

Prediksi Kelas Jamak dengan Deep Learning Berbasis Graphics Processing Units

Rahmadya Trias Handayanto ^{1,*}, Herlawati ²

¹ Fakultas Teknik, Universitas Islam 45; Jl. Cut Meutia No. 83, Bekasi Timur, Bekasi, Jawa Barat 17113. Telp: 021-8801027, 8802015, Fax: 021-8801192; e-mail: rahmadya.trias@gmail.com

² Fakultas Teknik, Universitas Bhayangkara Jakarta Raya; Jl. Raya Perjuangan, Marga Mulya, Bekasi Utara, Jawa Barat 17121. Telp: 021-88955882, 889955883, e-mail: herlawati@ubharajaya.ac.id

* Korespondensi: e-mail: rahmadya.trias@gmail.com

Abstract

For the first time, machine learning did the classical classification process using two classes (bi-class) such as class -1 and class +1, 0 and 1, or the form of categories such as true and false. Famous methods used are Artificial Neural Networks (ANN) and Support Vector Machine (SVM). The current development was a problem with more than two classes, known as multi-class classes. For SVM sometimes the plural classes are overcome by doing a gradual process like a decision tree (DT) method. Meanwhile, ANN has experienced rapid development and is currently being developed with a large number of layers with the new activation functions, i.e. the rectified linear units (ReLU), and the probabilistic-based activation, i.e. softmax, including its optimizer methods (adam, sgd, and others). Then the term changed to Deep Learning (DL). This study aimed to compare two well-known methods (DL and SVM) in classifying multiple classes. The number of DL layers was six with the neuron composition are 128, 64, 32, 8, 4, and 3, while SVM uses a radial kernel base function with gamma and c respectively 0.7 and 5. Besides, this study intends to compare the use of the Graphics Processing Unit (GPU) available on Google Interactive Notebook (Google Colab), an online Python language programming application. The results showed that DL accuracy outperformed SVM but required large computational resources, with the accuracy for DL and SVM are 99% and 98%, respectively. However, the use of the GPU can overcome these problems and is proven to increase the speed of the process as much as 47 times.

Keyword: Artificial Neural Networks, Graphics Processing Unit, Google Interactive Notebook, Rectified Linear units, Support Vector Machine.

Abstrak

Di awal perkembangannya mesin pembelajaran melakukan proses klasifikasi menggunakan dua kelas (bi-class) misalnya kelas -1 dan kelas +1, 0 dan 1, atau bentuk kategori seperti benar dan salah. Metode terkenal yang digunakan adalah Jaringan Syaraf Tiruan (JST) dan Support Vector Machine (SVM). Perkembangan selanjutnya adalah problem dengan kelas yang lebih dari dua kelas, dikenal dengan istilah kelas jamak (multi-class). Untuk SVM terkadang kelas jamak diatasi dengan melakukan proses berjenjang mirip pohon keputusan (decision tree). Sementara itu JST telah mengalami perkembangan yang pesat dan saat ini sudah dikembangkan dengan jumlah layer yang banyak disertai dengan fungsi-fungsi aktivasi terkini seperti rectified linear unit (ReLU), dan softmax yang berbasis probabilistik, termasuk juga

Available Online at <http://ejurnal.ubharajaya.ac.id/index.php/JKI>

metode-metode optimizernya (adam, sgd, dan lain-lain). Kemudian istilahnya berubah menjadi Deep Learning (DL). Penelitian ini mencoba membandingkan dua metode terkenal (DL dan SVM) dalam melakukan klasifikasi kelas jamak. Jumlah layer DL sebanyak enam dengan masing-masing neuron sebesar 128, 64, 32, 8, 4, dan 3, sementara SVM menggunakan kernel radial basis function dengan gamma dan c berturut-turut 0.7 dan 5. Selain itu penelitian ini bermaksud membandingkan penggunaan Graphics Processing Unit (GPU) yang tersedia di Google Interactive Notebook (Google Colab), sebuah aplikasi online pemrograman bahasa Python. Hasil penelitian menunjukkan akurasi DL unggul tipis dibanding SVM namun memerlukan sumber daya komputasi yang besar masing-masing dengan akurasi 99% dan 98%. Namun penggunaan GPU mampu mengatasi permasalahan tersebut dan terbukti meningkatkan kecepatan proses sebanyak 47 kali.

Kata kunci: Jaringan Syaraf Tiruan, Graphics Processing Unit, Google Interactive Notebook, Rectified Linear units, Support Vector Machine.

1. Pendahuluan

Saat ini perkembangan mesin pembelajaran (*machine learning*) sudah sangat pesat. Perkembangannya pun bukan hanya dari sisi perangkat lunak, perangkat kerasnya pun mengalami lonjakan yang drastis. Dengan perangkat keras yang baik, permasalahan yang dialami dalam penerapan *Deep Learning* (DL) dapat diatasi yaitu dengan meningkatkan jumlah ko-prosesor yang membantu kerja *Central Processing Unit* (CPU) berupa serangkaian *Graphics Processing Unit* (GPU) yang saat ini salah satu vendornya, NVIDIA, kerap meluncurkan versi-versi terbarunya.

Mesin pembelajaran merupakan bidang yang paling bersinggungan dengan perkembangan perangkat keras-perangkat keras tersebut, dipelopori oleh DL. Bidang yang merupakan kepanjangan dari statistik tersebut meluncurkan metode-metode terbarunya menyesuaikan dengan kemampuan GPU dalam menangani kebutuhan komputasi yang berat. Bahkan saat ini Google sebagai pelopor mesin pencarian telah memperkenalkan perangkat keras selain CPU dan GPU yang dikenal dengan nama *Tensor Processing Unit* (TPU) yang mengkombinasikan antara perangkat keras GPU yang paralel dengan struktur data TensorFlow yang juga paralel. Untuk mengatasi ketidakcocokan antara satu jenis perangkat keras dengan perangkat keras lainnya beberapa vendor menyediakan fasilitas penghubung, beberapa diantaranya adalah *Computer Unified Device Architecture* (CUDA) dan *Open Computing Language* (OpenCL) dengan prinsip kerja pemrosesan paralel (Koprawi, 2020). CUDA dan OpenCL banyak digunakan untuk bidang-bidang yang membutuhkan kerja berat prosesor seperti pemrosesan citra, enkripsi data, dan lain-lain. Prinsip yang membedakan antara CPU dan GPU adalah bagaimana prosesor tersebut menjalankan tugas komputasinya. CPU terdiri dari beberapa *core* yang dioptimalkan untuk beroperasi untuk tugas-tugas serial, sementara GPU yang juga terdiri dari jumlah *core* yang lebih banyak tetapi tetap dijaga kerjasamanya untuk menangani tugas-tugas paralel (Barney, 2020).

Jika di sisi perangkat keras terjadi kerjasama yang baik antar bidang ilmu, di sisi perangkat lunak dan metode, terjadi perpisahan antara cabang mesin pembelajaran dengan statistik dari sudut pandang objek penelitian. Jika bidang statistik fokus ke pemahaman terhadap model yang dibentuk, DL fokus ke hasil prediksi apakah akurat atau tidak.

Sementara model itu sendiri masih berupa kotak hitam (*black box*) yang menjadi pertanyaan peneliti-peneliti dari bidang studi statistika (Rasmussen & Williams, 2006). Sementara pemerhati mesin pembelajaran fokus ke kinerja model yang dirancang tanpa terlalu fokus ke penjelasan model yang masuk kategori *Artificial Intelligence* (AI) yang meniru penginderaan. Sebagai contoh, akan sulit mendefinisikan wajah seseorang baik secara linguistik maupun matematis, namun yang terpenting bagi bidang mesin pembelajaran adalah model yang mampu mengenal wajah seseorang (Kim, 2017).

Jaringan Syaraf Tiruan (JST) merupakan cikal bakal DL dengan prinsip meniru kerja otak makhluk hidup dengan komponen utamanya sel neuron dan saluran penghubung (sinapsis). Di awal perkembangannya, JST sempat sulit berkembang bahkan tidak sanggup menangani problem sederhana seperti *exclusive or* (XOR), sebelum diperkenalkannya *multilayer perceptron* dengan pembelajaran propagasi balik (*backpropagation*) di tahun 70-an (Hagan, M. T., Demuth, H. B., & Beale, 1997).

Masalah perangkat keras yang belum berkembang saat itu menjadi kendala ketika jumlah layer yang banyak, bahkan beberapa literatur di era itu menyebutkan bahwa layer optimal untuk JST adalah tiga layer (Fausett, 1994) yang tentu saja sangat terbatas jika digunakan untuk merancang DL dengan layer yang banyak. Banyak bidang-bidang yang sudah memanfaatkan JST dari sistem monitoring jaringan (Handayanto et al., 2011), geografi (Bhatti et al., 2015; Jiménez et al., 2018; Moayedi et al., 2019), hingga deteksi citra (Llobet et al., 1999; Putra et al., 2016) yang memiliki keunggulan dibanding metode lain (Whidhiasih, 2015; Yulianto et al., 2018).

Metode *Support Vector Machine* (SVM) yang dikembangkan Vapnik saat ini banyak digunakan karena tingkat akurasi yang tinggi (Cortes & Vapnik, 1995). Secara statistik dan matematis, SVM mirip dengan JST, dengan karakter awalnya yang digunakan untuk klasifikasi biner (*binary classification*). Selain bisa digunakan untuk klasifikasi jamak, metode ini bisa juga digunakan untuk regresi dengan istilah *Support Vector Regression* (SVR) dan sudah banyak diterapkan di beberapa kasus (Ayodeji & Liu, 2018; Drucker et al., 1997; Maharesi et al., 2013).

Penelitian ini membandingkan DL dengan SVM untuk kelas jamak. Selain membandingkan teknik-teknik implementasinya, penerapan DL dengan memanfaatkan GPU pada Google Interactive Notebook (Google Colab).

2. Metode Penelitian

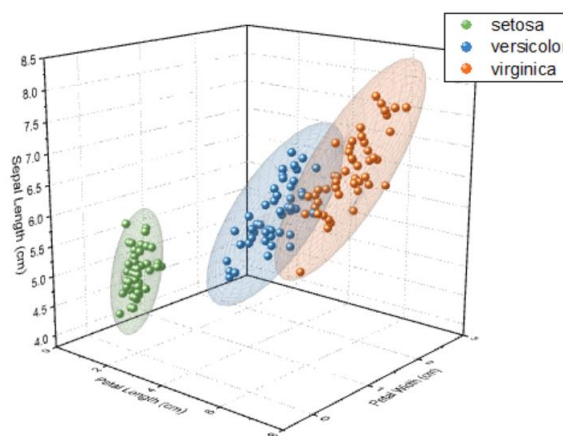
2.1 Data

Data untuk klasifikasi dalam penelitian ini menggunakan data iris yang dapat diakses di situs https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html. Data ini berisi klasifikasi kelopak bunga berdasarkan panjang sepal, lebar sepal, panjang petal dan lebar petal. Ada tiga kelas yang menjadi kategori kelopak bunga yaitu iris setosa, iris virginica dan iris versicolor (Gambar 1).

Data IRIS merupakan data sederhana dengan jumlah kelas minimal untuk kategori kelas jamak (hanya tiga kelas). Untuk data yang lain dapat diakses di situs scikit-learning di atas. Termasuk data yang berupa gambar, misalnya MNIST, yang terdiri dari 10.000 citra dengan 10 kelas/kategori. Situs tersebut selain menyediakan data juga menyediakan metode-metode mesin pembelajaran (termasuk DL di dalamnya) yang akan digunakan dalam penelitian ini.

Ada dua cara pengaksesan data iris, antara lain dengan mengunduh terlebih dahulu dan langsung lewat kode bahasa Python lewat fasilitas *import*. Dalam penelitian ini data diunduh terlebih dahulu mengingat untuk DL perlu kategori biner.

Fungsi DL dan SVM diambil dari pustaka (*library*) Skicit-Learning. DL menggunakan fungsi Keras dan TensorFlow dengan beberapa pustaka-pustaka pembantu dalam pengelolaan matriks dan data antara lain NumPy dan Pandas.



Sumber: (Originlab, n.d.)

Gambar 1. Visualisasi 3D Data Iris

2.2 Metode

Penelitian ini menggunakan dua metode terkenal yaitu DL dan SVM dalam melakukan proses klasifikasi data. Keduanya menggunakan proses klasifikasi yang sama. Langkah pertama adalah pemrosesan awal (*preprocessing*). Karena data sudah terproses dengan baik, maka tidak perlu dilakukan pemrosesan yang berarti. Khusus DL karena berupa kelas jamak, perlu perubahan dari kategori berupa "string" (setosa, versicolor dan virginica) perlu dikonversi menjadi 0, 1, dan 2. Langkah kedua adalah menyiapkan data pelatihan dan data

testing/validasi. Data pelatihan harus dibedakan dengan data validasi/testing untuk memperoleh perhitungan akurasi yang baik. Beberapa peneliti menganjurkan menggunakan validasi silang (*cross validation*) (Hou et al., 2015). Langkah ketiga adalah menyiapkan parameter-parameter pelatihan seperti jenis kernel untuk SVM, maupun jumlah neuron, fungsi aktivasi dan jumlah layer pada DL. Selain parameter, *hyper-parameter* juga perlu disiapkan seperti Epoch, stop condition, *optimizer*, dan lain-lain. Langkah keempat adalah proses pelatihan sekaligus melakukan pengecekan akurasi terhadap data validasi/testing. Di sini proses tidak serta merta selesai, terkadang perlu dilakukan perubahan parameter maupun *hyper-parameter* guna memperoleh akurasi model yang baik. Terkadang data validasi dibedakan dengan data testing. Di sini data testing merupakan data riil yang diambil dari sumber terbaru berbeda dengan data validasi yang memang diambil secara bersamaan dengan data pelatihan. Terakhir adalah menyimpan model yang sudah dilakukan proses pelatihan untuk siap digunakan untuk mengklasifikasi data baru.

a) **Deep Learning (DL)**

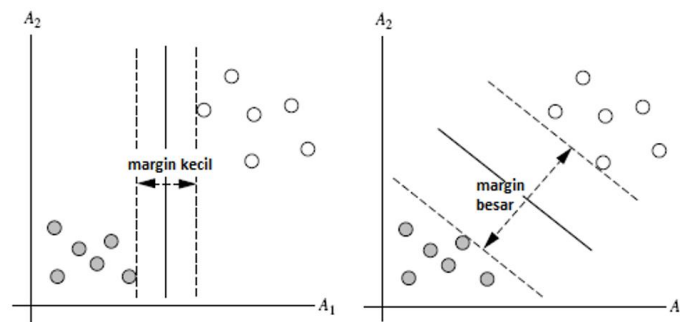
DL merupakan bagian dari mesin pembelajaran yang menerapkan model berbasis JST dengan multilayer. Berbeda dengan mesin pembelajaran yang menerapkan teknik komputasi tertentu, DL tidak memperhatikan konsep dasar, asalkan ada data dan struktur JST, model dapat dilatih segera (Kim, 2017).

Beberapa pengoptimasi model dapat diterapkan dalam DL, antara lain: *Stochastic Gradient Descent* (SGD), RMSProp, *Adaptive Subgradient Methods for Online Learning and Stochastic Optimization* (Adagrad), *An Adaptive Learning Rate Method* (Adadelata), Adam, Adamax, dan Nadam (Kingma & Ba, 2015). Pengoptimasi di atas sangat penting dalam perancangan DL karena berfungsi selain melakukan proses *fitting* antara data latih dan target, juga menghindari efek *overfitting* yang merugikan. Proses *overfitting* terjadi karena kegagalan model dalam membuat generalisasi pada proses pelatihan. Biasanya kasus yang terjadi adalah turunnya performa model ketika diuji dengan data di luar data pelatihan.

Parameter yang digunakan dalam penelitian ini adalah sebagai berikut. Layer sebanyak enam termasuk input dan output dengan masing-masing neuron sebanyak berturut-turut 128, 64, 32, 8, 4, dan 3. Fungsi aktivasi seluruhnya *rectified linear unit* kecuali output bertipe *Softmax*. Di sini output berjumlah tiga untuk mengakomodasi tiga kelas, dan fungsi *softmax* memberikan nilai probabilitas untuk tiap-tiap kelas yang dihasilkan proses klasifikasi. Jika probabilitas tertinggi pada kelas ketiga maka hasil klasifikasi memberikan jawaban kelas tiga "virginica", misalnya. Proses pelatihan menggunakan *loss function* 'sparse categorical crossentropy', *optimizer stochastic gradient descent* dan disertai *metric* untuk perhitungan akurasi pelatihan. Iterasi proses pelatihan sebesar 1000 kali guna memperoleh hasil yang akurat.

b) Support Vector Machine (SVM)

Untuk jumlah kelas yang tidak banyak SVM sangat efektif dalam proses klasifikasi. Metode ini menggunakan kernels dalam menentukan *support vector*, yaitu elemen data terluar sebagai pemisah dengan kelas lainnya. Metode ini bermaksud mencari bidang pemisah terlebar antara satu kelas dengan kelas lainnya seperti tampak pada gambar 2.



Sumber: (Widodo et al., 2013)

Gambar 2. Pencarian margin besar dengan SVM

Terkadang garis pemisah tidak mampu memisahkan secara langsung data yang tersebar sehingga perlu memanipulasi kernels menjadi bidang yang disebut *hyperplane*. Sehingga secara umum persamaannya adalah:

$$W \cdot X + b = 0 \tag{1}$$

$$w_0 + w_1x_1 + w_2x_2 = 0 \tag{2}$$

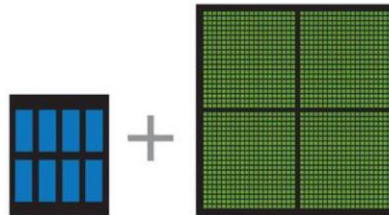
Parameter-parameter SVM yang digunakan dalam penelitian ini adalah sebagai berikut. Kernel yang digunakan *Radial Basis Function* (RBF) dengan Gamma sebesar 0.7 dan variabel C sebesar 5.

c) Google Interactive Notebook (Google Colab)

Perkembangan bahasa pemrograman Python yang sangat pesat menarik minat Google untuk membuat *Integrated Environment Development* (IDE) secara online yang dikenal dengan istilah Google Interactive Notebook atau biasa disebut Google Colab melalui situs resminya www.colab.research.google.com. Jenis *environment* yang digunakan adalah Jupyter Notebook dengan ekstensi file *.ipynb. Sebagai informasi Python memiliki beragam *environment* untuk pemrogramannya dari IDLE yang telah lama digunakan hingga Spyder dengan *environment* yang komplit. Namun dengan karakter yang berbasis web, Jupyter Notebook lebih disukai Google.

Dari sisi perangkat lunak Google Colab telah menyiapkan hampir sebagian besar pustaka (*library*) yang dibutuhkan. Dalam penelitian ini pustaka yang dibutuhkan adalah Keras, TensorFlow, NumPy, Pandas, dan pendukung lainnya, misalnya untuk pembuatan grafik lewat Matplotlib. Bahkan seluruh versi disediakan, misalnya TensorFlow versi 1.x maupun 2.x tersedia, begitu juga versi Python yang mulai dari versi 2.x hingga 3.x.

Dari sisi perangkat keras Google Colab menyediakan fasilitas berupa media penyimpanan yang terintegrasi dengan Google Drive, prosesor yang berupa CPU, GPU, dan TPU, serta RAM. Dengan jaminan kemampuan servernya hampir sebagian besar pemrosesan tidak memiliki kendala dengan Google Colab selama koneksi jaringan internet lancar. Dalam penelitian ini, fokus diberikan pada GPU yang saat ini banyak digunakan sebagai pendukung DL. GPU yang melakukan pemrosesan paralel cocok dengan pengolahan data dengan DL dengan struktur mult-core-nya (Gambar 3). GPU memiliki jumlah *core* yang jauh lebih banyak guna melakukan tugas-tugas pemrosesan paralel (Pan et al., 2008).

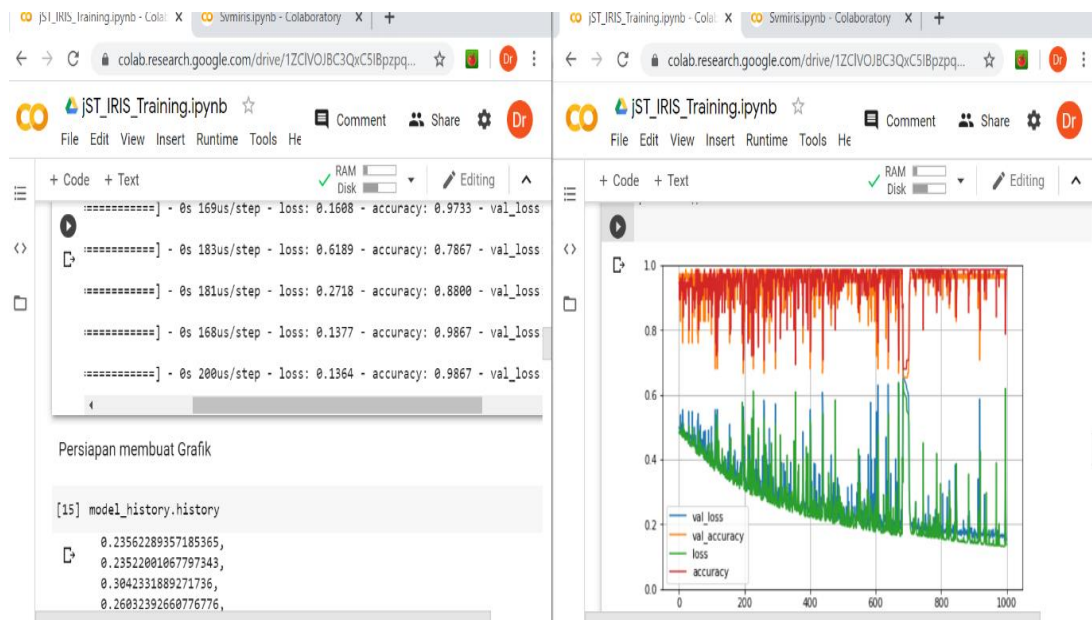


Sumber: (NVIDIA, 2020)

Gambar 3. CPU (kiri) dengan beberapa *core* sementara GPU (kanan) memiliki ratusan *core*

3. Hasil dan Pembahasan

Dalam penelitian ini model SVM dan DL dilatih menggunakan data yang sama yaitu data iris. Google Colab digunakan untuk menjalankan kode dua model tersebut. DL menunjukkan hasil akurasi yang cukup tinggi (99%) seperti ditunjukkan gambar 4.



Sumber: Hasil Penelitian (2020)

Gambar 4. Kinerja Pelatihan DL

Untuk SVM menggunakan pengecekan akurasi lewat matriks *confusion* seperti ditunjukkan dalam gambar 5.

```
[3] /usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:760:
y = column_or_1d(y, warn=True)
SVC(C=5, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=3, gamma=0.7, kernel='rbf',
max_iter=1, probability=False, random_state=None, shrinking=True,
tol=0.001, verbose=False)

y_pred=models.predict(x_val)
print(classification_report(y_val,y_pred))
```

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	20
Iris-versicolor	0.90	1.00	0.95	9
Iris-virginica	1.00	0.94	0.97	16
accuracy			0.98	45
macro avg	0.97	0.98	0.97	45
weighted avg	0.98	0.98	0.98	45

Sumber: Hasil Penelitian (2020)

Gambar 5. Kinerja Pelatihan SVM

SVM menunjukkan hasil konsisten dengan akurasi 98% sedikit dibawah DL namun tetap menunjukkan kinerja yang baik. Penggunaan GPU pada Google Colab terbukti meningkatkan waktu proses yang lebih baik seperti ditunjukkan pada gambar 6 berikut ini.

```
[1] print('CPU (s):')
cpu_time = timeit.timeit('cpu()', number=10, setup="from __main__
print(cpu_time)
print('GPU (s):')
gpu_time = timeit.timeit('gpu()', number=10, setup="from __main__
print(gpu_time)
print('GPU speedup over CPU: {}'.format(int(cpu_time/gpu_time))
```

```
Time (s) to convolve 32x7x7x3 filter over random 100x100x100x3 i
CPU (s):
2.4718096449999933
GPU (s):
0.05207660299998906
GPU speedup over CPU: 47x
```

Sumber: Hasil Penelitian (2020)

Gambar 6. Perbandingan Kinerja DL dengan dan tanpa GPU

Kinerja menggunakan GPU (0.052) terbukti lebih cepat dibanding tanpa GPU (2.472) sebesar 47 kali lebih cepat.

4. Kesimpulan

Hasil pengolahan data, pelatihan, dan pengujian menunjukkan kedua metode (DL dan SVM) mampu dengan baik melakukan proses klasifikasi. DL unggul tipis (99% melawan 98%) namun memerlukan proses yang lebih banyak (dengan seribu iterasi). Namun penggunaan GPU dapat membantu permasalahan DL yang membutuhkan sumber daya komputasi yang lebih besar. Hasil uji menunjukkan penggunaan GPU jauh lebih cepat jika tanpa GPU (sebesar 47 kali lebih cepat). Penelitian berikutnya perlu dilakukan pengujian dengan data yang lebih kompleks seperti data berupa citra maupun suara guna memperoleh informasi lebih lanjut akan ketangguhan kedua metode tersebut.

Daftar Pustaka

- Ayodeji, A., & Liu, Y. kuo. (2018). SVR optimization with soft computing algorithms for incipient SGTR diagnosis. *Annals of Nuclear Energy*, 121, 89–100. <https://doi.org/10.1016/j.anucene.2018.07.011>
- Barney, B. (2020). *Introduction to Parallel Computing*.
- Bhatti, S. S., Tripathi, N. K., Nitivattananon, V., Rana, I. A., & Mozumder, C. (2015). A multi-scale modeling approach for simulating urbanization in a metropolitan region. *Habitat International*, 50, 354–365. <https://doi.org/10.1016/j.habitatint.2015.09.005>
- Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1023/A:1022627411411>
- Drucker, H., Surges, C. J. C., Kaufman, L., Smola, A., & Vapnik, V. (1997). Support vector regression machines. *Advances in Neural Information Processing Systems*, 1, 155–161.
- Fausett, L. (1994). *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice-Hall, Inc.
- Hagan, M. T., Demuth, H. B., & Beale, M. (1997). *Neural Network Design*. PWS Publishing Co.
- Handayanto, R. T., Haryono, & Prianggono, J. (2011). Real-time neural network-based network analyzer for hotspot area. *ICACISIS 2011 - 2011 International Conference on Advanced Computer Science and Information Systems, Proceedings*.
- Hou, J. C., Hu, Y. H., Hou, L. X., Guo, K. Q., & Satake, T. (2015). Classification of ripening stages of bananas based on support vector machine. *International Journal of Agricultural and Biological Engineering*, 8(6), 99–103. <https://doi.org/10.3965/j.ijabe.20150806.1275>
- Jiménez, A. A., Vilchez, F. F., González, O. N., & Flores, S. M. L. M. (2018). Analysis of the Land Use and Cover Changes in the Metropolitan Area of Tepic-Xalisco (1973–2015) through Landsat Images. *Sustainability*, 10(6). <https://doi.org/10.3390/su10061860>
- Kim, P. (2017). *MATLAB Deep Learning*. Apress. <https://doi.org/10.1007/978-1-4842-2845-6>

- Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.
- Koprawi, M. (2020). *Parallel Computation in Uncompressed Digital Images Using Computer Unified Device Architecture and Open Computing Language*. 8(1), 31–38.
- Llobet, E., Hines, E. L., Gardner, J. W., & Franco, S. (1999). Non-destructive banana ripeness determination using a neural network-based electronic nose. *Measurement Science and Technology*, 10(6), 538–548. <https://doi.org/10.1088/0957-0233/10/6/320>
- Maharesi, R., Teknologi, F., Jurusan, I., Informatika, T., & Gunadarma, U. (2013). Penggunaan Support Vector Regression (Svr) Pada Prediksi Return Saham Syariah BEI. *Proceeding PESAT*, 5, 8–9.
- Moayedi, H., Mehrabi, M., Mosallanezhad, M., Rashid, A. S. A., & Pradhan, B. (2019). Modification of landslide susceptibility mapping using optimized PSO-ANN technique. *Engineering with Computers*, 35(3), 967–984. <https://doi.org/10.1007/s00366-018-0644-0>
- NVIDIA. (2020). *NVIDIA on GPU Computing and the Difference Between GPUs and CPUs*.
- Originlab. (n.d.). *Transparent Parametric Surfaces combined with Scatter Plot*.
- Pan, L., Gu, L., & Xu, J. (2008). Implementation of medical image segmentation in CUDA. *5th Int. Conference on Information Technology and Applications in Biomedicine, ITAB 2008 in Conjunction with 2nd Int. Symposium and Summer School on Biomedical and Health Engineering, IS3BHE 2008*, 82–85. <https://doi.org/10.1109/ITAB.2008.4570542>
- Putra, I. W. S. E., Wijaya, A. Y., & Soelaiman, R. (2016). *Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101 Image Classification Using Convolution Neural Network (Cnn) on Caltech 101*.
- Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.
- Whidhiasih, R. N. (2015). Identifikasi tingkat manis buah belimbing berdasarkan citra red green blue menggunakan fuzzy neural network. *PIKSEL: Penelitian Ilmu Komputer Sistem Embedded and Logic*, 3(2), 109–120.
- Widodo, P. P., Handayanto, R. T., & Herlawati. (2013). *Penerapan Data Mining dengan Matlab*. Informatika.
- Yulianto, D., Whidhiasih, R. N., & Maimunah, M. (2018). Klasifikasi Tahap Kematangan Pisang Ambon Berdasarkan Warna Menggunakan Naive Bayes. *PIKSEL: Penelitian Ilmu Komputer Sistem Embedded and Logic*, 5(2), 60–67. <https://doi.org/10.33558/piksel.v5i2.268>