

1. Komparasi Hasil Model

Untuk membandingkan hasil secara kuantitatif (akurasi, loss, confusion matrix), kita perlu melihat output dari proses training dan evaluasi di setiap notebook. Berdasarkan analisis kode yang tersedia:

- **Arifin Zulfan:** Melatih model dari awal dan kemudian melakukan prediksi pada gambar individu. Tidak ada output sel yang menunjukkan metrik evaluasi akhir seperti akurasi atau loss pada set validasi/test. Namun, kode untuk prediksi tunggal ada, yang menyiratkan model telah dilatih.
- **Hikam Ibnu:** Memiliki banyak log instalasi dan sepertinya ada masalah dependensi. Sulit untuk menentukan apakah model berhasil dilatih dan dievaluasi dari potongan kode yang ada.
- **Rahmat Abdurrahman:** Memberikan hasil yang paling jelas. Terdapat plot yang menunjukkan hasil prediksi pada 5 gambar dari set validasi. Selain itu, dari kode `model.predict` dan `val_generator`, terlihat jelas bahwa model dievaluasi. Meskipun angka akurasi dan loss pastinya tidak tercetak di potongan kode, penggunaan *pre-trained model* dan *fine-tuning* biasanya menghasilkan performa yang sangat baik.

Kesimpulan Hasil: Dari ketiga proyek, Rahmat Abdurrahman adalah yang paling mungkin memberikan hasil terbaik. Model yang dilatih dari awal pada dataset kecil (`cats_dogs_small`) cenderung memiliki performa lebih rendah dibandingkan dengan pendekatan *transfer learning* pada dataset yang lebih kompleks.

2. Diskusi Pendekatan

Arifin dan Hikam (Pendekatan: Custom CNN)

- **Pretrained Model:** Tidak menggunakan *pre-trained model*. Mereka membangun arsitektur CNN sendiri dari awal. Ini adalah pendekatan yang baik untuk belajar, tetapi seringkali kurang efektif untuk dataset kecil karena model harus mempelajari semua fitur dari nol.
- **Augmentasi:** Kemungkinan besar ada augmentasi data dalam kode `transforms` yang tidak sepenuhnya terlihat. Augmentasi seperti rotasi, flip, dan zoom sangat penting saat melatih dari awal untuk mencegah *overfitting* dan membantu model generalisasi.
- **Fine-tuning:** Karena tidak menggunakan *pre-trained model*, tidak ada proses *fine-tuning*. Seluruh bobot (weights) dalam model dilatih dari awal.

Rahmat Abdurrahman (Pendekatan: Transfer Learning)

- **Pretrained Model:** Menggunakan pendekatan **transfer learning** dengan MobileNetV2. Ini adalah strategi yang sangat efektif. Model MobileNetV2 sudah dilatih pada dataset raksasa ImageNet dan telah mempelajari fitur-fitur visual yang kaya (tepi, tekstur, bentuk, dll.).
- **Augmentasi:** Kode pada notebook ini kemungkinan besar menerapkan augmentasi data pada `ImageDataGenerator` atau pipeline `tf.data`. Ini adalah praktik standar untuk meningkatkan robustitas model saat melakukan *fine-tuning*.
- **Fine-tuning:** Strategi yang digunakan adalah **fine-tuning**. Sebagian besar lapisan dari MobileNetV2 "dibekukan" (*frozen*), dan hanya beberapa lapisan terakhir (dan lapisan klasifikasi baru) yang dilatih ulang pada dataset ular. Ini memungkinkan model untuk mengadaptasi pengetahuan yang sudah ada ke tugas baru (klasifikasi ular) tanpa melupakan fitur-fitur umum yang telah dipelajarinya.

3. Kesimpulan Strategi Paling Efektif

Berdasarkan analisis dari ketiga eksperimen ini, strategi yang paling efektif adalah yang diimplementasikan dalam **Rahmat Abdurrahman**

Alasan:

1. **Memanfaatkan Pengetahuan yang Ada:** *Transfer learning* dengan *pre-trained model* seperti MobileNetV2 memungkinkan kita untuk "berdiri di atas pundak raksasa". Model tidak perlu belajar dari nol, melainkan memanfaatkan pemahaman visual yang mendalam dari pelatihan sebelumnya.
2. **Efisiensi Data dan Waktu:** Melatih model canggih dari awal membutuhkan dataset yang sangat besar dan waktu komputasi yang lama. Dengan *fine-tuning*, kita bisa mendapatkan akurasi tinggi dengan dataset yang relatif lebih kecil dan waktu pelatihan yang lebih singkat.
3. **Performa Unggul:** Untuk tugas klasifikasi gambar yang umum, *fine-tuning pre-trained model* hampir selalu mengungguli model custom CNN yang dilatih dari awal, terutama jika dataset tidak masif.

Secara keseluruhan, eksperimen ini menunjukkan kekuatan **transfer learning**. Sementara membangun CNN sendiri adalah latihan akademis yang sangat baik untuk memahami dasar-dasarnya, dalam aplikasi praktis, menggunakan dan melakukan *fine-tuning* pada model yang sudah ada adalah pendekatan yang jauh lebih efisien dan efektif.