

Kelompok 4

Adam Toyib Nur Wahid / 442023611073

Fayshal Karan Athilla / 442023611088

Andrian Maulana / 442023611059

Ahmad Irhan Fauzi / 442023611061

Daffa Rafly Muhammad / 442023611063

Muhammad Dwi Apriansyah / 442023611065

Analisis Perbandingan Proyek Transfer Learning Mahasiswa

Ringkasan Proyek Keseluruhan

1. Fayshal Karan Athilla (UNIDA GONTOR)

Proyek: Klasifikasi dua jenis objek: apel dan pisang[cite: 1, 4].

Dataset: Menggunakan 'freshapples' dan 'rottenapples' sebagai "apel", dan 'freshbanana' dan 'rottenbanana' sebagai "pisang" dari dataset yang lebih besar berisi 6 kelas[cite: 5].

Set Pelatihan: 868 gambar[cite: 5].

Set Validasi: 216 gambar[cite: 5].

Model & Pendekatan: MobileNetV2 (tanpa top layer) sebagai ekstraktor fitur, dengan GlobalAveragePooling2D dan sebuah Dense layer (2 neuron, softmax)[cite: 5].

Pelatihan: 10 epoch, optimizer Adam, loss function categorical_crossentropy[cite: 5].

2. Andrian Maulana (Universitas Darussalam Gontor)

Proyek: Perbandingan tiga model transfer learning untuk klasifikasi dua jenis objek (contoh: keledai vs kuda; aktual dalam laporan: singa vs citah)[cite: 13, 15, 16].

Dataset: Dua kelas (singa dan citah), minimal total 200 gambar (100 per kelas)[cite: 16].

Pembagian: 70% data pelatihan, 30% data validasi[cite: 17].

Augmentasi: Rotasi dasar dan flipping[cite: 17].

Model & Pendekatan: Membandingkan VGG16, ResNet18, dan MobileNetV2[cite: 14, 19].

Pra-pemrosesan: Ubah ukuran ke 224×224 , normalisasi[cite: 18].

Finetuning: Membekukan sebagian besar layer dan hanya melatih classifier (layer terakhir)[cite: 19].

Pelatihan: 10 epoch, optimizer Adam, CrossEntropyLoss[cite: 19, 20].

3. Adam Toyib Nur Wahid (Universitas tidak disebutkan secara eksplisit, namun format laporan mirip UNIDA)

Proyek: Deteksi masker wajah (with_mask vs without_mask)[cite: 36].

Dataset: "Face Mask Detection Dataset" dari Kaggle[cite: 45].

Kelas: 'with_mask', 'without_mask'[cite: 46, 55].

Pembagian: 80% data latih, 20% data uji[cite: 51].

Ukuran batch: 32[cite: 52].

Model & Pendekatan: ResNet18 dengan bobot praterlatih ImageNet[cite: 37, 53, 54].

Prapemrosesan: Ubah ukuran ke 224×224 , konversi ToTensor, normalisasi[cite: 48, 49, 50].

Finetuning: Mengganti fully connected layer terakhir (model.fc) dengan layer `nn.Linear` baru dengan 2 fitur output[cite: 55]. Semua 11,18 juta parameter dapat dilatih (trainable)[cite: 56].

Pelatihan: 10 epoch, optimizer Adam (learning rate $1e4$), `nn.CrossEntropyLoss`[cite: 58, 59].

Perbandingan Hasil Model

Fitur	Fayshal K. A. (Apel/Pisang)	Andrian M. (Singa/Citah)	Adam T. N. W. (Deteksi Masker)
Model Pra-terlatih	MobileNetV2	VGG16, ResNet18, MobileNetV2	ResNet18
Epoch	10	10	10
Akurasi Latih Akhir	~100% (<i>epoch</i> ke-9)	VGG16: 1.0000 ResNet18: 1.0000 MobileNetV2: 1.0000	Tidak disebutkan secara eksplisit, namun <i>Train Loss</i> adalah 0.0003 pada <i>epoch</i> ke-10
Akurasi Val/Uji Akhir	~100% (<i>epoch</i> ke-9)	VGG16: 0.9250 ResNet18: 0.9000 MobileNetV2: 0.9000	0.9992 (Akurasi Uji)
Loss Latih Akhir	Sangat rendah (<i>epoch</i> ke-9)	Tidak disebutkan secara eksplisit untuk masing-masing, namun grafik disediakan	0.0003 (<i>epoch</i> ke-10)
Loss Val/Uji Akhir	Sangat rendah (<i>epoch</i> ke-9)	Tidak disebutkan secara eksplisit untuk masing-masing, namun grafik disediakan	0.0009 (<i>Loss</i> Uji pada <i>epoch</i> ke-10)
Confusion Matrix	Disediakan. Apel diprediksi pisang: 51, Pisang diprediksi apel: 51	Tidak disediakan.	Tidak disediakan, namun akurasi tinggi menunjukkan perbedaan yang baik
Overfitting	Minimal, latih/val konvergen dengan baik.	VGG16 menunjukkan sedikit <i>overfitting</i> . ResNet18 & MobileNetV2 stabil	

Diskusi Pendekatan

Augmentasi:

Fayshal: Tidak secara eksplisit menyebutkan penggunaan augmentasi tetapi menyarakannya untuk perbaikan di masa depan guna meningkatkan generalisasi[cite: 11].

Andrian: Menggunakan augmentasi dasar (rotasi dan flipping) untuk memperkaya variasi data[cite: 17].

Adam: Tidak secara eksplisit menyebutkan augmentasi data dalam langkahlangkah prapemrosesan[cite: 48, 49, 50].

Pemilihan Model Praterlatih:

Fayshal: Memilih MobileNetV2, yang dikenal karena efisiensinya[cite: 5].

Andrian: Bereksperimen dengan VGG16 (bisa jadi powerful tetapi lebih besar), ResNet18 (keseimbangan yang baik antara akurasi dan efisiensi), dan MobileNetV2 (efisien)[cite: 19]. Pendekatan komparatif ini berharga untuk memahami tradeoff.

Adam: Menggunakan ResNet18, arsitektur yang kuat dan banyak digunakan[cite: 37, 53]. Ia mencatat efektivitasnya untuk tugas tersebut[cite: 67].

Finetuning:

Fayshal: Menggunakan MobileNetV2 sebagai ekstraktor fitur dan menambahkan layer klasifikasi khusus[cite: 5]. Ini adalah teknik transfer learning yang umum dan efektif.

Andrian: Membekukan sebagian besar layer dari model praterlatih dan hanya melatih classifier (layer terakhir)[cite: 19]. Ini cocok ketika dataset baru berukuran kecil atau mirip dengan dataset asli tempat model dilatih.

Adam: Mengganti layer fc terakhir dari ResNet18 dan membuat semua 11,18 juta parameter dapat dilatih[cite: 55, 56]. Meskipun ia menyebutkan "finetuning pada lapisan terakhir"[cite: 68], membuat semua parameter dapat dilatih menunjukkan pendekatan finetuning yang lebih ekstensif daripada hanya classifier. Ini dapat menghasilkan adaptasi yang lebih baik jika dataset cukup besar dan berbeda.

Menarik Kesimpulan Bersama: Strategi Paling Efektif dari Eksperimen

1. Transfer Learning Sangat Efektif: Ketiga proyek berhasil menunjukkan bahwa transfer learning menggunakan model praterlatih seperti MobileNetV2, VGG16, dan ResNet18 dapat mencapai akurasi tinggi dalam tugas klasifikasi citra, bahkan dengan dataset yang mungkin terbatas[cite: 4, 14, 33, 75], dan secara signifikan mempercepat pengembangan.

2. Pemilihan Model Penting:

Perbandingan Andrian menunjukkan bahwa VGG16 dapat mencapai akurasi validasi yang tinggi tetapi rentan terhadap overfitting[cite: 22, 31].

ResNet18 memberikan kinerja yang stabil dan kuat baik untuk Andrian (akurasi validasi 90%) [cite: 25, 26] maupun Adam (akurasi uji 99.92%)[cite: 39, 67].

MobileNetV2 disorot karena efisiensinya oleh Andrian (akurasi validasi 90%, sebanding dengan ResNet18 dalam pengujiannya) [cite: 28, 29, 32] dan merupakan pilihan Fayshal, menghasilkan hasil yang sangat baik[cite: 6, 10].

Pilihan tergantung pada keseimbangan yang dibutuhkan antara akurasi, ukuran model, dan sumber daya komputasi[cite: 34].

3. Strategi Finetuning adalah Kunci:

Mengganti classifier head dan melatihnya (seperti yang dilakukan Andrian dengan membekukan sebagian besar layer) adalah titik awal yang baik dan dapat menghasilkan hasil yang kuat[cite: 19].

Pendekatan Fayshal menggunakan base model sebagai ekstraktor fitur dan menambahkan beberapa layer juga standar dan efektif[cite: 5].

Pendekatan Adam untuk membuat semua layer dapat dilatih (atau setidaknya lebih banyak layer daripada hanya classifier) setelah mengganti head dapat mencapai kinerja yang sangat tinggi, seperti yang terlihat dari akurasinya 99.92%[cite: 39, 64], asalkan dataset dapat mendukungnya dan dilakukan dengan hati-hati untuk menghindari overfitting.

4. Prapemrosesan dan Augmentasi Data:

Prapemrosesan standar (ubah ukuran, normalisasi) adalah umum dan perlu[cite: 18, 48, 49, 50].

Penggunaan augmentasi dasar oleh Andrian kemungkinan berkontribusi pada kekokohan modelnya[cite: 17]. Fayshal mengidentifikasi kurangnya augmentasi sebagai area potensial untuk perbaikan, terutama mengingat hasil confusion matrix[cite: 11].

5. Evaluasi Melampaui Akurasi:

Laporan Fayshal, meskipun mencapai akurasi yang mendekati sempurna, menyertakan confusion matrix[cite: 7, 9]. Ini mengungkapkan bahwa model mengalami beberapa kesulitan dalam membedakan antara apel dan pisang (kesalahan klasifikasi yang sama untuk keduanya)[cite: 9]. Ini adalah wawasan penting, menunjukkan bahwa akurasi tinggi tidak selalu berarti klasifikasi sempurna untuk semua instance dan dapat menunjukkan area untuk perbaikan (misalnya, augmentasi yang lebih bertarget, penyesuaian arsitektur)[cite: 11].

Laporan Adam dan Andrian lebih fokus pada kurva akurasi dan loss[cite: 23, 27, 30, 66]. Sementara kurva loss Adam menunjukkan konvergensi yang sangat baik tanpa overfitting yang signifikan[cite: 66], confusion matrix masih dapat menawarkan wawasan yang lebih dalam tentang kesalahan 0.08%.

Singkatnya, strategi paling efektif yang muncul dari eksperimen ini adalah:

Memanfaatkan arsitektur praterlatih yang sudah mapan (ResNet18 dan MobileNetV2 menunjukkan keseimbangan kinerja dan efisiensi yang baik).

Menerapkan strategi finetuning yang jelas, dimulai dengan melatih hanya classifier dan berpotensi membuka lebih banyak layer jika dataset memungkinkan dan kinerja lebih lanjut diperlukan.

Menggunakan augmentasi data untuk meningkatkan generalisasi dan kekokohan, terutama jika hasil awal menunjukkan kebingungan antar kelas.

Memanfaatkan metrik evaluasi di luar akurasi keseluruhan, seperti confusion matrix dan kinerja per kelas, untuk memahami kelemahan model dan memandu perbaikan.