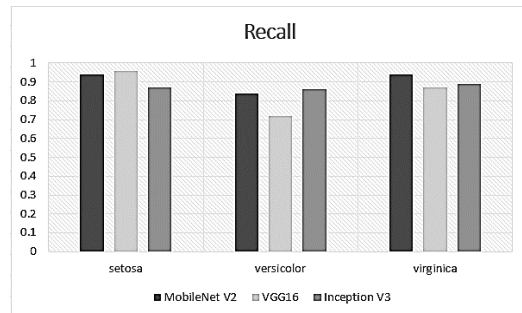


**Gambar 12.** Matriks konfusi Inception V3.

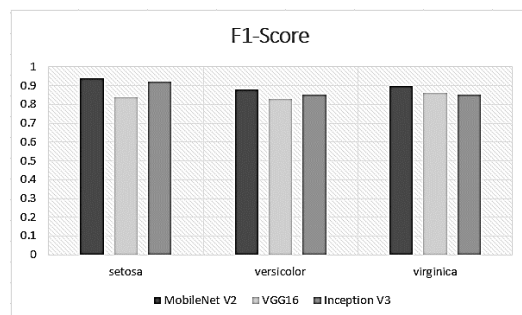
Dalam hal akurasi, MobileNet V2 memberikan nilai tertinggi sebesar 0,91, di atas akurasi Inception V3 dan VGG16 masing-masing sebesar 0,87 dan 0,84. Precision tertinggi diberikan oleh VGG16 untuk prediksi iris versicolor dan oleh Inception V3 untuk prediksi iris setosa, keduanya dengan nilai 0,98. Recall tertinggi diberikan oleh VGG16 untuk prediksi iris setosa. F1-score tertinggi diberikan oleh MobileNet V2 sebesar 0,94. Rekapitulasi ketiga model untuk setiap metriknya secara grafis diberikan oleh Gambar 13, 14, 15, dan 16.



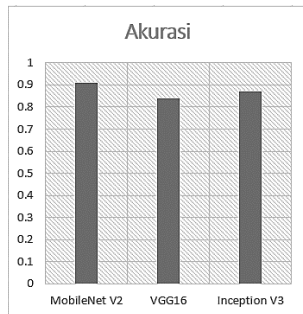
**Gambar 13.** Precision untuk setiap model berdasarkan jenis iris.



**Gambar 14.** Recall untuk setiap model berdasarkan jenis iris.



**Gambar 15.** F1-score untuk setiap model berdasarkan jenis iris.



Gambar 16. Akurasi untuk setiap model.

Rekapitulasi metrik secara tabular diberikan oleh Tabel 2. Dari rekapitulasi metrik tersebut diketahui bahwa tidak terdapat model yang mengungguli kedua model lainnya untuk semua metrik

Tabel 2. Rekapitulasi metrik untuk setiap model.

Kelas	precision	recall	f1-score	accuracy
<b>MobileNet V2</b>				
setosa	0.94	0.94	0.94	0.91
versicolor	0.92	0.84	0.88	
virginica	0.86	0.94	0.9	
<b>VGG16</b>				
setosa	0.74	0.96	0.84	0.84
versicolor	0.98	0.72	0.83	
virginica	0.85	0.87	0.86	
<b>Inception V3</b>				
setosa	0.98	0.87	0.92	0.87
versicolor	0.85	0.86	0.85	
virginica	0.81	0.89	0.85	

Perbandingan metrik secara agregat antara ketiga model diberikan oleh Tabel 3. Dalam hal ini, MobileNet V2 mengungguli kedua model lainnya dengan mean tertinggi dan deviasi standar terendah. Kinerja terbaik berikutnya diberikan oleh Inception V3, diikuti oleh VGG16.

Dalam hal waktu training, Inception V3 memiliki waktu tersingkat, diikuti oleh VGG16 dan MobileNet V2. Hasil ini mengindikasikan terdapat *trade-off* antara kinerja dengan waktu training.

Tabel 3. Mean dan variansi metrik dan waktu training untuk setiap model.

Model	Mean	Std Dev	Waktu Training (detik)
MobileNet V2	0.9070	0.0347	131.862
VGG16	0.8490	0.0769	58.478
Inception V3	0.8750	0.0443	39.838

## PENUTUP

Penelitian ini menerapkan transfer learning model pralatih MobileNet V2, VGG16 dan Inception V3 untuk klasifikasi bunga iris berbasis citra. Ketiga model hasil pelatihan dibandingkan berdasarkan metrik *precision*, *recall*, *F1*, dan akurasi. Secara umum, dari ketiga model tersebut tidak terdapat model yang unggul mutlak terhadap model lainnya. Ditinjau dari mean dan variansi keempat metrik, model MobileNet V2 memberikan mean tertinggi dengan variansi terendah. Dalam hal waktu training, model model Inception V3 membutuhkan waktu paling singkat.

Untuk kelanjutan penelitian, dapat dilakukan *fine tuning* hiperparameter untuk perbaikan kinerja model, ataupun menerapkan model pralatih lainnya.

## DAFTAR PUSTAKA

- [1] R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Ann Eugen*, vol. 7, no. 2, pp. 179–188, Sep. 1936, doi: 10.1111/j.1469-1809.1936.tb02137.x.
- [2] J. P. Pinto, S. Kelur, and J. Shetty, "Iris Flower Species Identification Using Machine Learning Approach," in *2018 4th International Conference for Convergence in Technology (I2CT)*, IEEE, Oct. 2018, pp. 1–4. doi: 10.1109/I2CT42659.2018.9057891.
- [3] A. Eldem, H. Eldem, and D. Ustun, "A Model of Deep Neural Network for Iris Classification With Different Activation Functions," in *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, Sep. 2018, pp. 1–4. doi: 10.1109/IDAP.2018.8620866.
- [4] A. Shukla, A. Agarwal, M. Lucknow, I. H. Pant, and P. Mishra, "Flower Classification using Supervised Learning," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 5, pp. 757–762, 2020.
- [5] H. Hiary, H. Saadeh, M. Saadeh, and M. Yaqub, "Flower classification using deep convolutional neural networks," *IET Computer Vision*, vol. 12, no. 6, pp.

- 855–862, Sep. 2018, doi: 10.1049/iet-cvi.2017.0155.
- [6] M.-E. Nilsback and A. Zisserman, “Automated Flower Classification over a Large Number of Classes,” in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, IEEE, Dec. 2008, pp. 722–729. doi: 10.1109/ICVGIP.2008.47.
- [7] M.-E. Nilsback and A. Zisserman, “A Visual Vocabulary for Flower Classification,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR’06)*, IEEE, 2006, pp. 1447–1454. doi: 10.1109/CVPR.2006.42.
- [8] Jie Zou and G. Nagy, “Evaluation of model-based interactive flower recognition,” in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, IEEE, 2004, pp. 311–314 Vol.2. doi: 10.1109/ICPR.2004.1334185.
- [9] S. M. Omer, R. M. Hasan, and B. N. Anwer, “An Image Dataset Construction for Flower Recognition Using Convolutional Neural Network,” *Science Journal of University of Zakho*, vol. 8, no. 3, pp. 112–117, Sep. 2020, doi: 10.25271/sjuoz.2020.8.3.747.
- [10] N. FatihahSahidan, A. K. Juha, N. Mohammad, and Z. Ibrahim, “Flower and leaf recognition for plant identification using convolutional neural network,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 2, pp. 737–743, 2019, doi: 10.11591/ijeecs.v16.i2.pp737-743.
- [11] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *3rd International Conference on Learning Representations (ICLR) 2015*, Sep. 2015, pp. 1–14. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.
- [13] C. Szegedy *et al.*, “Going Deeper with Convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.
- [15] “Iris Computer Vision.” <https://www.kaggle.com/datasets/jeffheaton/iris-computer-vision> (accessed Jul. 05, 2023).