



Nama: **Eden Wijaya (122140187), Intan Permata Sari (122140207),
Bayu Ega Ferdana (122140129)**
Mata Kuliah: **Deep Learning (IF25-40401)**

Tugas Ke: **3**
Tanggal: 3 Oktober 2025

1 Pendahuluan

Residual Network (ResNet) [?] mengatasi *degradation problem* pada jaringan dalam melalui *skip connection* sehingga blok mempelajari fungsi residual $F(x)$ dan mengembalikan $y = F(x) + x$. Tugas ini dilaksanakan dalam tiga tahap: (1) **Plain-34** tanpa *skip* untuk mengobservasi degradasi, (2) **ResNet-34** standar, dan (3) dua modifikasi sesuai ketentuan ToR: **(F)** Dropout pada kepala klasifikasi dan **(G)** penggantian optimizer ke **SGD+Nesterov**. Struktur mengikuti contoh laporan (Pendahuluan, Metodologi, Hasil & Analisis Q1–Q4, Peran AI).

2 Metodologi

2.1 Dataset dan Pra-pemrosesan

Dataset berisi citra RGB **5 kelas makanan Indonesia**. Semua citra diubah ke 224×224 dan dinormalisasi dengan statistik ImageNet (mean [0.485, 0.456, 0.406], std [0.229, 0.224, 0.225]). Augmentasi *train*: **RandomHorizontalFlip(p=0.5)** dan **RandomRotation(15)**. Validasi: *resize + normalisasi*. Data dibagi **80:20** (*train:val*) dengan *stratified split*. **batch_size = 32**.

2.2 Setup Eksperimen

Semua model dilatih **10 epoch** dengan **CrossEntropyLoss**. Optimizer default Tahap 1–2: **Adam** ($lr = 0.001$). Pada Tahap 3, kami menguji dua modifikasi: **(F)** Dropout ($p = 0.5$) pada kepala klasifikasi; **(G)** **SGD+Nesterov** ($lr = 0.01$, $momentum = 0.9$). Pipeline data disetarakan agar perbandingan adil.

Tabel 1: Konfigurasi pelatihan (disetarakan).

Epoch	10	Batch Size	32
Image Size	224×224	Loss	CrossEntropyLoss
Augmentasi (train)	Flip 0.5, Rotasi 15°	Normalisasi	ImageNet mean/std
Optimizer (T1–T2)	Adam ($lr = 0.001$)	Optimizer (G)	SGD+Nesterov ($lr = 0.01, m = 0.9$)
Dropout (F)	$p = 0.5$ (kepala)	Device	cuda/cpu

2.3 Arsitektur Model

Plain-34. Sama kedalaman ResNet-34 ([3, 4, 6, 3]) tetapi *skip* ditiadakan (stack konvolusi murni).

ResNet-34 standar. `torchvision.models.resnet34` dengan `num_classes=5` [?].

Tahap 3F (Dropout). Kepala klasifikasi: `Dropout(0.5) → Linear(5)`.

Tahap 3G (SGD). Model ResNet-34 standar dilatih ulang dengan `SGD(lr=0.01, momentum=0.9, nesterov=True)`.

3 Hasil dan Analisis

Q1: Apa itu Degradation Problem?

Pada jaringan *plain* yang semakin dalam, performa dapat menurun meskipun kapasitas meningkat—bukan karena overfitting semata, tetapi karena optimisasi yang sukar dan gradien melemah. **Plain-34** menunjukkan akurasi validasi lebih rendah dari ResNet-34, selaras dengan teori degradasi.

Q2: Bagaimana Residual Connection Mengatasi Degradasi?

Skip connection menyediakan *gradient highway* sehingga sinyal kesalahan tetap kuat saat *backpropagation*, membuat **ResNet-34** konvergen lebih stabil dan meraih akurasi lebih tinggi dibanding **Plain-34** pada konfigurasi identik.

Q3: Analisis Modifikasi (Dropout dan SGD)

Dropout (F) menambah regularisasi pada kepala klasifikasi; pada horizon 10 epoch performanya masih di bawah baseline ResNet-34 (Adam). **SGD+Nesterov (G)** sedikit melampaui Adam pada validasi terbaik, menunjukkan dinamika generalisasi yang berbeda pada dataset ini.

Q4: Perbandingan Komprehensif

Tabel 2: Perbandingan Akurasi Validasi Terbaik (isi dengan nilai aktual dari log).

Model	Val Acc Terbaik	Catatan
Plain-34	0.5766	degradasi performa
ResNet-34 (Adam)	0.7387	residual learning
ResNet-34 + Dropout (F, Adam)	0.6802	regularisasi kepala
ResNet-34 (SGD+Nesterov) (G)	0.7523	hasil terbaik pada 10 epoch

[width=0.86]Figure/compare_{mods}.png

Gambar 1: Kurva akurasi validasi Plain-34, ResNet-34, Dropout (F), dan SGD (G).

4 Potongan Kode Inti

```

1 model_dropout = models.resnet34(weights=None)
2 num_fts = model_dropout.fc.in_features
3 model_dropout.fc = nn.Sequential(
4     nn.Dropout(p=0.5),
5     nn.Linear(num_fts, 5)
6 )
7 optimizer_dropout = torch.optim.Adam(model_dropout.parameters(), lr=1e-3)
8 # history_dropout = train_model(...); best_dropout = max(history_dropout['val_acc'])

```

Kode 1: Modifikasi (F): kepala klasifikasi dengan Dropout.

```
1 model_sgd = models.resnet34(weights=None)
2 num_fts = model_sgd.fc.in_features
3 model_sgd.fc = nn.Linear(num_fts, 5)
4 optimizer_sgd = torch.optim.SGD(
5     model_sgd.parameters(), lr=1e-2, momentum=0.9, nesterov=True
6 )
7 # history_sgd = train_model(...); best_sgd = max(history_sgd['val_acc'])
```

Kode 2: Modifikasi (G): pelatihan dengan SGD + Nesterov.

5 Peran dan Kontribusi AI Assistant

Prompt/Masalah. Meminta penyusunan laporan LaTeX siap kompil yang mengikuti ToR (tiga tahap; dua modifikasi) dan contoh laporan.

Kontribusi. Memberikan alur pengerjaan tugas, memberikan rekomendasi pilihan, Menyusun struktur, tabel setup (Table 1), tabel hasil (Table 2), serta membantu penyusunan, pembuatan dan penjelasan dari kode tersebut.

Verifikasi. Angka performa diisi dari log pelatihan; narasi disesuaikan dengan dinamika kurva dan gap Plain vs ResNet.

References

<https://chatgpt.com/share/68df66b-793c-8001-a181-093b0f4e224c>