

Analisa *Learning rate* dan *Batch size* Pada Klasifikasi Covid Menggunakan *Deep learning* dengan *Optimizer Adam*

Naim Rochmawati¹, Hanik Badriyah Hidayati², Yuni Yamasari³,

Hapsari Peni Agustin Tjahyaningtjas⁴, Wiyli Yustanti⁵, Agus Prihanto⁶

^{1,3,4,5,6} Teknik Informatika, Universitas Negeri Surabaya

² Fakultas Kedokteran, Universitas Airlangga

Naimrochmawati@unesa.ac.id

Abstrak— *Deep learning* semakin berkembang pesat dan banyak dimanfaatkan dalam berbagai bidang kehidupan. Salah satunya bisa dimanfaatkan untuk klasifikasi *image* medis penderita covid. Keras adalah salah satu *framework deep learning* yang paling banyak digunakan. Dalam Keras, terdapat beberapa macam algoritma *optimizer*. Salah satunya adalah *optimizer Adam*. Untuk menggunakan *optimizer Adam* ini, perlu menentukan angka *learning rate*. Penentuan angka *learning rate* sangat penting karena salah dalam menentukan angka *learning rate* akan berdampak pada hasil *deep learning* yang dilakukan. *Batch size* juga salah satu *hyperparameter* penting dalam *deep learning*. Penelitian ini bertujuan untuk mengetahui dan membandingkan beberapa *learning rate* dan *batch size* agar diketahui efek dan dampaknya pada hasil *loss* dan akurasi training dan validasi pada proses *deep learning* yang dilakukan. Ada 6 *learning rate* dan 3 *batch size* yang akan dibandingkan. Hasil yang optimal diantara 6 *learning rate* dalam penelitian ini adalah 0.0001 dan 0.00001. Sedangkan *batch size* yang paling bagus hasilnya dari tiga angka yang dibandingkan adalah *batch size* 5.

Kata Kunci— *learning rate*, *batch size*, *optimizer*, Adam, klasifikasi, covid, *deep learning*, akurasi, *loss*.

I. PENDAHULUAN

Sejak pandemic dimulai dari bulan desember 2019 yang lalu, mau tidak mau, manusia dipaksa berdampingan hidup dengan covid-19, yaitu virus yang menyerang system pernafasan [1]. Setelah sekian lama wabah tersebar, berbagai bidang kehidupan terkena dampaknya. Data penderita covid juga semakin lengkap dan bisa diakses *public*. Data ini bila diolah akan menjadi satu informasi yang bermanfaat bagi manusia dan juga untuk perkembangan ilmu pengetahuan.

Salah satu ilmu yang bisa digunakan untuk mengolah data ini adalah *deep learning*. *Deep learning* adalah bagian dari pembelajaran mesin yang berkaitan dengan algoritma dimana cara kerja dari algoritma ini meniru struktur dan fungsi otak yang disebut jaringan saraf tiruan. Penelitian terkait *deep learning* telah banyak dilakukan dalam berbagai bidang. Seperti misalnya di bidang *social*, ekonomi[2] dan bahkan hingga masalah emosi[3], [4] *music*[5] juga *social media*[6]. Beberapa penelitian terkait bidang kesehatan adalah deteksi kanker payudara [7], deteksi covid [8], tumor otak [9], klasifikasi *image* medis[10] dan sebagainya.

Framework deep learning yang paling banyak digunakan adalah Keras. *Framework* ini termasuk 5 pemenang tim dalam Kaggle [11]. Dalam *framework* keras, terdapat beberapa model yang bisa digunakan. Salah satunya adalah Inception-V3. Inception-V3 ini adalah arsitektur yang dikembangkan

berdasarkan Convolutional Neural Network [12]. Dalam *framework* keras terdapat juga beberapa macam *optimizer*. Salah satu diantaranya adalah *optimizer Adam* yaitu algoritma yang merupakan perkembangan dari algoritma *Stochastic Gradient Descent* (SGD) klasik dimana bobot *network* telah diperbarui. Algoritma ini pertama kali dikenalkan oleh Diederik Kingma[13].

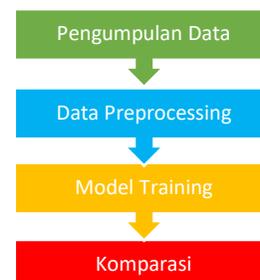
Dalam penggunaan algoritma *optimizer Adam*, perlu penentuan besar *learning rate* yang akan digunakan dalam melakukan training dataset. *Learning rate* adalah salah satu parameter training yang ditetapkan untuk menghitung nilai koreksi bobot pada waktu proses training. Memilih *learning rate* memang merupakan tantangan tersendiri karena jika nilai terlalu kecil maka training akan membutuhkan waktu yang lama sebaliknya jika terlalu besar maka pembelajaran akan menjadi kurang optimal karena terlalu cepat dan proses training menjadi tidak stabil.

Batch size juga perlu ditentukan dalam training. *Batch size* adalah istilah yang digunakan dalam pembelajaran mesin dan mengacu pada jumlah contoh pelatihan yang digunakan dalam satu iterasi dan merupakan salah satu *hyperparameter* terpenting untuk disesuaikan dengan sistem *deep learning*.

Besarnya *learning rate* dan *batch size* yang dimasukkan tentu akan berdampak pada hasil proses *deep learning*. Penelitian ini bertujuan untuk mengklasifikasi *image* covid atau non covid (normal) dimana dalam penelitian ini akan dibandingkan hasil akurasi antara beberapa macam angka *learning rate* dan ukuran *batch size*. Hal ini dimaksudkan untuk mengetahui bagaimana efek hasil pada proses *deep learning* khususnya pada pengukuran hasil akurasi dan *loss* pada *training* dan validasi.

II. METODOLOGI PENELITIAN

Langkah-langkah yang dilakukan dalam penelitian ini adalah seperti yang digambarkan dalam gambar 1 berikut ini:



Gbr. 1 langkah dalam penelitian.

Untuk lebih jelasnya, berikut langkah-langkah yang dilakukan:

A. Pengumpulan Data

Dataset yang digunakan dalam penelitian ini diambil dari gabungan beberapa source dataset kaggle. Dataset kemudian dipisah menjadi 3 bagian yaitu direktori train, val dan test. Terdapat 5186 gambar dalam direktori train, 520 gambar dalam direktori test, 519 gambar dalam direktori val.

B. Preprocessing Data

Sebelum dilakukan train, data gambar yang ada dalam dataset disamakan dulu ukurannya. Caranya dengan diresize. Dalam penelitian ini, gambar diresize menjadi ukuran 299x299 pixel. Contoh gambar yang digunakan dalam dataset penelitian ini bisa dilihat pada gambar 2 dibawah ini:



Gbr. 2 kiri x-Ray non covid dan kanan x-Ray penderita covid

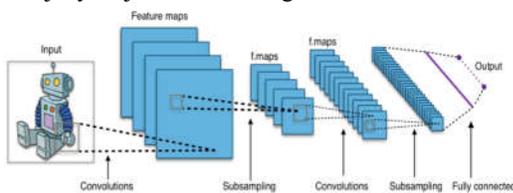
Dalam gambar 2, sebelah kiri adalah gambar x-Ray pasien normal (bukan covid) dan tidak ada gambar abu-abu di sekeliling paru-paru. Sedangkan gambar yang kanan, terlihat banyak bayang-bayang warna abu-abu di sekeliling paru-paru karena sebelah kanan adalah gambar contoh x-Ray pasien covid.

Augmentasi data dilakukan dengan cara diputar dengan range 20, zoom 0.2, shear 0.2, geser samping kanan kiri 0.2 dan geser atas bawah 0.2. sehingga walaupun sampel yang digunakan hanya sebagian, sebenarnya sudah cukup karena setiap gambar akan diaugmentasi dengan banyak cara seperti yang telah diterangkan diatas.

C. Model Training

Model training yang digunakan dalam penelitian ini menggunakan arsitektur Convolutional Neural Network Inception-V3. Algoritma yang digunakan untuk *optimizernya* adalah menggunakan *optimizer* Adam. *Loss* yang digunakan adalah *binary cross entropy*.

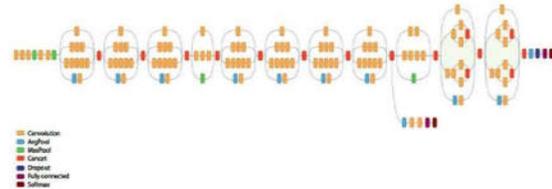
Pada dasarnya inception-V3 dikembangkan berdasarkan arsitektur Convolutional Neural Network (CNN). Prinsip dan cara kerjanya dijelaskan dalam gambar 3 di bawah ini:



Gbr. 3 Convolutional Neural Network (CNN)[14]

Yang pertama kali dilakukan adalah mengolah inputan gambar dengan melakukan konvolusi pada gambar inputan. Hasil keluarannya akan menjadi input bagi proses selanjutnya. Begitu seterusnya hingga dihasilkan output akhir.

Inception-V3 menggunakan bobot Imagenet. Arsitekturnya digambarkan dalam gambar 4 berikut ini:



Gbr. 4 Arsitektur Inception-V3[15]

Dalam arsitektur inception-V3 dalam gambar 4 terdapat beberapa macam langkah dilakukan diantaranya adalah konvolusi, Average Pool, MaxPool, dropout, fully connected, softmax. Dalam penelitian ini pada dasarnya menggunakan langkah yang sama. Hanya saja untuk fungsi aktivasi yang digunakan adalah relu dan sigmoid. Karena klasifikasi dalam penelitian ini merupakan klasifikasi dua kelas atau yang biasa disebut dengan *binary classification*.

Spesifikasi Hardware yang digunakan dalam penelitian ini menggunakan Intel-Core i7 7700HQ CPU @ 2.80GHz, 16 Gb DDR4 SDRAM, NVIDIA GPU dengan CUDA 9 dan CuDNN 7.4.

Software yang digunakan dalam penelitian ini menggunakan sistem operasi Windows 10, python yang digunakan adalah versi 3.6 dengan tensorflow versi 1.12.

D. Komparasi

Hasil akhir *deep learning* dipengaruhi oleh banyak hal. Bukan hanya oleh *learning rate* dan *batch size* saja. Akan tetapi, dalam penelitian ini difokuskan hanya pada analisa hasil komparasi beberapa *learning rate* dan *batch size* saja. Oleh karena itu, dalam tahapan ini akan dibagi menjadi dua bagian komparasi yaitu:

1) *Membandingkan Learning rate*: *learning rate* yang akan dibandingkan adalah: 0.01, 0.001, 0.0001, 0.00001, 0.000001 dan 0.0000001. dalam membandingkan *learning rate* ini, hanya akan menggunakan sebagian sampel training dan *validation*, yaitu 500 sampel training, 100 sampel *validation* dan 100 sampel testing. Dipilih sampel sebagian ini karena ketika uji coba, hasil yang diperoleh lebih terlihat beda efeknya dibandingkan ketika menggunakan keseluruhan sampel baik sampel training, *validation* dan testing.

2) *Membandingkan Batch size*: *Batch size* yang dibandingkan ada tiga macam yaitu 5, 10 dan 15. Dalam membandingkan *batch size* ini, sampel yang digunakan juga sama seperti ketika membandingkan *learning rate*, yaitu 500 sampel training, 100 sampel *validation* dan 100 sampel testing.

Untuk pengukuran hasil komparasi akan menggunakan pengukuran *loss* dan akurasi. *Loss* yang digunakan dalam

penelitian ini adalah binary cross entropy. Secara matematis, rumus dapat ditulis sebagai berikut:

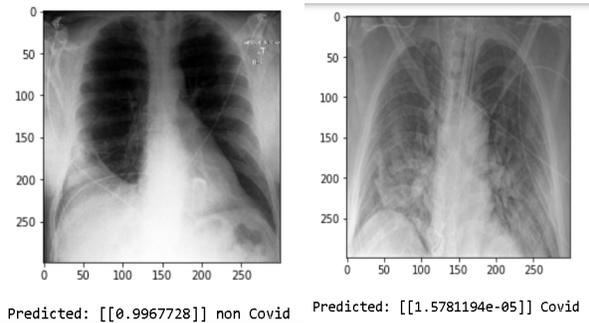
$$\text{LogLos} = \frac{-1}{N} \sum_{i=1}^N y_i * \log(p(y_i)) + (1 - y_i) * \log(1 - p(y_i)) \dots\dots\dots(1)$$

Dimana:
 N = jumlah sampel
 y_i = label
 $p(y_i)$ = prediksi

Untuk pengukuran *loss*, semakin kecil hasilnya semakin baik. Berbeda dengan akurasi, bila angka semakin besar maka hasil semakin baik. Range nilai baik *loss* dan akurasi adalah antara 0 dan 1. Jadi tidak ada angka dibawah 0 atau nilai diatas 1.

III. HASIL DAN PEMBAHASAN

Setelah dataset diolah dan dilakukan proses training, maka hasil proses pengolahan tadi akan dicek dengan dua gambar yang mewakili gambar covid dan gambar non covid. Paengambilan testing ini menggunakan parameter epoch 20, *batch* 5 dan *learning rate* 0.0001. Hasilnya sebagai berikut:



Gbr. 5 hasil testing prediksi dua gambar

Gambar 5 adalah hasil dari prediksi model yang digunakan dalam penelitian ini. Gambar kiri diambil dari direktori testing non covid dan gambar yang kanan diambil dari direktori testing yang covid. Hasil perhitungan gambar yang kiri diprediksi sebagai non covid(normal) dengan hasil perhitungan 0.996 dan dan yang kanan hasil prediksi adalah covid dengan hasil perhitungannya adalah 0,0000157.

Komparasi dilakukan dalam dua tahap yaitu:

A. Learning rate

Komparasi *learning rate* dilakukan dengan dengan parameter epoch 20 dan *batch size* 5. Learning rate yang digunakan ada 6 yaitu: 0.01, 0.001, 0.0001, 0.00001, 0.000001 dan 0.0000001. Hasilnya bisa dilihat pada table I berikut ini:

TABEL I
 KOMPARASI LR

Lr	Loss dan akurasi
0.01	
0.001	
0.0001	
0.00001	
0.000001	

Akurasi training menggunakan garis kuning, akurasi validasi menggunakan warna biru. Training loss menggunakan garis merah dan loss validasi menggunakan garis hijau. Untuk membaca hasil, maka hasil akurasi (warna kuning dan biru) akan bagus apabila mendekati angka 1 dengan kata lain semakin tinggi angkanya maka hasil semakin baik. Sebaliknya untuk loss (warna hijau dan merah), semakin kecil angkanya mendekati nol maka hasil semakin bagus.

Dari table I dapat dilihat bahwa *Learning rate* 0.01 dan 0.001 untuk akurasi memang bagus. Kurva mendekati angka 1. Akan tetapi untuk kurva loss nilainya tinggi sekali naik turun. Bahkan ada titik kurva yang nilainya mendekati angka 1 di *learning rate* 0.001 sedangkan di *learning rate* 0.01 mencapai 0.8. Angka ini termasuk sangat tinggi. Artinya loss terlalu besar.

Hasil yang paling bagus adalah *learning rate* 0.0001 dan 0.00001. karena hasil akurasi baik training dan validasi nilai mendekati satu sedangkan loss baik training dan validasi mempunyai nilai rendah.

Untuk *learning rate* 0.000001 dan 0.0000001 masih lumayan bagus bila dibandingkan *learning rate* 0.01 dan 0.001. akan tetapi hasilnya masih belum sebgus *learning rate* 0.0001 dan 0.00001. Bisa dilihat angka kurva mayoritas masih di angka 0.2 dan 0.3. sedangkan *learning rate* 0.0001 dan 0.00001 mayoritas dibawah angka 0.2.

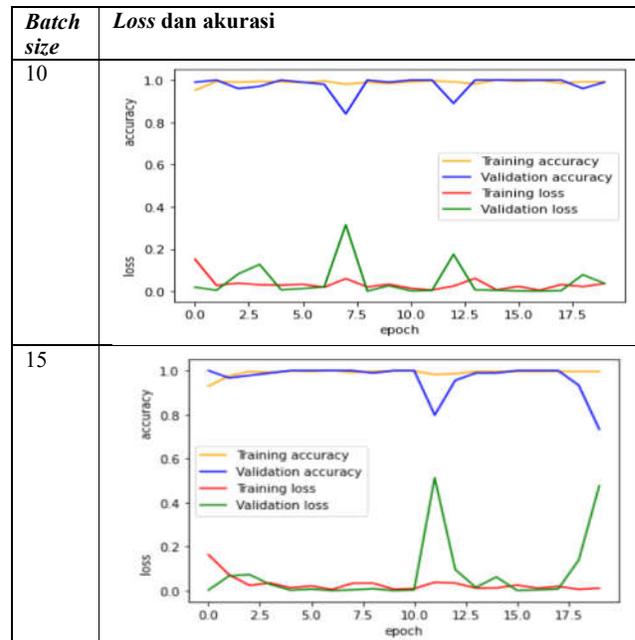
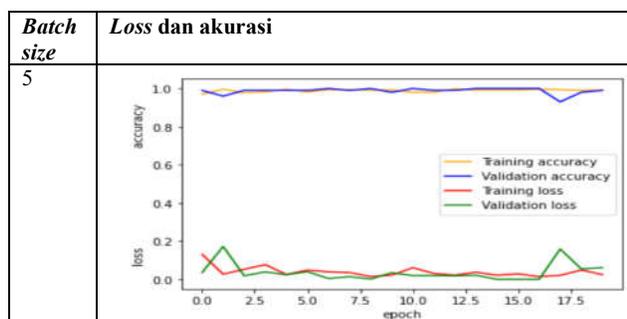
Jadi hasil yang paling optimal diantara 6 angka learning rate yang dibandingkan adalah 0.0001 dan 0.00001.

B. Batch size

Komparasi *batch size* dilakukan dengan dengan parameter epoch 20 dan *learning rate* 0.0001. Penelitian yang dilakukan ini membandingkan ukuran *batch size* kecil dengan membandingkan *batch size* 5,10 dan 15. Akan tetapi meskipun angka yang dibandingkan relative kecil, sudah cukup mewakili karena sudah dapat dilihat efek dari penggunaan *batch size* yang berbeda.

Hasilnya terdapat dalam table II berikut ini:

TABEL III
 KOMPARASI LR



Dari table II dapat dilihat bahwa semakin besar *batch size* yang digunakan, angka *loss* yang didapatkan semakin besar juga. Demikian juga sebaliknya, *batch size* yang kecil menghasilkan angka *loss* yang semakin kecil. Dari ketiga angka *batch size* yang dibandingkan, hasil yang paling optimal adalah *batch size* 5. Kurva loss pada *batch size* 5 mendekati angka nol (dibawah 0.2) sedangkan pada *batch size* 10 dan 15, ada titik-titik kurva yang *loss*nya memiliki angka yang sangat tinggi.

Biasanya *batch size* besar digunakan karena memungkinkan percepatan komputasi. Jika menggunakan ukuran *batch size* kecil akan menghabiskan waktu yang sangat lama. Akan tetapi, dibalik percepatan komputasi, ada harga yang harus dibayar. Ukuran *batch* yang terlalu besar akan memberikan hasil yang kurang optimal. Semakin besar ukuran *batch size* semakin tidak teliti hasilnya. Untuk itu para pengguna *deep learning* biasanya akan menimbang antara kemampuan tool yang dipakai, waktu yang harus diluangkan untuk proses training yang panjang dan keoptimalan hasil.

IV. KESIMPULAN

Dari percobaan yang telah dilakukan, untuk dataset yang digunakan dalam penelitian ini yang paling cocok diantara 6 *learning rate* yang diujicobakan adalah 0.0001 dan 0.00001. sedangkan *batch size* yang paling bagus dari tiga ukuran *batch size* yang diujicobakan adalah *batch size* 5. Angka dalam penelitian ini belum tentu berlaku untuk dataset yang lain. Karena bisa jadi *learning rate* yang dihasilkan disini ketika diujicobakan pada dataset dan metode yang lain bisa menghasilkan *output* yang berbeda.

Untuk yang akan datang, penelitian bisa dilakukan dengan membandingkan optimizer yang lain seperti SGD, RMSProp, NAG atau Nadam misalnya. Untuk dataset bisa menggunakan

dataset yang lebih besar, batch size dengan ukuran yang lebih bervariasi.

REFERENSI

- [1] N. Rochmawati *et al.*, "Covid Symptom Severity Using Decision Tree," in *2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE)*, Oct. 2020, pp. 1–5, doi: 10.1109/ICVEE50212.2020.9243246.
- [2] L. Barbaglia, S. Consoli, and S. Manzan, *Exploring the Predictive Power of News and Neural Machine Learning Models for Economic Forecasting*, vol. 12591 LNAI. Springer International Publishing, 2021.
- [3] S. C. Neoh *et al.*, "Intelligent facial emotion recognition using a layered encoding cascade optimization model," *Appl. Soft Comput. J.*, vol. 34, pp. 72–93, Sep. 2015, doi: 10.1016/J.ASOC.2015.05.006.
- [4] J. J. Wong and S. Y. Cho, "A local experts organization model with application to face emotion recognition," *Expert Syst. Appl.*, vol. 36, no. 1, pp. 804–819, Jan. 2009, doi: 10.1016/J.ESWA.2007.10.030.
- [5] C. Huang and Q. Zhang, "Research on Music Emotion Recognition Model of *Deep learning* Based on Musical Stage Effect," *Sci. Program.*, vol. 2021, pp. 1–10, Oct. 2021, doi: 10.1155/2021/3807666.
- [6] "Social Network Data Mining with *Deep learning* Techniques | Hindawi." <https://www.hindawi.com/journals/scn/si/504835/> (accessed Nov. 11, 2021).
- [7] N. Evans and A. Tedder, "Holographic model of hgdronization," *Phys. Rev. Lett.*, vol. 100, no. 16, 2008, doi: 10.1103/PhysRevLett.100.162003.
- [8] W. Yustanti, N. Rahmawati, and Y. Yamasari, "Klastering Wilayah Kota/Kabupaten Berdasarkan Data Persebaran Covid-19 Di Propinsi Jawa Timur dengan Metode K-Means," *JIEET (Journal Inf. Eng. Educ. Technol.)*, vol. 4, no. 1, pp. 1–9, Jun. 2020, doi: 10.26740/JIEET.V4N1.P1-9.
- [9] H. P. A. Tjahyaningtjas *et al.*, "Brain Tumor Classification in MRI Images Using En-CNN," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 4, pp. 437–451, 2021, doi: 10.22266/ijies2021.0831.38.
- [10] P. Lakhani, D. L. Gray, C. R. Pett, P. Nagy, and G. Shih, "Hello World *Deep learning* in Medical Imaging," doi: 10.1007/s10278-018-0079-6.
- [11] "Keras: the Python *deep learning* API." <https://keras.io/> (accessed Nov. 09, 2021).
- [12] M. Yani, B. Irawan, and C. Setiningsih, "Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail," *J. Phys. Conf. Ser.*, vol. 1201, no. 1, 2019, doi: 10.1088/1742-6596/1201/1/012052.
- [13] "Keras: Introduction to the Adam Optimization Algorithm - OnnoCenterWiki." https://lms.onnocenter.or.id/wiki/index.php/Keras:_Introduction_to_the_Adam_Optimization_Algorithm (accessed Nov. 09, 2021).
- [14] "Convolutional Neural Networks in Python - DataCamp." <https://www.datacamp.com/community/tutorials/convolutional-neural-networks-python> (accessed Nov. 09, 2021).
- [15] B. Mustafa, D. Keysers, and N. Houlsby, "S Calable T Ransfer L Earning," no. July, pp. 1–12, 2021, doi: 10.5121/ijaia.2019.11404.