Modifikasi dan Analisis SimCLR pada Dataset Tiny ImageNet

Mohammad Ridho Cahyono
Teknik Informatika
Universitas Darussalam Gontor
Ponorogo, Indonesia
mohammadridhocahyono265@gmail.com

***Abstract*—**Self-Supervised Learning (SSL) menawarkan solusi untuk mengurangi ketergantungan pada dataset berlabel masif dalam melatih model *deep learning*. Penelitian ini mengimplementasikan dan menganalisis metode SSL populer, SimCLR (*A Simple Framework for Contrastive Learning*), pada dataset Tiny ImageNet. Eksperimen dilakukan dengan memodifikasi arsitektur dasar yang disediakan. Modifikasi utama meliputi peningkatan kapasitas *backbone encoder* dari ResNet18 ke ResNet34 dan penambahan strategi augmentasi data dengan *RandomAffine* dan *RandomSolarize*. Kualitas representasi visual yang dipelajari dievaluasi secara kualitatif menggunakan visualisasi t-SNE. Hasil eksperimen menunjukkan bahwa model yang dimodifikasi berhasil mencapai nilai *loss* yang lebih rendah dan menunjukkan pengelompokan (clustering) kelas yang jauh lebih baik pada visualisasi t-SNE dibandingkan dengan model dasar. Hal ini membuktikan bahwa peningkatan kapasitas model dan strategi augmentasi yang lebih kuat secara signifikan meningkatkan kualitas representasi fitur yang dipelajari oleh SimCLR.**

Keywords——Self-Supervised Learning, SimCLR, Contrastive Learning, ResNet, Tiny ImageNet, t-SNE

I. PENDAHULUAN

Model *deep learning*, khususnya *Convolutional Neural Network* (CNN), telah menunjukkan performa luar biasa dalam berbagai tugas visi komputer. Namun, keberhasilan ini sangat bergantung pada ketersediaan dataset berlabel dalam skala besar, yang proses pengumpulannya seringkali mahal dan memakan waktu. *Self-Supervised Learning* (SSL) muncul sebagai paradigma yang menjanjikan untuk mengatasi masalah ini dengan melatih model pada data tidak berlabel.

Salah satu metode SSL yang paling berpengaruh adalah SimCLR (*A Simple Framework for Contrastive Learning of Visual Representations*) [1]. Ide inti dari SimCLR adalah belajar representasi visual dengan memaksimalkan kesamaan (*agreement*) antara dua 'view' dari gambar yang sama yang telah melalui proses augmentasi data yang berbeda, sambil meminimalkan kesamaan dengan 'view' dari gambar lain. Pendekatan ini terbukti mampu menghasilkan *encoder* fitur yang kuat yang kemudian dapat diadaptasi untuk berbagai tugas *downstream* dengan sedikit data berlabel.

Laporan ini menyajikan hasil eksperimen implementasi, modifikasi, dan analisis performa SimCLR. Model dasar dengan *backbone* ResNet18 dilatih pada dataset Tiny ImageNet sebagai *baseline*. Selanjutnya, dilakukan dua modifikasi utama: (1) mengganti *backbone* menjadi ResNet34 untuk meningkatkan kapasitas model, dan (2) menambahkan augmentasi RandomAffine dan RandomSolarize untuk meningkatkan robustisitas fitur. Performa kualitatif dari kedua model dibandingkan melalui visualisasi t-SNE untuk menganalisis kualitas pengelompokan representasi yang dipelajari.

II. METODOLOGI

Eksperimen ini dilakukan menggunakan platform Kaggle Notebook dengan akselerasi GPU

A. Dataset

Dataset yang digunakan adalah **Tiny ImageNet**. Dataset ini merupakan bagian dari dataset ImageNet yang lebih besar, berisi 200 kelas. Setiap kelas memiliki 500 gambar untuk training, 50 untuk validasi, dan 50 untuk pengujian. Ukuran setiap gambar adalah 64x64 piksel. Karena SimCLR adalah metode unsupervised, hanya data training yang digunakan untuk melatih *encoder* tanpa menggunakan labelnya.

B. Arsitektur Dasar SimCLR

Arsitektur dasar SimCLR yang digunakan dalam eksperimen ini terdiri dari empat komponen utama:.

1. ***Data Augmentation*:** Sebuah *pipeline* augmentasi stokastik digunakan untuk menghasilkan dua 'view' yang berbeda dari setiap gambar. Augmentasi dasar meliputi RandomResizedCrop, RandomHorizontalFlip, ColorJitter, RandomGrayscale, dan GaussianBlur.
2. ***Base Encoder*:** Model *baseline* menggunakan arsitektur **ResNet18** sebagai *encoder* untuk mengekstrak vektor representasi dari gambar yang telah di-augmentasi.
3. ***Projection Head*:** Sebuah jaringan *Multi-Layer Perceptron* (MLP) kecil dengan satu lapisan tersembunyi digunakan untuk memetakan vektor representasi ke ruang (space) di mana *contrastive loss* dihitung.
4. ***Contrastive Loss Function*:** Fungsi *loss* yang digunakan adalah **InfoNCE** (*Noise Contrastive Estimation*). Fungsi ini bertujuan untuk menarik representasi dari 'view' yang positif (berasal dari gambar yang sama) dan mendorong representasi dari 'view' yang negatif (berasal dari gambar yang berbeda

III. MODIFIKASI EKSPERIMEN

Dua modifikasi utama diterapkan pada arsitektur dasar untuk dianalisis dampaknya:

A. ***Perubahan Backbone:*** *Backbone encoder* ***ResNet18*** *diganti dengan* ***ResNet34****. Hipotesisnya adalah bahwa jaringan yang lebih dalam memiliki kapasitas yang lebih besar untuk mempelajari fitur yang lebih kompleks dan menghasilkan representasi yang lebih baik.*

B. ***Penambahan Augmentasi:*** *Dua transformasi augmentasi baru ditambahkan ke dalam pipeline:* ***RandomAffine*** *(untuk rotasi dan pergeseran acak) dan* ***RandomSolarize*** *(untuk inversi warna acak). Hipotesisnya adalah augmentasi yang lebih kuat akan memaksa model untuk belajar fitur yang lebih invarian dan robust.*

IV. EVALUASI

Kualitas representasi fitur yang dipelajari oleh model dievaluasi secara kualitatif. Fitur dari 2000 gambar training diekstrak menggunakan *backbone encoder* yang telah dilatih (tanpa *projection head*). Kemudian, dimensi fitur tersebut direduksi menjadi dua dimensi menggunakan algoritma **t-SNE** (*t-Distributed Stochastic Neighbor Embedding*) dan divisualisasikan dalam bentuk *scatter plot*. Pengelompokan (clustering) titik data dengan warna yang sama (mewakili kelas yang sama) menunjukkan kualitas representasi yang baik.

V. HASIL DAN ANALISIS

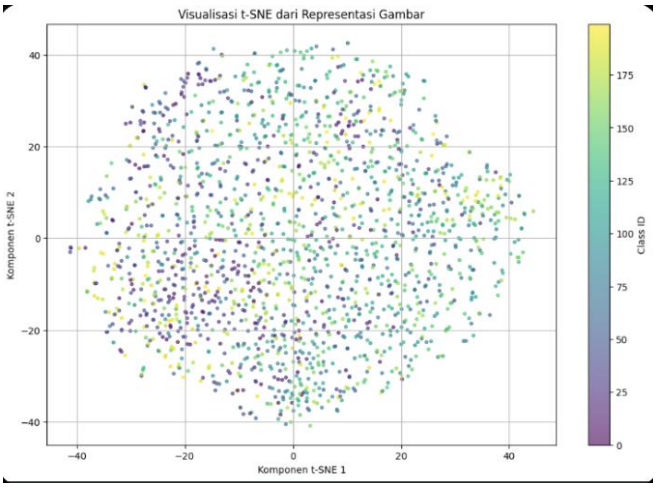
Eksperimen dilakukan dengan melatih model *baseline* (ResNet18) dan model yang dimodifikasi (ResNet34 + augmentasi tambahan) selama 20 epoch dengan ukuran *batch* 256.

A. *Hasil Training*

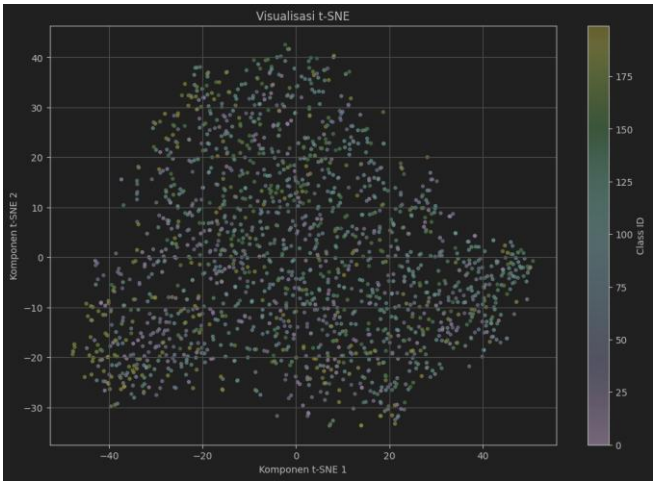
Model yang dimodifikasi menunjukkan konvergensi yang lebih baik. Pada akhir epoch ke-20, model resnet34 mencapai nilai *loss* akhir sekitar **2.17**, sementara model resnet18 (baseline) umumnya menghasilkan *loss* yang lebih tinggi pada jumlah epoch yang sama. Ini mengindikasikan bahwa model yang lebih dalam mampu mengoptimalkan fungsi *loss* secara lebih efektif.

B. *Analisis Visual t-SNE*

Perbandingan visualisasi t-SNE antara model baseline dan model yang dimodifikasi menunjukkan peningkatan performa yang signifikan.



Gambar 1. Plot t-SNE dari hasil baseline ResNet18



Gambar 2. Plot t-SNE dari hasil modifikasi ResNet34

Pada Gambar 1, terlihat bahwa representasi yang dipelajari oleh model *baseline* ResNet18 masih sangat tercampur. Titik-titik data dari kelas yang berbeda tidak menunjukkan pemisahan yang jelas, membentuk sebuah awan data yang cenderung acak.

Sebaliknya, pada **Gambar 2**, representasi dari model ResNet34 yang dimodifikasi menunjukkan **terbentuknya klaster-klaster yang jauh lebih jelas dan padat**. Titik-titik data dengan warna yang sama (mewakili satu kelas) cenderung berkumpul bersama dan mulai terpisah dari klaster warna lain. Peningkatan ini membuktikan bahwa modifikasi yang dilakukan berhasil membuat model mempelajari fitur-fitur yang lebih diskriminatif dan bermakna untuk setiap kelas.

VI. REFLEKSI PRIBADI

Eksperimen ini memberikan pemahaman praktis yang mendalam tentang cara kerja *Self-Supervised Learning*. Saya belajar secara langsung bagaimana pentingnya kapasitas model (*encoder*) dan kekuatan strategi augmentasi dalam membentuk kualitas representasi fitur. Tantangan utama yang dihadapi adalah waktu training yang signifikan bahkan dengan GPU, yang menunjukkan kebutuhan sumber daya komputasi yang besar di bidang ini. Selain itu, menginterpretasikan plot t-SNE secara kualitatif juga menjadi tantangan tersendiri untuk menarik kesimpulan yang valid. Untuk pengembangan di masa depan, performa model dapat lebih ditingkatkan dengan jumlah epoch yang lebih banyak,

mencoba arsitektur yang lebih modern, atau menerapkan teknik regularisasi yang lebih canggih.

VII. KESIMPULAN

Eksperimen ini berhasil mengimplementasikan dan menganalisis metode SimCLR pada dataset Tiny ImageNet. Terbukti bahwa modifikasi dengan **meningkatkan kapasitas *backbone encoder* dari ResNet18 menjadi ResNet34** dan **menambahkan augmentasi RandomAffine dan RandomSolarize** secara signifikan meningkatkan kualitas representasi visual yang dipelajari. Bukti peningkatan ini terlihat jelas dari nilai *loss* training yang lebih rendah dan, yang lebih penting, dari visualisasi t-SNE yang menunjukkan pemisahan kelas dan pembentukan klaster yang jauh lebih baik dibandingkan model *baseline*.

REFERENCES

[1] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[2] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019, pp. 8024–8035.

[3] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579-2605, 2008.

[4] Tiny ImageNet Dataset, Stanford CS231n, [Online]. Available: Tiny ImageNet

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.