JITEKH, Vol 7, No 1, Tahun 2019, 34-38
ISSN 2338-5677 Cetak
ISSN 2549-6646 Online

KLASIFIKASI GAMBAR DATASET FASHION-MNIST MENGGUNAKAN *DEEP CONVOLUTIONAL NEURAL NETWORK*

Stephanus Priyowidodo
Program Studi D3 Manajemen Informatika, Universitas Harapan Medan
Email : stephanus.priyowidodo@gmail.com

Abstrak

Penelitian ini menggunakan dataset Fashion-MNIST dari Zalando Research, yang terdiri atas 60000 gambar untuk pelatihan dan 10000 gambar untuk pengujian, masing-masing gambar berukuran 28x28 piksel. Metode *deep learning* yang digunakan adalah *Deep Convolutional Neural Network* (DCNN), dengan fungsi aktivasi *relu* di bagian dalam *layer* dan *softmax* pada bagian akhir *layer*. Akurasi pengujian tanpa *dropout* mendapatkan hasil 92.69% dengan *loss* 0.445 dan dengan *dropout* mendapatkan hasil 92.84%, *loss* 0.206.
Kata Kunci: *Deep Learning, Image Classification*

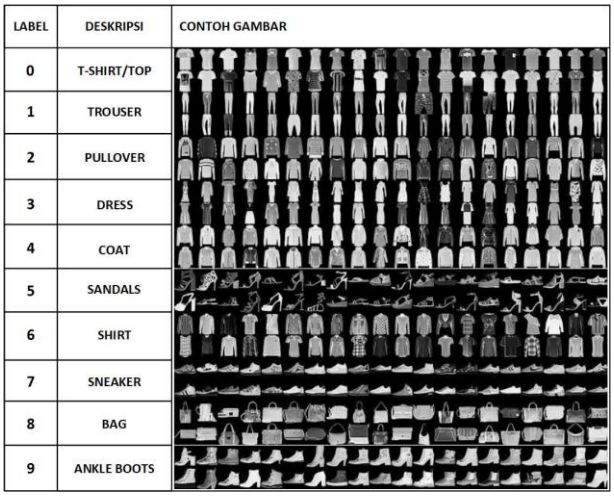
Abstract

This study uses the Fashion-MNIST dataset from Zalando Research, which consists of 60000 images for training and 10000 images for testing, each image size 28x28 pixels. The deep learning method used is the Deep Convolutional Neural Network (DCNN), with the activation function relu on the inside of the layer and softmax at the end of the layer. Test accuracy without using dropout gets 92.69% with loss of 0.445 and using dropout gets 92.84%, loss 0.206.
Keywords: *Deep Learning, Image Classification*

1. PENDAHULUAN

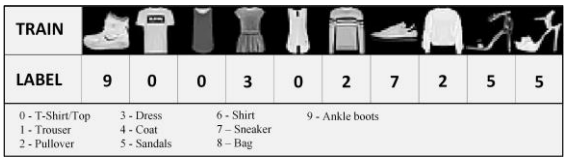
Fashion-MNIST (Xiao, 2017) merupakan dataset yang diambil dari kumpulan gambar produk *fashion* pada situs web Zalando. Sebelumnya telah ada dataset tulisan tangan angka MNIST dari LeCun dkk.[1998] yang merupakan dataset terbanyak digunakan sebagai dataset pengujian pada *deep learning*.

Dataset Fashion-MNIST memiliki jumlah dan ukuran yang sama dengan MNIST, namun dengan jenis klasifikasi berbeda. Klasifikasi Fashion-MNIST yakni *T-Shirt/Top*, *Trouser*, *Pullover*, *Dress*, *Coat*, *Sandals*, *Shirt*, *Sneaker*, *Bag* dan *Ankle boots*, seperti terlihat pada gambar 1.



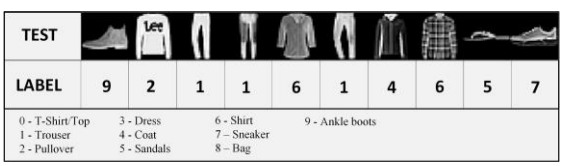
Gambar 1. Klasifikasi Fashion-MNIST

Dataset Fashion-MNIST terbagi atas dua bagian, pelatihan (*train*) dan pengujian (*test*). Pada file pelatihan terdapat 60000 gambar berukuran 28x28 piksel yang masing-masing gambar berhubungan dengan informasi label pada file label pendampingnya. Pada gambar 2 terlihat 10 gambar pertama pada file pelatihan dan 10 label pertama pada file label pelatihan.



Gambar 2. 10 gambar/label train Fashion-MNIST

Pada file pengujian terdapat 10000 gambar disertai 10000 label. 10 gambar pertama pengujian Fashion-MNIST beserta label-nya terlihat pada gambar 3.



Gambar 3. 10 gambar/label test Fashion-MNIST

ISSN 2338-5677 Cetak
ISSN 2549-6646 Online

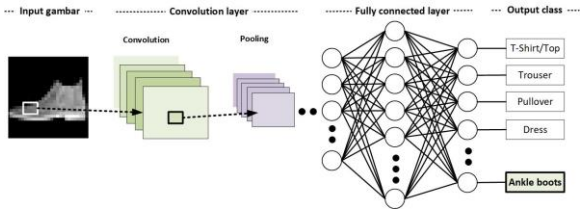
Struktur data dan teknik pembacaan dataset MNIST dibahas secara detail pada Susilawati [2017].

2. METODE PENELITIAN

Klasifikasi gambar yang digunakan pada penelitian ini adalah Deep Convolutional Neural Network (DCNN), dengan pengukuran *test loss* dan *test accuracy*.

2.1 Arsitektur CNN

CNN adalah *feedforward network* yang informasinya mengalir hanya ke satu arah, dari *input* ke *output*. Arsitektur CNN secara umum berisi *layer convolutional* dan *layer pooling* (*subsampling*) yang dikelompokkan ke dalam modul-modul. Modul-modul disusun *stack* di atas modul lainnya membentuk Deep CNN. Gambar 4 mengilustrasikan arsitektur DCNN yang digunakan pada pengenalan dataset Fashion-MNIST.



Gambar 4. Arsitektur CNN

2.2 Convolutional Layer

Convolutional layer berfungsi sebagai ekstraktor fitur, yang berfungsi mempelajari representasi fitur dari gambar *input*. *Neuron* pada layer ini disusun menjadi peta fitur. Setiap *neuron* dalam peta fitur memiliki bidang reseptif, yang terhubung ke lingkungan *neuron* pada *layer* sebelumnya melalui serangkaian bobot yang dapat dilatih, sering disebut *bank filter* (LeCun dkk., 2015). *Input* dikonvolusi dengan bobot yang dipelajari untuk menghitung peta fitur baru, dan hasil konvolusi dikirim melalui fungsi aktivasi *nonlinier*. Semua *neuron* dalam peta fitur memiliki bobot yang dibatasi agar berbeda. Peta fitur yang berbeda pada lapisan konvolusional yang sama memiliki bobot yang berbeda, sehingga beberapa fitur dapat diekstraksi di setiap lokasi (LeCundkk., 1998; LeCun dkk., 2015). Peta fitur keluaran ke-*k* dapat dihitung dengan :

$$Y_k = f(W_k * x) \qquad (2.1)$$

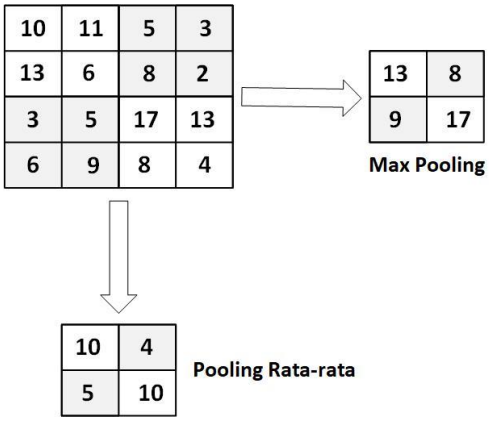
di mana gambar input dilambangkan dengan *x*; *filter* konvolusional yang terkait dengan peta fitur ke-*k* dilambangkan dengan W_k; tanda multiplikasi dalam konteks ini mengacu pada operator konvolusional 2D, yang digunakan untuk menghitung produk dalam dari *model filter* pada setiap lokasi gambar *input* dan $f(\cdot)$ mewakili fungsi aktivasi *nonlinier* (Yu dkk., 2014). Fungsi aktivasi *nonlinier* memungkinkan ekstraksi fitur *nonlinier*. Fungsi aktivasi *sigmoid* dan *hyperbolic tangent* dianggap tradisional dan digantikan oleh fungsi aktivasi *rectified linear units* (ReLU) (Nair & Hinton, 2010) (LeCun dkk., 2015).

2.3 Pooling Layer

Fungsi *pooling layer* adalah untuk mengurangi resolusi spasial dari peta fitur agar tercapai invarian spasial untuk distorsi masukan dan translasi (LeCun dkk., 1989a, 1989b; LeCun dkk., 1998, 2015; Ranzato dkk., 2007). Awalnya digunakan rata-rata dari kumpulan lapisan agregasi untuk menyebarkan rata-rata semua nilai input dari lokasi kecil gambar ke lapisan berikutnya (LeCun dkk., 1989a, 1989b; LeCun dkk., 1998). Namun saat ini digunakan *max pooling*, layer agregasi menyebarkan nilai maksimum pada bidang reseptif ke lapisan berikutnya (Ranzato dkk., 2007). Secara formal, *max pooling* memilih elemen terbesar pada setiap bidang reseptif sehingga :

$$Y_{ki\,j} = \max_{(p,q)\in\Re_{i\,j}} x_{kpq} \qquad (2.2)$$

di mana *output* dari operasi *pooling*, yang terkait dengan peta fitur ke-*k*, dilambangkan oleh Y_k. x_{kpq} menunjukkan elemen lokasi *(p, q)* yang diisi dari wilayah *pooling* $\Re ij$, yang membentuk bidang reseptif di sekitar posisi *(i, j)* (Yu dkk., 2014). Gambar 5 mengilustrasikan perbedaan antara *max pooling* dan rata-rata *pooling*. Terdapat input gambar 4x4, bila diterapkan filter 2x2 dengan dua langkah, maka *output max pooling* adalah nilai maksimum dari wilayah 2x2, sedangkan *output* rata-rata diperoleh dari rata-rata nilai di masing-masing wilayah sub-sample.



Gambar 5. *Max Pooling* dan *Pooling* Rata-rata

ISSN 2338-5677 Cetak
ISSN 2549-6646 Online

2.4 Fully Connected Layer

Beberapa *convolutional layer* dan *pooling layer* di-*stack* satu sama lain untuk mengekstraksi representasi fitur yang lebih abstrak, yang bergerak melintasi jaringan. Kemudian tersambung ke lapisan yang sepenuhnya terhubung (*fully connected layer*) yang berfungsi menginterpretasikan representasi fitur dan melakukan fungsi penalaran tingkat tinggi (Hinton dkk., 2012; Simonyan & Zisserman, 2014). Untuk permasalahan klasifikasi digunakan operator *softmax* pada bagian akhir DCNN (Krizhevsky dkk., 2012; Simonyan & Zisserman, 2014).

2.5 DCNN Model

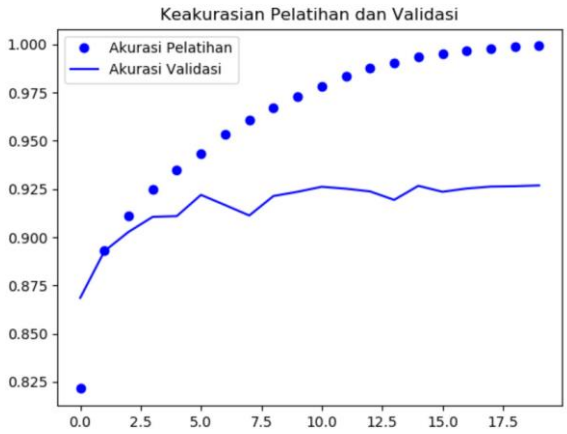
Model jaringan yang digunakan pada penelitian ini terlihat seperti pada gambar 4. Terdapat dua lapisan konvolusional, pertama lapisan konvolusi 32 dengan *kernel size* 3x3, fungsi aktivasi ReLU. Kedua, lapisan konvolusi 64 dengan *kernel size* 3x3 dan fungsi aktivasi ReLU. Selanjutnya di *max pooling* dengan ukuran pool 2x2. *Output max pooling* menjadi masukan dari lapisan *fully connected* dengan 128 *neuron* dan fungsi aktivasi ReLU. *Output*nya dihubungkan ke lapisan akhir sejumlah *class* pada Fashion-MNST dengan fungsi aktivasi *softmax*. Algoritma optimasi yang digunakan adalah *Adam optimizer*.

3. HASIL DAN PEMBAHASAN

3.1 DCNN Tanpa Dropout

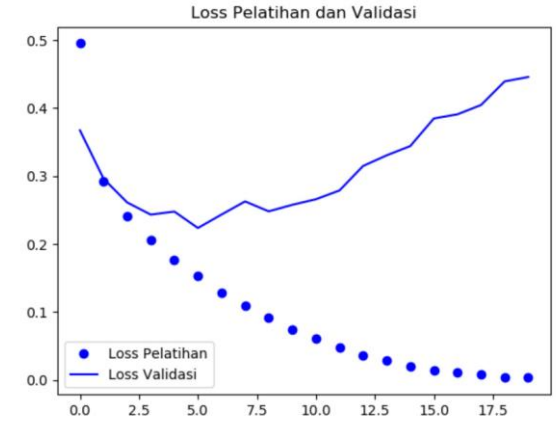
Hasil ujicoba pertama memperlihatkan kinerja DCNN sangat baik, terlihat dari hasil rata-rata akurasi pengujian 92,69%, dengan *loss* pengujian 0,4458.

Dari grafik gambar 6, akurasi pelatihan sejak awal menaik dimulai dari *epoch1* sebesar 0.8216, *epoch2* 0.8933 sampai *epoch20* 0.9992. Hasil berbeda terlihat pada akurasi saat validasi, pada *epoch1* sampai *epoch6* grafik menanjak naik meski tidak signifikan. Pada *epoch7* mulai menurun dari 0.9220 menjadi 0.9167, 0.9113 dan mendatar sampai *epoch* terakhir.



Gambar 6. Akurasi Pelatihan dan Validasi

Loss terlihat pada gambar 7, *loss* saat pelatihan sejak *epoch1* mengalami penurunan yang sangat signifikan, dimulai dari 0.4952, 0.2926, 0.2414, 0.2059 menurun terus hingga *epoch20* sebesar 0.0035. Namun berbeda dengan *loss* validasi, terlihat menurun di awal *epoch* sampai *epoch4* dengan masing-masing nilai 0.3674, 0.2962, 0.2613, 0.2434. *Epoch5 loss* mulai terlihat tidak stabil menjadi 0.2479 dan menurun pada epoch7 0.2237, dan terus menaik hingga akhir *epoch* sebesar 0.4458.



Gambar 7. *Loss* Pelatihan dan Validasi

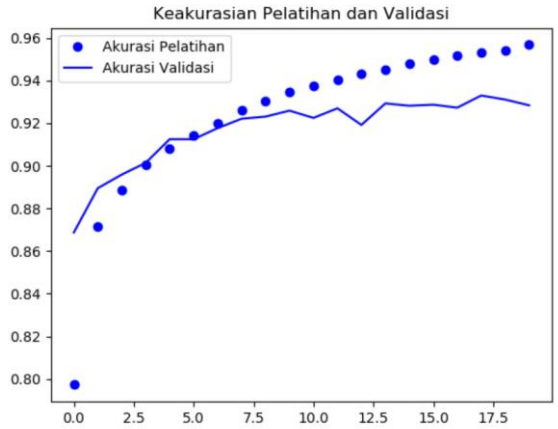
Dari grafik *loss* yang menaik saat validasi, kemungkinan telah terjadi *overfitting* pada jaringan. Pada ujicoba kedua akan ditambahkan fungsi *dropout* (Hinton, 2012; Srivastava, 2014) pada jaringan dengan tujuan mengurangi terjadinya *overfitting*.

3.2 DCNN dengan Dropout

Pada ujicoba kedua ditambahkan fungsi *dropout* pada lapisan antara *max-pooling* dan lapisan *fully connected* sebesar 0.25. Dan *dropout* 0.5 diantara lapisan *fully connected* 128 neuron dan lapisan akhir 10 neuron.

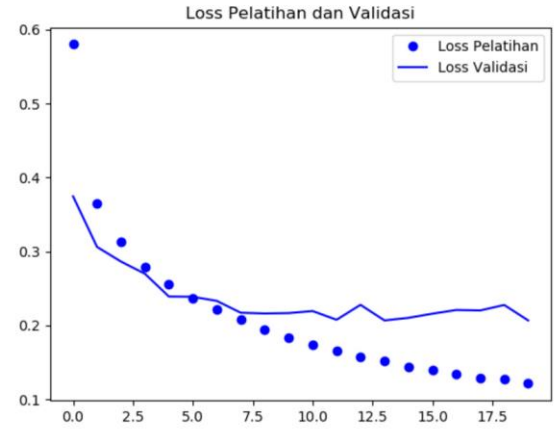
ISSN 2338-5677 Cetak
ISSN 2549-6646 Online

Hasil ujicoba kedua menunjukkan peningkatan kinerja DCNN, hasil rata-rata akurasi pengujian 92.84% dan loss pengujian 0.2066.



Gambar 8. Akurasi Pelatihan dan Validasi *Dropout*

Grafik gambar 8 memperlihatkan efek fungsi *dropout* pada DCNN. Untuk akurasi pelatihan kinerjanya hampir sama dengan ujicoba pertama, nilainya menaik dari awal sampai akhir *epoch*, dimulai dengan nilai 0.7973, 0.8717, 0.8888, 0.9004, 0.9083, 0.9145, 0.9202 sampai pada *epoch20* 0.9568. Akurasi validasi menjadi lebih baik, menaik sejak *epoch1* sampai *epoch7* dengan nilai 0.8687, 0.8895, 0.8959, 0.9014, 0.9125, 0.9125, 0.9177, meskipun masih terjadi penurunan pada *epoch8* dan cenderung stabil sampai *epoch20*.



Gambar 9. *Loss* Pelatihan dan Validasi *Droput*

Loss ujicoba kedua terlihat pada gambar 9, terlihat loss saat pelatihan cenderung turun sejak awal *epoch* hingga akhir *epoch*, dengan nilai 0.5801, 0.3643, 0.3131, 0.2787, 0.2551, 0.2368 dan menurun terus sampai *epoch20* 0.1214. Dengan *dropout*, *loss* validasi menjadi jauh lebih baik dibandingkan dengan ujicoba pertama. Nilai *loss* validasi epoch1 0.3745, menurun terus hngga epoch4, 0.3061, 0.2863, 0.2698, kemudian naik sedikit pada *epoch5* sebesar 0.2390 dan menurun sampai akhir *epoch*.

4. PENUTUP

4.1 Kesimpulan

Deep Convolutional Neural Network (DCNN) terbukti mampu mengklasifikasikan gambar pada dataset Fashion-MNIST dengan hasil yang sangat baik. Saat pengujian data, penggunaan *dropout* mampu mengurangi *overfitting*, menaikkan akurasi dan mengurangi *loss*. Hasil tanpa *dropout* 92.69% dengan *loss* 0.445 dan dengan *dropout* 92.84%, *loss* 0.206, melebihi akurasi maksimal yang diperoleh *Zalando Research*, pemilik *dataset* Fashion-MNIST (Xiao dkk., 2017) yakni 89.7%.

4.2 Saran

Memaksimalkan *dropout* dan memodifikasi *model* jaringan kemungkinan dapat meningkatkan kinerja DCNN.

5. DAFTAR PUSTAKA

[1.] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), Advances in neural information processing systems, 25 (pp. 1097–1105). Red Hook, NY: Curran.

[2.] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

[3.] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989a). Handwritten digit recognition with a back-propagation network. In D. S. Touretzky (Ed.), Advances in neural information processing systems, 2 (pp. 396–404). Cambridge, MA: MIT Press.

[4.] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989b). Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4), 541–551.

[5.] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444

[6.] Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (pp. 807–814). N.p.: International Machine Learning Society.

[7.] Ranzato, M. A., Huang, F. J., Boureau, Y., & LeCun, Y. (2007). Unsupervised learning of

ISSN 2338-5677 Cetak
ISSN 2549-6646 Online

invariant feature hierarchies with applications to object recognition. In Proceedings IEEE Conference on Computer Vision and Pattern Recognition (pp. 1–8). Los Alamitos, CA: IEEE Computer Society.

[8.] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv 1409.1556.

[9.] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R.(2014). Dropout: Asimple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929–1958.

[10.] Susilawati (2017). Algoritma Restricted Boltzman Machines (RBM) untuk Pengenalan Tulisan Tangan. Seminar Nasional Teknologi Informatika, 140-148.

[11.] Xiao, H., Rasul, K., Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747

[12.] Yu, D., Wang, H., Chen, P., & Wei, Z. (2014). Mixed pooling for convolutional neural networks. In Proceedings of the 9th International Conference on Rough Sets and Knowledge Technology (pp. 364–375). Berlin: Springer.