1. **Latar Belakang**

Kemajuan teknologi kecerdasan buatan, khususnya dalam bidang *Computer Vision*, telah membuka peluang besar untuk mengembangkan sistem pengenalan objek yang andal. Salah satu pendekatan yang efektif dalam pengenalan gambar adalah *Convolutional Neural Networks (CNN)*, jaringan saraf tiruan yang dirancang untuk mengenali pola visual secara otomatis. CNN telah digunakan secara luas dalam berbagai aplikasi, mulai dari pengenalan wajah hingga klasifikasi objek di industri. Dalam proyek ini, CNN diterapkan untuk mengenali dan mengklasifikasikan 150 jenis Pokémon berdasarkan gambar, menjadikannya studi kasus yang menarik dan menantang.

Klasifikasi Pokémon menggunakan implementasi CNN memiliki beberapa keunggulan. Setiap Pokémon memiliki karakteristik visual yang unik, seperti bentuk tubuh, warna, dan pola, yang mencerminkan kompleksitas visual tingkat tinggi. Selain itu, dengan jumlah kelas yang besar, yaitu 150 jenis Pokémon, proyek ini termasuk dalam kategori klasifikasi multi-kelas, yang merupakan tantangan signifikan dalam *Machine Learning*. Variasi data, seperti sudut pandang, pencahayaan, dan posisi objek dalam gambar, juga memberikan peluang untuk menguji kemampuan model dalam menangani perbedaan konteks visual. Lebih jauh lagi, tantangan ini mencerminkan masalah dunia nyata, seperti pengenalan wajah atau klasifikasi objek, menjadikannya relevan tidak hanya secara akademik tetapi juga praktis.

Proyek ini bertujuan untuk mengembangkan model yang mampu mengidentifikasi Pokémon secara akurat dengan memanfaatkan teknik modern seperti augmentasi data, optimasi dengan algoritma Adam, dan fungsi loss categorical crossentropy. Dengan pendekatan ini, proyek ini diharapkan tidak hanya memberikan solusi untuk pengenalan Pokémon tetapi juga memperlihatkan potensi CNN dalam menangani masalah klasifikasi multi-kelas yang kompleks. *Showcase* ini juga diharapkan dapat menjadi inspirasi untuk aplikasi serupa di bidang lain.

1. **Teknik Machine Learning yang Digunakan**

Beberapa teknik machine learning yang digunakan dalam progam ini adalah,

1. Convolutional Neural Network

CNN adalah jenis arsitektur jaringan saraf tiruan yang dirancang khusus untuk memproses data berbentuk grid, seperti gambar. CNN unggul dalam menangkap pola spasial dan hubungan hierarkis antar fitur dalam data gambar.

1. **Struktur CNN**

CNN terdiri dari beberapa lapisan utama:

1. **Convolutional Layer**:
   * Lapisan ini bertugas mengekstrak fitur dari gambar input melalui operasi **konvolusi**.
   * Operasi konvolusi melibatkan penerapan **kernel/filter** kecil (matriks) pada gambar untuk mendeteksi pola seperti tepi, garis, atau tekstur.
   * **Contoh penggunaan pada program**:

Conv2D(32, (3, 3), *activation*='relu')

* + - **32**: Jumlah filter yang digunakan (semakin banyak, semakin kompleks fitur yang dapat ditangkap).
    - **(3, 3)**: Ukuran kernel filter (biasanya kecil, seperti 3x3 atau 5x5).
    - **ReLU**: Fungsi aktivasi yang membuat semua nilai negatif menjadi nol untuk menjaga non-linearitas.

1. **Pooling Layer**:
   * Lapisan pooling bertujuan untuk mengurangi dimensi fitur yang diekstraksi oleh lapisan konvolusi, menjaga fitur yang paling penting.
   * **MaxPooling** adalah jenis pooling yang paling umum digunakan, memilih nilai maksimum dalam area tertentu dari fitur.
   * **Contoh penggunaan pada program:**

MaxPooling2D(2, 2)

* + - **(2, 2)**: Ukuran pooling yang mengurangi dimensi fitur menjadi setengah dari ukuran aslinya.

1. **Flatten Layer**:
   * Setelah fitur penting diekstraksi, data berbentuk matriks (grid) perlu diubah menjadi vektor satu dimensi agar dapat digunakan oleh lapisan Dense.
   * **Contoh penggunaan pada program:**

Flatten()

1. **Fully Connected Layer (Dense Layer)**:
   * Lapisan Dense adalah bagian dari jaringan saraf feedforward yang menerima data vektor dan melakukan klasifikasi berdasarkan pola yang dipelajari.
   * **Contoh penggunaan pada program:**

Dense(512, *activation*='relu'),

Dense(num\_classes, *activation*='softmax')

* + - Lapisan pertama memiliki 512 neuron dengan aktivasi ReLU.
    - Lapisan kedua adalah lapisan output dengan jumlah neuron sesuai jumlah kelas (150 Pokémon), menggunakan aktivasi **softmax** untuk memberikan probabilitas klasifikasi.

1. **Fungsi dalam Program yang Dibuat**

Dalam proyek klasifikasi Pokémon, CNN digunakan untuk mengenali berbagai fitur unik dari gambar Pokémon, seperti:

* Bentuk: Pikachu memiliki tubuh bulat kecil, sementara Charizard memiliki bentuk tubuh besar dengan sayap.
* Warna: Setiap Pokémon memiliki kombinasi warna unik.
* Tekstur/Pola: Beberapa Pokémon memiliki garis atau tanda khusus yang mudah dikenali.

1. **Tahapan Pelatihan CNN dalam Program**
2. **Penerapan konvolusi dan Pooling:**
   * CNN menerapkan beberapa lapisan konvolusi dan pooling untuk mengekstraksi fitur dari gambar Pokémon dengan resolusi 150x150.
   * Implementasinya sebagai berikut:

model = Sequential([

    Input(*shape*=(150, 150, 3)),  *# Input layer untuk gambar RGB ukuran 150x150*

    Conv2D(32, (3, 3), *activation*='relu'),

    MaxPooling2D(2, 2),

    Conv2D(64, (3, 3), *activation*='relu'),

    MaxPooling2D(2, 2),

    Conv2D(128, (3, 3), *activation*='relu'),

    MaxPooling2D(2, 2),

    Flatten(),

    Dense(512, *activation*='relu'),

    Dropout(0.5),  *# Regularisasi*

    Dense(num\_classes, *activation*='softmax')

])

1. **Pelatihan dengan Backpropagation**
   * Model dilatih menggunakan **backpropagation**, di mana CNN memperbarui bobot filter dan parameter lain berdasarkan gradien yang dihitung dari loss.
2. **Prediksi Kelas**
   * Setelah pelatihan selesai, model memanfaatkan lapisan fully connected untuk memberikan prediksi akhir berupa probabilitas kelas.
3. Data Augmentation

Data augmentation adalah teknik untuk memperbesar jumlah dan variasi data pelatihan secara artifisial. Teknik ini menciptakan variasi dari gambar yang ada, seperti memutar, memperbesar, atau memindahkan gambar. Penerapannya dalam program adalah sebagai berikut,

train\_datagen = ImageDataGenerator(

*rescale*=1./255,

*rotation\_range*=20,

*width\_shift\_range*=0.2,

*height\_shift\_range*=0.2,

*shear\_range*=0.2,

*zoom\_range*=0.2,

*horizontal\_flip*=True,

*fill\_mode*='nearest'

)

 **rotation\_range=20**: Memutar gambar hingga 20 derajat.

 **width\_shift\_range=0.2, height\_shift\_range=0.2**: Menggeser gambar secara horizontal/vertikal sebesar 20%.

 **shear\_range=0.2**: Melakukan transformasi miring pada gambar.

 **zoom\_range=0.2**: Memperbesar atau memperkecil gambar hingga 20%.

 **horizontal\_flip=True**: Membalik gambar secara horizontal.

Fungsinya dalam program adalah Augmentasi data membuat model lebih robust terhadap variasi gambar di dunia nyata, seperti rotasi atau pencahayaan yang berbeda, sehingga mengurangi risiko overfitting.

1. Normalization

Normalisasi adalah proses mengubah nilai data ke rentang tertentu, biasanya [0, 1] dalam kasus gambar. Ini dilakukan dengan membagi setiap nilai piksel (0–255) dengan 255. Fungsinya pada program ini adalah untuk memastikan nilai input berada dalam skala yang seragam, yang penting untuk menghindari gradien besar dan mempercepat proses pelatihan.

1. Categorical Crossentropy

Categorical crossentropy adalah fungsi loss yang digunakan untuk masalah klasifikasi multi-kelas. Fungsi ini menghitung jarak antara distribusi prediksi (output softmax) dengan distribusi label sebenarnya (one-hot encoded). Fungsinya pada project ini adalah membantu model untuk belajar membuat prediksi yang mendekati distribusi probabilitas dari label target.

1. Softmax Output Layer

Softmax adalah fungsi aktivasi yang digunakan di layer output untuk menghasilkan distribusi probabilitas. Setiap kelas mendapatkan probabilitas antara 0 dan 1, dan total probabilitas semua kelas adalah 1. Fungsinya pada program ini adalah memungkinkan model memutuskan Pokémon mana yang paling mungkin berdasarkan gambar input, dengan memberikan skor probabilitas untuk setiap kelas.

1. **Hasil Eksperimen**
   1. Evaluasi Performa Model

Proses pelatihan model dilakukan selama 25 epoch dengan menggunakan dataset yang terbagi menjadi data pelatihan dan validasi. Model dilatih menggunakan optimasi Adam dan fungsi loss categorical crossentropy untuk meminimalkan kesalahan prediksi pada 150 kelas Pokémon. Evaluasi dilakukan berdasarkan metrik akurasi dan loss pada data pelatihan serta validasi.

Berikut gambaran pelatihan selama 25 epoch pada terminal.A screenshot of a computer

Description automatically generated

Dari pelatihan tersebut didapat grafik *validation accuracy and validation loss* sebagai berikut,

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated

Grafik akurasi menunjukkan peningkatan bertahap selama pelatihan, dengan akurasi validasi stabil pada nilai tinggi sebelum mencapai titik konvergensi. Grafik loss menunjukkan penurunan signifikan pada awal pelatihan, dengan nilai loss validasi mulai stabil setelah beberapa epoch. Performa dapat ditingkatkan lagi dengan menaikkan banyak epoch, hanya saja pelatihannya akan menjadi lebih lama. Pelatihan dengan 25 epoch ini memakan waktu selama 15 menit.

**Hasil evaluasi akhir:**

Akurasi Pelatihan: 51%

Akurasi Validasi: 54%

Loss Pelatihan: 1.72

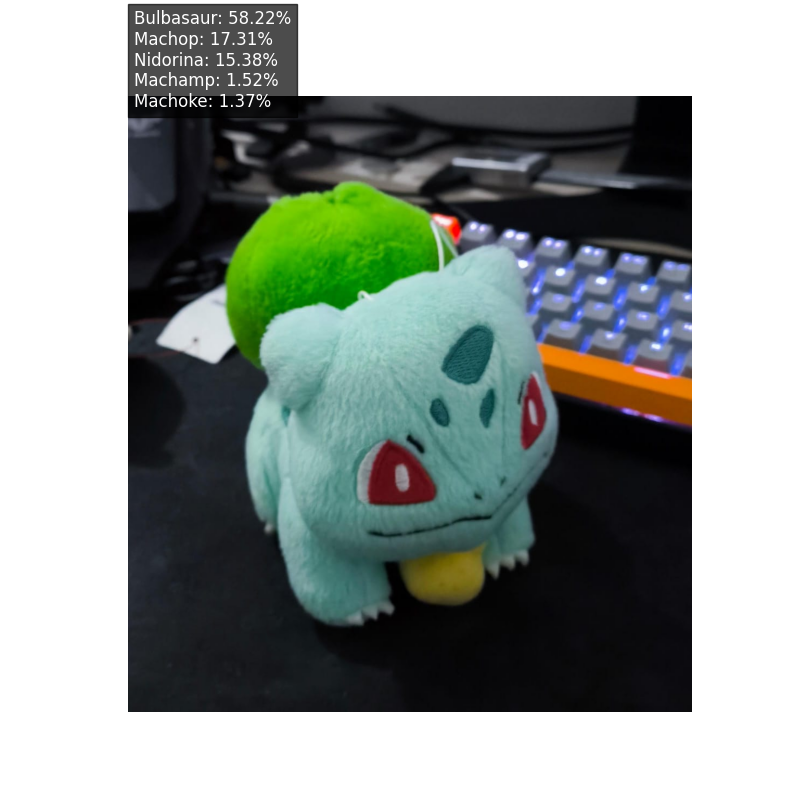
Loss Validasi: 1.70

* 1. Pengujian Data Baru

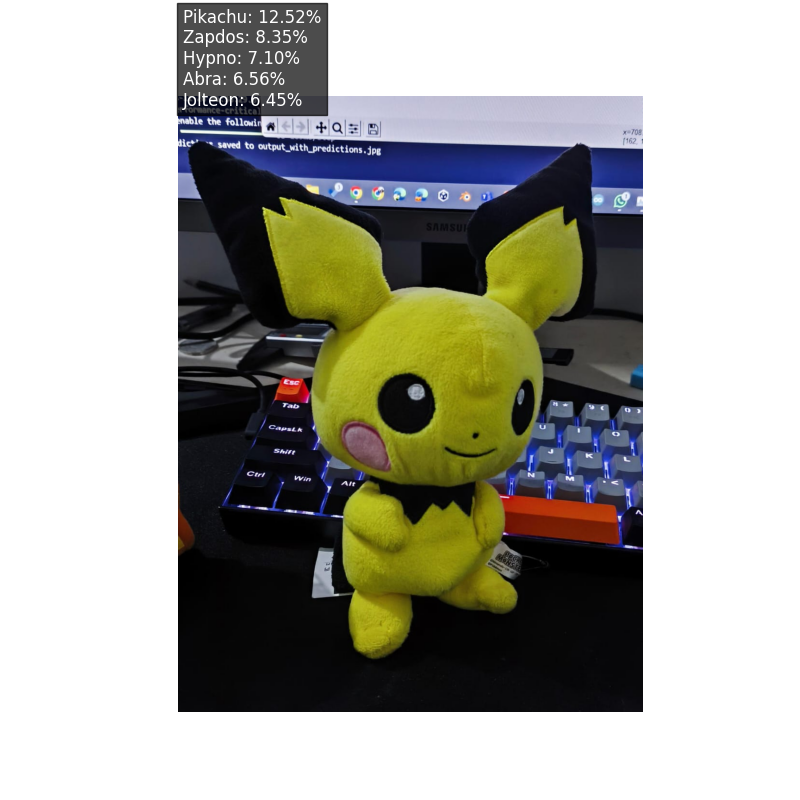
Model diuji dengan data gambar baru yang tidak terdapat dalam dataset pelatihan. Pengujian ini bertujuan untuk mengukur kemampuan generalisasi model dalam mengklasifikasikan Pokémon.

Hasil pengujian menunjukkan bahwa model mampu memberikan prediksi dengan tingkat akurasi yang cukup. Sebagai contoh:

1. Foto Bulbasaur yang diambil langsung oleh penulis dari koleksi plushie Pokémon penulis:



1. Foto Pichu yang juga diambil langsung oleh penulis. Pichu dipilih walau tidak ada di dataset untuk menguji kemampuan model dalam mengenali kemiripan Pokémon satu dengan yang lain. Hal ini dipilih karena Pichu merupakan Pokémon yang berevolusi menjadi Pikachu. Sehingga kita dapat melihat apakah model dapat mengenali kemiripan Pokémon sebelum dan setelah evolusi.

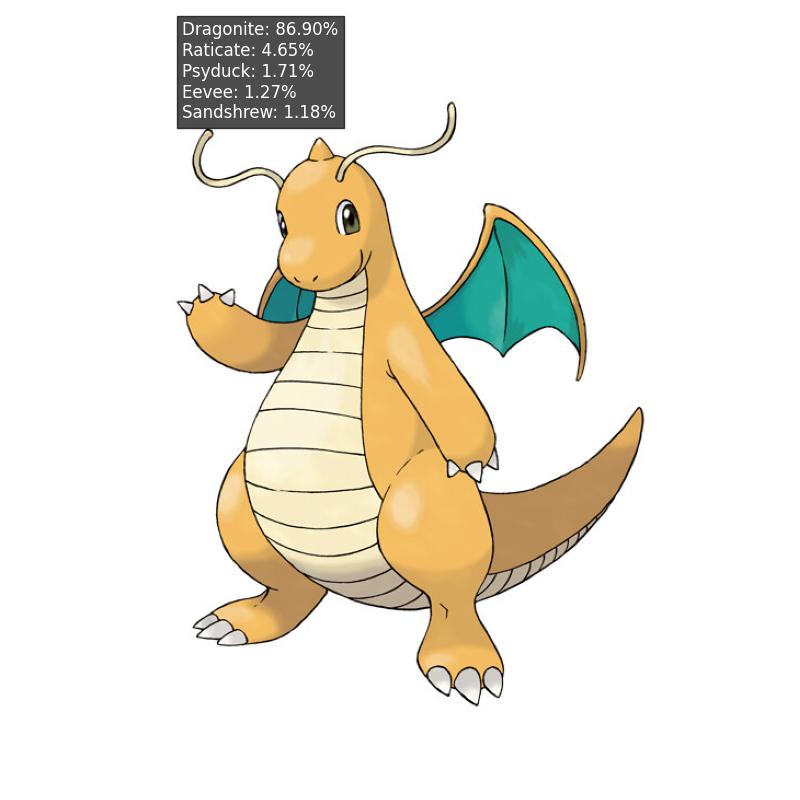
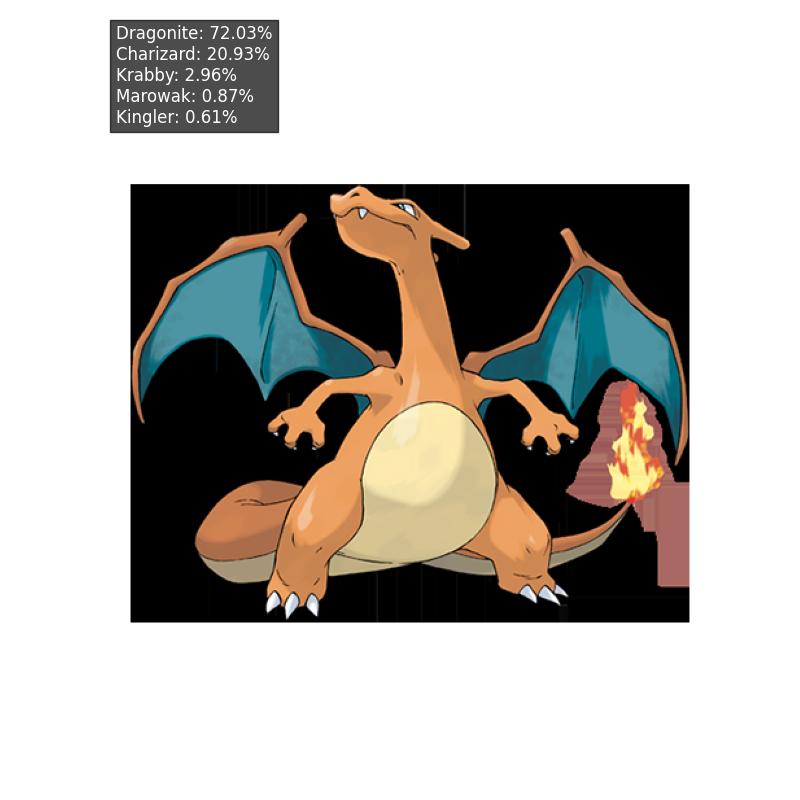


1. Gambar Charizard outline, hal ini ditujukan untuk menguji kemampuan model untuk mengenali Pokemon dari outline nya saja tanpa ada indikasi warna. Hasilnya model tidak dapat melakukan prediksi yang benar hanya pada outlinenya saja.

A purple line drawing of a cartoon dragon

Description automatically generated

1. Gambar Charizard dan Dragonite. Dua Pokemon ini dipilih secara khusus karena secara anatomis dan warnanya tergolong mirip sehingga dapat digunakan sebagai data pengujian apakah model dapat membedakan antara kedua jenis Pokemon dengan benar.



Dari pengujian tersebut dapat dilihat jelas bahwa model tidak dapat mengklasifikasikan Charizard dengan benar dan malah mengklasifikasikannya sebagai dragonite. Hal ini menjadi catatan bahwa untuk 25 epoch pelatihan, model belum cukup untuk membedakan pokemon yang mirip.

1. Gambar Eevee dengan aksesories digunakan untuk menguji performa model dalam mengklasifikasi pokemon apabila pokemon menggunakan aksesories tertentu. Hasilnya model masih dapat mengklasifikasikan Eeevee dengan aksesories bunga di kepala dengan benar.

A cartoon animal with flowers in hair

Description automatically generated

1. Gambar Vulpix dengan style 3D ditujukan untuk menguji model dalam mengenali Pokemon dengan art style yang berbeda dari biasanya. Hasilnya model masih dapat mengenali Pokemon dengan benar.

A cartoon animal with a tail

Description automatically generated

1. Gambar Pidgey dan Spearow ditujukan untuk menguji keandalan model dalam mengklasifikasikan Pokemon yang juga mirip dalam segi anatomisnya karena kedua Pokemon merupakan tipe flying yang berbentuk seperti burung.

Cartoon bird with wings spread

Description automatically generated A cartoon of a bird

Description automatically generated

Hasil menunjukkan bahwa dengan epoch 25, model sudah dapat membedakan Pidgey dan Spearow yang keduanya tergolong mirip secara anatomis.

1. **Flow Diagram/Pipeline**

Berikut flow diagram/pipeline dari program yang dibuat secara *end-to-end.*

A diagram of a diagram

Description automatically generated

**1. Input Dataset**

* **Deskripsi**: Langkah pertama adalah mempersiapkan dataset yang terdiri dari gambar Pokémon yang digunakan untuk pelatihan dan validasi.
* **Proses**:
  + Gambar-gambar Pokémon disusun dalam folder berdasarkan kelas (tiap Pokémon).
  + Dataset dibagi menjadi dua bagian: pelatihan dan validasi.

**2. Preprocessing**

* **Deskripsi**: Setelah dataset disiapkan, gambar perlu diproses agar sesuai dengan format input model.
* **Proses**:
  + **Normalisasi**: Semua gambar diubah ke rentang nilai [0, 1] dengan membagi nilai pixel dengan 255.
  + **Data Augmentation**: Variasi gambar dilakukan melalui transformasi seperti rotasi, pemotongan, zoom, flipping, dll. Hal ini bertujuan untuk meningkatkan keragaman data pelatihan dan mengurangi overfitting.

**3. Model Architecture (CNN)**

* **Deskripsi**: Pada tahap ini, arsitektur CNN dibangun, yang terdiri dari beberapa lapisan untuk ekstraksi fitur dan klasifikasi.
* **Proses**:
  + **Conv2D**: Lapisan konvolusi untuk mengekstraksi fitur seperti tepi dan bentuk.
  + **MaxPooling2D**: Mengurangi dimensi fitur dengan memilih nilai maksimum dalam area tertentu.
  + **Flatten**: Mengubah output dari lapisan konvolusi menjadi vektor satu dimensi.
  + **Dense**: Lapisan sepenuhnya terhubung untuk membuat prediksi berdasarkan fitur yang telah dipelajari.
  + **Softmax**: Pada lapisan output, fungsi aktivasi softmax digunakan untuk menghasilkan probabilitas untuk setiap kelas Pokémon.

**4. Training Process**

* **Deskripsi**: Pada tahap ini, model dilatih menggunakan data pelatihan dan dievaluasi menggunakan data validasi.
* **Proses**:
  + **Loss Function**: Fungsi loss **categorical crossentropy** digunakan untuk menghitung kesalahan prediksi antara hasil yang diprediksi dan label asli.
  + **Optimizer**: Algoritma **Adam** digunakan untuk memperbarui bobot model berdasarkan gradien.
  + **Metrics**: Metrik **accuracy** digunakan untuk memantau performa model selama pelatihan.

**5. Model Evaluation**

* **Deskripsi**: Setelah pelatihan selesai, model dievaluasi menggunakan data validasi yang belum dilihat sebelumnya.
* **Proses**:
  + Menghitung akurasi dan loss pada data validasi untuk mengukur kemampuan model dalam menggeneralisasi data yang belum pernah dilihat.
  + Grafik pelatihan dan validasi akurasi/loss dapat divisualisasikan untuk menunjukkan seberapa baik model bekerja.

**6. Model Deployment**

* **Deskripsi**: Setelah model dilatih dan diuji, model disimpan dan dapat digunakan untuk prediksi pada gambar baru.
* **Proses**:
  + **Model Saving**: Model disimpan dalam format .keras untuk digunakan kembali di masa depan.
  + **Model Loading**: Model yang sudah dilatih kemudian dimuat kembali untuk digunakan pada data uji baru.

**7. Prediction**

* **Deskripsi**: Setelah model siap, langkah berikutnya adalah menggunakan model untuk memprediksi gambar Pokémon baru.
* **Proses**:
  + **Preprocessing Image**: Gambar baru diproses dengan cara yang sama seperti data pelatihan (normalisasi dan resizing).
  + **Prediction**: Model melakukan prediksi dengan mengeluarkan probabilitas untuk setiap kelas.
  + **Top-5 Predictions**: Model dapat menghasilkan lima prediksi teratas beserta probabilitasnya.

**8. Output**

* **Deskripsi**: Setelah model memberikan prediksi, hasil tersebut ditampilkan kepada pengguna.
* **Proses**:
  + Menampilkan gambar dengan label prediksi dan probabilitas kelas teratas.
  + Menyimpan gambar dengan overlay prediksi sebagai output.