

## DETEKSI COVID-19 PADA CITRA SINAR-X DADA MENGGUNAKAN DEEP LEARNING YANG EFISIEN

Novanto Yudistira<sup>\*1</sup>, Agus Wahyu Widodo<sup>2</sup>, Bayu Rahayudi<sup>3</sup>

<sup>1,2,3</sup> Laboratorium Sistem Cerdas, Fakultas Ilmu Komputer, Universitas Brawijaya

Email: <sup>1</sup>yudistira@ub.ac.id, <sup>2</sup>a\_wahyu\_w@ub.ac.id, <sup>3</sup>ubay1@ub.ac.id

<sup>\*</sup>Penulis Korespondensi

(Naskah masuk: 20 Juni 2020, diterima untuk diterbitkan: 26 November 2020)

### Abstrak

Deteksi Covid-19 merupakan tahapan penting untuk mengenali secara dini pasien terduga Covid-19 sehingga dapat dilakukan langkah lanjutan. Salah satu cara pendeteksian adalah melalui citra sinar-x paru. Namun demikian, selain dibutuhkan suatu model algoritma yang dapat menghasilkan akurasi tinggi, komputasi yang ringan merupakan hal yang dibutuhkan sehingga dapat diaplikasikan dalam alat pendeteksi. Model deep CNN dapat melakukan deteksi dengan akurat namun cenderung memerlukan penggunaan memori yang besar. CNN dengan parameter yang lebih sedikit dapat menghemat *storage* maupun penggunaan memori sehingga dapat berproses secara real time baik berupa alat pendeteksi maupun sistem pengambilan keputusan via *cloud*. Selain itu, CNN dengan parameter yang lebih kecil juga dapat untuk diaplikasikan pada FPGA dan perangkat keras lainnya yang mempunyai kapasitas memori terbatas. Untuk menghasilkan deteksi COVID-19 pada citra sinar-x paru yang akurat namun komputasinya juga ringan, kami mengusulkan arsitektur CNN kecil namun handal dengan menggunakan teknik pertukaran *channel* yang disebut ShuffleNet. Dalam penelitian ini, kami menguji dan membandingkan kemampuan ShuffleNet, EfficientNet, dan ResNet50 karena mempunyai jumlah parameter yang lebih kecil dibanding CNN pada umumnya seperti VGGNet atau FullConv yang menggunakan lapisan konvolusi secara penuh namun mempunyai kemampuan deteksi yang mumpuni. Kami menggunakan 1125 citra sinar-x dan mencapai akurasi 86.93 % dengan jumlah parameter model yang 18.55 kali lebih sedikit dari EfficientNet dan 22.36 kali lebih sedikit dari ResNet50 untuk mendeteksi 3 kategori yaitu Covid-19, Pneumonia, dan normal melalui uji *5-fold crossvalidation*. Memori yang diperlukan oleh masing-masing arsitektur CNN tersebut untuk melakukan sekali deteksi berhubungan secara linier dengan jumlah parameternya dimana ShuffleNet hanya memerlukan memori GPU sebesar 0.646 GB atau 0.43 kali dari ResNet50, 0.2 kali dari EfficientNet, dan 0.53 kali dari FullConv. Lebih lanjut, ShuffleNet melakukan deteksi paling cepat yaitu sebesar 0.0027 detik.

**Kata kunci:** COVID-19, deteksi, CNN, deep learning, efisien

## COVID-19 DETECTION ON X-RAY IMAGES USING EFFICIENT DEEP LEARNING

### Abstract

*Covid-19 detection is an important step in identifying early patients with suspected Covid-19 so that further steps can be taken. One way of detection is through pulmonary x-ray images. However, besides requiring an algorithm model that can produce high accuracy, lightweight computation is needed so that it can be applied in a detector. The deep CNN model can detect accurately but tends to require large memory usage. CNN with fewer parameters can save storage and memory usage so that it can process in real time both in the form of detection devices and decision-making systems via the cloud. In addition, CNN with smaller parameters can also be applied to FPGA and other hardware that have limited memory capacity. To produce accurate COVID-19 detection on x-ray images with lightweight computation, we propose a small but reliable CNN architecture using a channel shuffle technique called ShuffleNet. In this study, we tested and compared the capabilities of ShuffleNet, EfficientNet, and ResNet because they have a smaller number of parameters than usual deep CNN, such as VGGNet or FullConv which uses a full convolution layers with a robust detection capability. We used 1125 x-ray images and achieved an accuracy of 86.93% with a number of model parameters of 18.55 times less than EfficientNet and 22.36 times less than ResNet50 to detect 3 categories namely Covid-19, Pneumonia, and normal through the 5-fold cross validation. The memory required by each CNN architecture to perform one detection is linearly related to the number of parameters where ShuffleNet only requires GPU memory of 0.646 GB or 0.43 times that of ResNet50, 0.2 times of EfficientNet, and 0.53 times of FullConv. Furthermore, ShuffleNet performs the fastest detection at 0.0027 seconds.*

**Keywords:** COVID-19; detection; CNN; deep learning; efficient

## 1. PENDAHULUAN

Penyakit COVID-19 (*Coronavirus Disease 2019*) pertama kali diidentifikasi di provinsi Hubei Cina melalui adanya laporan jenis Pneumonia yang tidak diketahui penyebabnya. Semenjak 31 Desember 2019, COVID-19, yang mana virus tersebut yaitu SARS-CoV-2, telah menyebar cepat hingga menjadi sebuah pandemi baru (F. Wu, 2020)(C.Huang, 2019)(WHO,2020). Jenis virus baru ini mulai menyebar dari Wuhan ke sebagian besar Cina dalam rentang 30 hari (Z. Wu, 2020) meskipun penyebarannya itu tidak signifikan dibanding yang terjadi di dalam provinsi Hubei itu sendiri. Sedangkan di Amerika Serikat, tujuh kasus pertama dilaporkan terjadi pada tanggal 20 Januari 2020 hingga mencapai lebih dari 300.000 kasus pada tanggal 5 April 2020 (M.L. Holshue, 2020). Virus ini tidak hanya dapat menular antar manusia akan tetapi juga udara (Zhang, Renyi, 2020). Termasuk virus jenis ini adalah Virus SARS-CoV dan MERS-CoV yang dapat menyebabkan sindrom pernafasan akut parah hingga dapat menyebabkan kematian pada manusia. Beberapa penyakit penyerta COVID-19 adalah demam, batuk, tenggorokan sore, sakit kepala, kelelahan, nyeri otot, dan sesak napas (T. Singhal, 2020).

Teknik pengujian paling umum yang saat ini digunakan untuk diagnosis COVID-19 adalah reaksi berantai transkripsi-polimerase terbalik (RT-PCR). Namun pengujian secara RT-PCR mempunyai sensitivitas yang rendah dalam mendeteksi Pneumonia COVID-19. Sebagai alternatifnya, citra radiologi dada seperti computed tomography (CT) dan sinar-x dapat membantu mendiagnosis secara dini penyakit ini (Z.Y. Zu, 2020). Pasien biasanya memiliki citra CT yang normal dalam rentang 0-2 hari pertama (A. Bernheim, 2020). Dalam sebuah penelitian pada citra CT paru-paru pasien yang selamat dari penyakit paru-paru yang bernama COVID-19 Pneumonia ini, tanda-tanda penyakit secara signifikan muncul sepuluh hari setelah timbul gejala (F. Pan, 2020).

Para peneliti menyatakan bahwa penggabungan fitur citra klinis dengan hasil laboratorium dapat membantu dalam deteksi dini COVID-19 (H. Shi, 2020)(W. Zhao, 2020)(Y. Li, 2020). Citra radiologi yang diperoleh pasien dengan kasus COVID-19 terdapat informasi yang berguna untuk diagnosa. Pada beberapa penelitian ditemukan bahwa terjadi beberapa perubahan pada foto *rontgen* dada antara sebelum dan sesudah gejala COVID-19 (J. W. Chan, 2020). Kong dkk (W. Kong, 2020) menemukan bahwa terjadi kekeruhan ruang udara *infrahilar* kanan pada pasien COVID-19. Yoon dkk (S.H. Yoon, 2020) melaporkan bahwa satu dari tiga pasien yang diteliti didapati kekeruhan pada *nodular* tunggal di daerah paru-paru kiri bawah. Sedangkan di bagian lainnya terdapat empat dan lima kekeruhan yang tidak teratur di kedua paru-paru.

Pemanfaatan kecerdasan buatan akan membantu mengurangi dampak dari kurangnya alat tes RT-PCR sehingga meminimalisir biaya dan waktu tunggu pengujian. Sedangkan citra radiologi telah banyak digunakan dalam pencitraan medis sehingga dapat bermanfaat pula dalam mendeteksi COVID-19. Wang dan Wong (L. Wang, 2020) mengajukan model pembelajaran dalam untuk mendeteksi COVID19 dan meraih akurasi 92,4% untuk mengklasifikasi normal, Pneumonia, dan COVID-19. Hemdan dkk (E.E.D. Hemdan, 2020) mengajukan model bernama COVIDX-Net yang terdiri atas 7 model CNN. Narin dkk (A. Narin, 2020) mencapai deteksi 98% untuk citra *rontgen* dada dengan menggunakan model ResNet50 sedangkan Sethy dan Behera (P.K. Sethi) mengkombinasikan fitur ResNet50 dengan pengklasifikasi SVM untuk mencapai akurasi terbaik. Ada beberapa penelitian dengan memanfaatkan modaliti CT untuk mendeteksi COVID-19 (Y. Song, 2020)(S. Wang, 2020). Salah satu permasalahan yang muncul adalah tes diagnostik yang dilakukan setelah 5–13 hari pada pasien yang sebenarnya sudah pulih, kembali ditemukan positif COVID-19 (L. Lan, 2020). Temuan penting ini menunjukkan kepada kita bahwa pasien yang pulih dapat terus menyebarkan virus. Oleh karena itu, diperlukan metode diagnosis yang lebih akurat. Salah satu kelemahan paling penting dari analisis radiografi dada adalah ketidakmampuan untuk mendeteksi tahap awal COVID-19 karena mereka tidak memiliki sensitivitas yang cukup tinggi dalam pendeteksian (Z.Y. Zu, 2020). Namun, model pembelajaran seperti CNN (*Convolutional Neural Network*) yang dilatih dengan baik dapat mendeteksi dengan berfokus pada titik-titik yang tidak terlihat oleh mata manusia. Lebih jauh lagi, model CNN yang akurat dengan jumlah parameter minimum diperlukan agar dapat diaplikasikan pada alat atau komputer dengan tidak membebani komputasi.

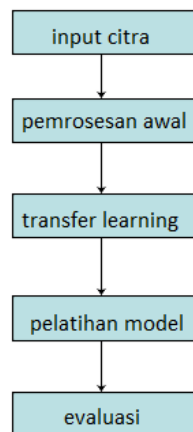
Dalam studi ini, model pembelajaran CNN yang mendalam dan efisien diusulkan untuk mendiagnosis secara otomatis COVID-19. Model yang diusulkan memiliki arsitektur yang tidak membutuhkan ekstraksi fitur apapun melainkan citra mentah yang diproses langsung oleh CNN yang efisien dan efektif menggunakan model pengacakan *channel* (*channel shuffle*) di dalamnya.

## 2. METODE PENELITIAN DAN MATERIAL



Gambar 1. Dari kiri ke kanan: citra sinar-x dada pasien yang terpapar COVID-19, kondisi normal, dan terpapar pneumonia

Pada penelitian ini, dataset citra sinar-X diperoleh dari sumber berbeda yang merupakan hasil dari diagnosis COVID-19 yaitu dari Cohen JP (J.P. Cohen, 2020) dan Wang et al (X. Wang, 2017). Data citra sinar-x dada tersebut dapat diunduh di alamat <https://github.com/muhammedtalo/COVID-19>. Repositori data citra sinar-x COVID-19 dikembangkan oleh Cohen JP (J.P. Cohen, 2020) untuk mengumpulkan data dari berbagai sumber atau secara *crowdsourcing* dengan para peneliti dari berbagai negara. Repositori ini terus diperbarui sehingga total terdapat 1125 citra yang terdiri atas 125 citra pasien positif COVID-19, 500 citra normal, dan 500 citra Pneumonia. Dari 125 citra pasien yang positif COVID-19, terdapat 43 pasien yang berjenis kelamin perempuan dan 82 pasien yang berjenis kelamin laki-laki. Gambar 1 berturut-turut menunjukkan contoh citra pasien terpapar COVID-19, kondisinya normal, dan terpapar Pneumonia yang mana secara sepintas sulit dibedakan melalui mata telanjang.



Gambar 2. Alur pelatihan

Gambar 2 menunjukkan proses pelatihan yang diawali dari input citra dan kemudian dilakukan pemrosesan awal berupa penyeragaman ukuran spasial ke 256x256 serta normalisasi. Pada dewasa ini, umum pemanfaatan *transfer learning* sebagai suatu strategi untuk meningkatkan akurasi dengan menggunakan parameter awal berupa bobot yang bersumber dari hasil pelatihan dari dataset lain pada model yang sama. Secara spesifik, bobot sumber adalah bobot hasil dari pelatihan pada dataset yang lebih besar seperti halnya ImageNet (Deng, 2009) dan mempunyai kemiripan karakteristik citra input. Pada penelitian ini kita melakukan *transfer learning* dari ImageNet yang terdiri dari 14 juta citra natural serta 1000 kategori. Selayaknya jaringan syaraf tiruan, model CNN juga menggunakan teknik propagasi maju dan propagasi balik. Propagasi maju dimulai dari input citra kemudian dilakukan filter berjenjang sesuai dengan kernel *convolution* pada setiap jenjangnya. Selain itu, pada setiap jenjangnya terdapat juga fungsi *pooling* yang mereduksi ukuran spasial inputnya. Proses propagasi maju kemudian

diakhiri dengan *fully connected layer* (fc) yang merupakan tahapan pengklasifikasian untuk menentukan bobot akhir input sesuai kategorinya. Kemudian, propagasi balik akan melakukan *update* terhadap bobot-bobot yang ada dalam kernel CNN dan nilai bobot fc-nya berdasar pada *error* yang dihasilkan dari fungsi *loss* dengan menggunakan teknik *chain rule*. Secara umum, ada 3 kategori yang akan digunakan sebagai tujuan klasifikasi yaitu citra sinar-x pasien yang terpapar COVID-19, citra pasien normal, dan citra pasien Pneumonia.

Pelatihan model dilakukan dengan iterasi selama 20 *epoch* menggunakan optimasi Adam dan learning rate 0.0001. Untuk mereduksi overfitting, digunakanlah *regularizer* berupa *weight decay* senilai 0.00001. Ukuran *batch* yang digunakan adalah 4 untuk menghemat memori selama pelatihan. *Batch* merupakan sekelompok data yang diproses oleh model pada setiap iterasinya. Fungsi *loss* yang digunakan adalah *softmax crossentropy* yang memperhitungkan bobot akhir dengan normalisasi softmax (0-1) dan jarak terhadap label.

Evaluasi dilakukan dengan *confusion matrix* dengan melihat akurasi secara global antar kategori yang bertujuan untuk melihat *true positive*, *true negative*, *false positive*, dan *false negative*-nya. Selain itu juga dilakukan evaluasi penggunaan memori (*Graphical Processing Unit*) GPU serta kecepatan deteksinya. Hasil akurasi dari model yang diajukan dikomparasi dengan model lainnya seperti ResNet50 (He, 2016) dan EfficientNet (Tan, 2019). EfficientNet yang digunakan sebagai komparasi adalah EfficientNet b5. Pelatihan dan pengujian dilakukan pada komputer dengan spesifikasi prosesor Intel Core i5-9300H CPU 2.4 GHz, RAM 8 GB, dan GPU NVIDIA GeForce GTX 1050 4GB. Sedangkan pada sisi perangkat lunak digunakan sistem operasi Microsoft Windows Home 10, *Integrated Development Environment* (IDE) Spyder 3, bahasa pemrograman Python 3, manajemen paket Anaconda 2, dan *library* Pytorch 1.6.

### 3. DEEP LEARNING EFISIEN (FullConv vs ShuffleNet)

Tabel 1. Arsitektur Full CNN (FullConv)

Layer	Output size	KSize	Stride	Repeat	Output channels (g groups 2)
Image	256x256				3
Conv1	128x128	3x3	2	1	24
MaxPool	64x64	3x3	2		
Conv2	32x32	1x1	1	1	116
Conv3	16x16	1x1	1	1	232
Conv4	8x8	1x1	1	1	464
Conv5	8x8	1x1	1	1	1024

Layer	Output size	KSize	Stride	Repeat	Output channels (g groups 2)
Global Pool	1856x1856	7x7			
FC					3
Jumlah parameter					10.9 juta

Tabel 2. Arsitektur ShuffleNet v2 1.0

Layer	Output size	KSize	Stride	Repeat	Output channels (g groups 2)
Image	256x256				3
Conv1	128x128	3x3	2	1	24
MaxPool	64x64	3x3	2		
Stage 2	32x32		2	1	116
	32x32		1	2	116
Stage 3	16x16		2	1	232
	16x16		1	4	232
Stage 4	8x8		2	1	464
	8x8		1	2	464
Conv5	8x8	1x1	1	1	1024
Global Pool	1x1	7x7			
FC					3
Jumlah parameter					1.2 juta

Munculnya teknologi pembelajaran mesin yang mendalam seperti CNN telah merevolusi kecerdasan buatan dalam beberapa tahun belakangan ini. Kata mendalam mengacu pada peningkatan besarnya jaringan berdasarkan jumlah lapisan atau tahapan. Secara khusus, didalam CNN terdapat struktur operasi matematika bernama *convolution* yang berlapis-lapis. Struktur CNN memiliki lapisan *convolution* yang mengekstraksi fitur dari input dengan filter yang pada dasarnya merupakan bentuk estimasi. Dengan estimasi ini CNN mampu mengurangi beban kinerja komputasi, diikuti lapisan-lapisan berikutnya yang sepenuhnya terhubung layaknya jaringan saraf. Dengan menghubungkan satu atau lebih lapisan seperti itu, model CNN yang dalam terbentuk dan parameter internalnya disesuaikan untuk melakukan tugas tertentu, seperti klasifikasi atau pengenalan objek. Alih-alih memulai pengembangan model yang mendalam dari awal, pendekatan yang lebih rasional adalah membangun jaringan menggunakan model yang sudah terbukti. Oleh karena itu, saat merancang model mendalam yang digunakan dalam penelitian ini, model ShuffleNet (Zhang, 2018) yang menggunakan unit pengacakan *channel* dipilih sebagai investigasi awal.

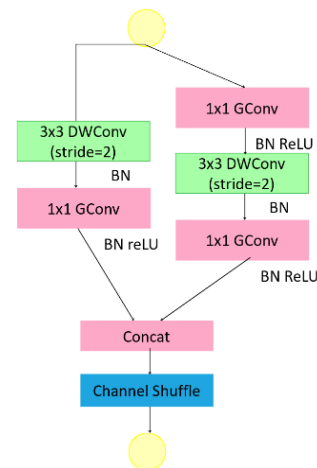
Apabila kita bandingkan dengan model yang menggunakan CNN secara penuh yang dinamakan dengan arsitektur FullConv seperti tampak pada tabel 1, maka jumlah parameter yang dihasilkan oleh ShuffleNet lebih sedikit dibandingkan dengan FullConv dengan perbandingan 1:9. Banyaknya lapisan yang digunakan dalam FullConv sama dengan ShuffleNet yaitu 5 lapisan. Mulai lapisan ke 2 sampai

ke 5 pada FullConv menggunakan filter *convolution* dengan ukuran kernel 1x1 dengan jumlah *channel* yang sama dengan ShuffleNet yaitu masing-masing sebanyak 116, 232, 464, dan 1024. Masing-masing lapisan *convolution* ke 2 sampai 5 diganti dengan *stage shuffle* yang terdiri atas skema *convolution*, pengacakan *channel*, normalisasi *Batch Normalization* (BN) (Ioffe, 2015), dan aktivasi *Rectified Linear Unit* (ReLU).

Dengan melakukan pengacakan *channel*, jaringan akan memperoleh keuntungan diantaranya adalah:

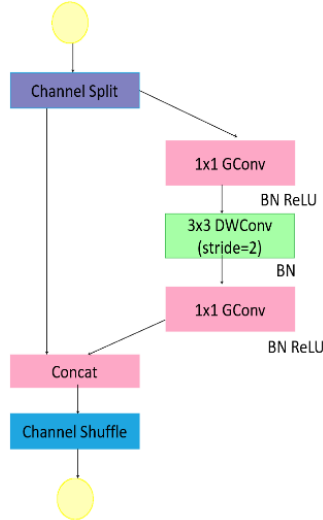
1. Jaringan akan memperoleh informasi lebih tentang struktur detail citra sehingga tidak perlu menambah jaringan lebih ke dalam.
2. Dapat bertindak sebagai *regularizer* yang mereduksi *overfitting* oleh karena output aktivasi yang lebih beragam.
3. Memperkecil kemungkinan terjadinya *vanishing gradient* yaitu fenomena dimana gradien terlalu dini bernilai sangat kecil sehingga tidak lagi terjadi *update* sebelum global optima tercapai.

Model ShuffleNet yang akan digunakan adalah versi kedua yang dikembangkan oleh Megvii dan Universitas Tsinghua (Ma, 2018). Jaringan tersebut terdiri dari 1 *convolution*, 3 *stage* (suatu tahapan yang terdiri dari *convolution* dan unit *shuffle*), 1 lapisan *convolution* akhir (Conv5), 1 lapisan *pooling* (menggunakan Maxpool), dan fc seperti ditunjukkan pada Tabel 2. Setiap tahapan lapisannya menggandakan jumlah output *channel* menjadi 2 kalinya.

Gambar 3. *Stage shuffle* ketika *stride* = 1

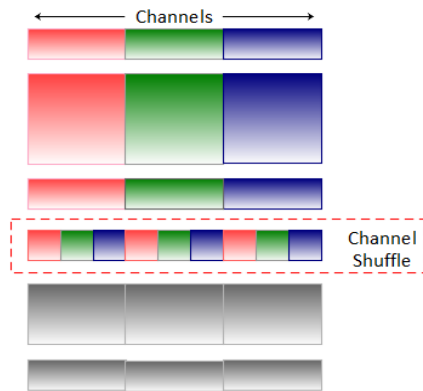
Lapisan *convolution* awal (Conv1) adalah jenis lapisan CNN yang terdiri atas 24 filter *channel* yang berukuran 3x3 dengan nilai langkah (*stride*) 2. Tahapan berikutnya adalah *stage 2, 3*, dan 4 dimana masing-masing terdiri atas dua *stage shuffle* dengan *stride*-nya masing-masing 2 dan 1 seperti yang ditunjukkan dalam Gambar 3 dan Gambar 4. *Stage shuffle* kedua dari masing-masing *stage 2,3*, dan 4 (Gambar 3) diulang 2, 4, dan 2 kali. GConv adalah

lapisan *convolution* berdasarkan titik dengan filter 1x1 dan diakhiri normalisasi BN serta fungsi aktivasi ReLU. *Depth Wise Convolution* (DWConv) adalah *convolution* 3x3 dengan *stride* 2 dan diakhiri normalisasi BN.



Gambar 4. Stage shuffle ketika *stride* = 2

### 3.1. Unit Shuffle



Gambar 5. Proses kerja unit shuffle

Proses kerja *channel shuffle* digambarkan seperti pada Gambar 5 diatas. Masing-masing grup *convolution* menerima hasil *convolution* dari grup yang lain sedemikian rupa sehingga pada setiap grup pada *channel shuffle* terdiri atas kombinasi *channel* yang berasal dari grup yang berbeda sehingga semakin menguatkan korelasi. Secara khusus, dari fitur yang dihasilkan dari lapisan grup sebelumnya, kita bisa membagi channel di setiap grup menjadi beberapa subgroup. Dengan demikian *convolution* tahap berikutnya dilakukan dengan kombinasi grup yang berbeda. Hal ini dapat diimplementasikan secara efisien dan elegan oleh unit *shuffle* seperti pada Gambar 5. Sebagai contoh, langkahnya dapat dibuat beberapa tahapan berikut:

1. Lapisan *convolutional* dengan *g* grup menghasilkan output  $g \times n$  channel. Maka apabila

dengan ukuran output aktivasinya *s*, maka total ukuran menjadi  $g \times n \times s \times s$

2. Dengan demikian dimensi *channel shuffle* menjadi  $(g, n, s, s)$ .
3. Lakukan transposisi sehingga dimensi *channel shuffle* menjadi  $(n, g, s, s)$
4. Kemudian lakukan penggabungan antara *g* grup dan *n* channel sehingga menjadi  $(n \times g, s, s)$
5. Menjadi input untuk lapisan *convolution* selanjutnya

Operasi *convolution* masih berlangsung bahkan seandainya pun dua grup *convolution* memiliki jumlah *channel* yang berbeda. Selain itu, bobot *channel shuffle* juga dapat dioptimasi oleh propagasi balik, yang berarti dapat ditambahkan ke dalam struktur jaringan untuk dilakukan pelatihan secara *end-to-end*. Operasi *channel shuffle* yang berulang juga memungkinkan untuk membentuk struktur jaringan yang lebih kuat karena lebih banyak *convolution* dilakukan pada tiap lapisan grup dengan kombinasi *channel* yang berbeda. Pada subbab berikutnya kami akan memperkenalkan fungsi aktivasi, normalisasi, dan fungsi *loss* yang digunakan oleh ShuffleNet.

### 3.2. Fungsi Aktivasi dan Loss

Permasalahan utama dalam penelitian ini adalah bagaimana mengklasifikasikan citra yang mempunyai detail yang halus. Model yang mampu melakukan klasifikasi pada objek yang seperti itu harus memiliki struktur yang dapat menangkap dan mempelajari perbedaan-perbedaan kecil daripada harus membangun jaringan yang lebih dalam, seperti model ResNet atau ResNext. Ilustrasi model shuffleNet pada Tabel 2 yang diusulkan dalam penelitian ini memiliki 5 lapisan *convolution* yang mana setiap *stage*-nya melakukan *convolution* maksimum 3 kali dan *stage shuffle* kedua dari setiap *stage* melakukan repetisi sebanyak 2, 4, atau 2 kali. Pada Gambar. 2 dan 3, setiap lapisan ShuffleNet memiliki satu lapisan *convolutional* GConv diikuti oleh BN serta operasi ReLU, sedangkan DWConv diikuti normalisasi BN. Operasi BN digunakan untuk menstandarisasi input dan operasi ini memiliki manfaat lain seperti mengurangi waktu pelatihan dan meningkatkan stabilitas model. Operasi BN dapat dikalkulasi dengan persamaan berikut:

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad (1)$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (2)$$

$$\bar{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (3)$$

Proses normalisasi berawal dari persamaan (1) untuk mencari nilai rata-rata dalam satu *batch*, dimana *x* adalah inputan untuk lapisan *convolution* berikutnya, *m* adalah banyaknya data dalam satu *batch*, dan  $\mu_B$  adalah rata-rata dari *batch*. Kemudian



masuk persamaan (2) untuk menghitung variasi dalam satu *batch*, dimana  $\sigma_B$  adalah variasi dari *batch*. Persamaan (3) menghitung normalisasi akhir inputan dalam satu *batch*, dimana  $\tilde{x}_i$  adalah normalisasi dari inputan *batch* sebelum diproses ke dalam *convolution* berikutnya sehingga ketika dilakukan propagasi balik dan *update*, parameter *convolution* yang dilewati inputan tadi akan mempunyai gradien berdasar pada hasil operasi *convolution* dengan inputan yang telah dinormalisasi.

Aktivasi ReLU (Nair, 2010) digunakan untuk mematikan neuron yang tidak berkontribusi terhadap keputusan akhir. Tidak seperti fungsi aktivasi Sigmoid yang melakukan *smoothing* dengan hasil rentang 0 dan 1, ReLU menjadikan nilai nol pada bagian nilai aktivasi yang negatif. Hal ini dilakukan untuk semakin menonjolkan bagian fitur yang menunjukkan kontribusi positif terhadap keputusan akhir dan menonaktifkan fitur yang mempunyai kontribusi negatif. Ketika input  $x$  masuk ke dalam fungsi aktivasi ReLU maka secara formal dapat dituliskan sebagai berikut:

$$\max(0, x) \quad (4)$$

Pada persamaan (4) diatas,  $x$  adalah inputan untuk fungsi aktivasi ReLU sehingga outputnya memiliki nilai fitur minimal 0.

Sedangkan operasi Maxpool digunakan di semua operasi pereduksian ukuran dimensi. Maxpool memperkecil ukuran suatu input dengan mengambil nilai maksimum dari hasil tiap operasi filter. Terakhir adalah fungsi *loss* yang digunakan sebagai acuan jaringan untuk *update*. Fungsi *loss* yang digunakan adalah *softmax crossentropy* yang bekerja dengan 3 kategori untuk melihat seberapa dekat bobot keputusan sementara dengan label *groundtruth* COVID-19, normal, dan Pneumonia. Pada saat propagasi maju, bagian terakhir dari jaringan yaitu *fc* melakukan tugas klasifikasi untuk menentukan label citra input *rontgen* dada apakah sebagai COVID-19, normal, atau Pneumonia dengan mengambil nilai maksimumnya. Fungsi *softmax crossentropy* (CE) adalah sebagai berikut:

$$f(x)_c = \frac{e^{x_c}}{\sum_j e^{x_j}} \quad (5)$$

$$CE = -\sum_i (t_i \log(f(x)_i)) \quad (6)$$

Pada persamaan (5),  $x_c$  adalah output dari lapisan terakhir (*fc*) pada index ke  $c$  atau bisa dikatakan kategori ke  $c$ . Sedangkan  $e$  adalah nilai konstanta eksponensial. Pada persamaan (6),  $C$  adalah total banyaknya kategori (dalam hal ini 3). Sedangkan  $t$  adalah label *groundtruth* atau acuan yang dalam hal ini berupa angka 0,1, atau 2 yang mewakili COVID-19, normal, atau Pneumonia. Optimasi Adam kemudian akan meminimumkan nilai CE untuk semua *batch* melalui beberapa iterasi secara propagasi balik dan *chain rule*.

#### 4. HASIL DAN PEMBAHASAN

Tabel 3. Matrik confusion ShuffleNet – 5 fold cross validation

	COVID-19	normal	pneumonia
COVID-19	89.66 %	4.14 %	6.21 %
normal	0 %	95 %	4.85 %
pneumonia	3.44 %	20.43 %	76.13 %

Tabel 4. Matrik confusion EfficientNet – 5 fold cross validation

	COVID-19	normal	pneumonia
COVID-19	88.27 %	1.38 %	10.34 %
normal	0 %	93.4 %	6.6 %
pneumonia	0.43 %	18.7 %	80.86 %

Tabel 5. Matrik confusion ResNet50 – 5 fold cross validation

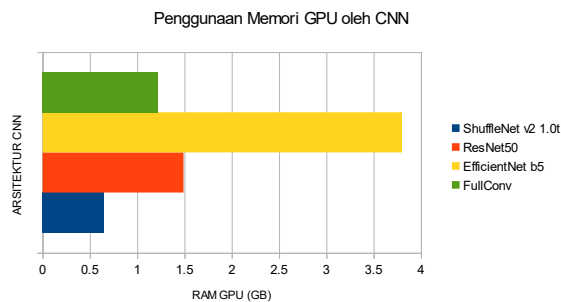
	COVID-19	normal	pneumonia
COVID-19	95.17 %	2.76 %	2.76 %
normal	0 %	95.72 %	4.47 %
pneumonia	0.21 %	18.27 %	81.5 %

Tabel 6. Akurasi deteksi model CNN

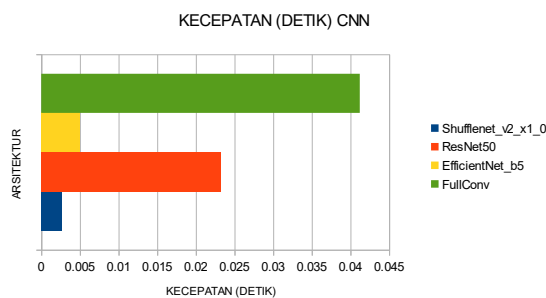
	Akurasi	# parameter
FullConv	86.93 %	10.951.059
ResNet50	90.8 %	23.514.179
EfficientNet	87.5 %	28.346.931
ShuffleNet	86.93 %	1.267.759

Kami melakukan percobaan untuk mendeteksi dan mengklasifikasikan COVID-19 menggunakan citra sinar-X dalam 3 kategori. Pertama, kami telah melatih model pembelajaran mendalam ShuffleNet, EfficientNet b5 untuk mengklasifikasikan citra sinar-X ke dalam tiga kategori: COVID-19, Normal, Pneumonia. Kinerja model yang diusulkan dievaluasi menggunakan prosedur validasi silang 5 kali (5-fold crossvalidation) dan dalam setiap percobaannya, pelatihan dan klasifikasi ke dalam 3 kategori tersebut dilakukan. Perbandingan data latih dan uji citra sinar-X pada setiap percobaan adalah 80:20. Semua bagian yang telah dibagi dalam data latih dan uji digunakan pada setiap tahap pelatihan dan pengujian. Kami telah melatih model selama 20 epoch. Hasil pengujian klasifikasi 3 kategori dirangkum dalam matrik *confusion* untuk mengetahui akurasinya secara presisi dan *recall* dengan merata-ratakan hasil dari 5 kali percobaan tersebut. Tabel 3 menunjukkan hasil klasifikasi ShuffleNet v2 dimana untuk data COVID-19, normal, dan Pneumonia terdeteksi akurasi berturut-turut sebesar 89.66%, 95%, dan 76.13%. *Confusion* paling besar terjadi antara Pneumonia dengan normal sebesar 20.43%. Sedangkan pada EfficientNet, sebagaimana diperlihatkan pada Tabel 4, COVID-19, normal, dan pneumonia terdeteksi akurasi masing-masing sebesar 88.27%, 93.4%, dan 80.86%. *Confusion* terbesar terjadi antara Pneumonia dengan normal sebesar 18.7%. Tabel 5 menunjukkan hasil klasifikasi ResNet50 dimana untuk COVID-19, normal, dan Pneumonia terdeteksi akurasi masing-masing sebesar 95.17%, 95.72%, dan 81.5%. *Confusion* paling besar juga terjadi antara Pneumonia

dengan normal sebesar 18.27%. *Confusion* yang terjadi antara Pneumonia dan normal untuk semua pengujian akurasi pada semua model menunjukkan adanya kemiripan karakteristik citra antara kedua kategori tersebut. Lebih lanjut hal ini menunjukkan pula bahwa citra sinar-X dada dari pasien yang terkonfirmasi COVID-19 terdapat pola khusus yang berbeda dengan pasien yang terkena Pneumonia maupun normal. Apabila *True Postive* (TP) ketiga kategori tersebut dirata-ratakan maka ShuffleNet v2 meraih akurasi 86.93% yang mana lebih baik 0.99 kali dari EfficientNet dan 0.95 kali dari ResNet50 atau dengan kata lain akurasinya sedikit lebih rendah dari lainnya seperti yang ditunjukkan pada tabel 6. Namun demikian, ShuffleNet mempunyai jumlah parameter yang 18.55 kali lebih sedikit dari EfficientNet dan lebih sedikit 22.36 kali dari ResNet50.



Gambar 6. Perbandingan penggunaan memori GPU diantara arsitektur CNN



Gambar 7. Perbandingan kecepatan deteksi diantara arsitektur CNN

Gambar 6 menunjukkan perbandingan penggunaan memori GPU yang dihabiskan oleh arsitektur ShuffleNet, ResNet50, EfficientNet, dan FullConv. Memori yang diperlukan oleh masing-masing arsitektur CNN tersebut untuk melakukan sekali deteksi berhubungan secara linier dengan jumlah parameternya dimana ShuffleNet hanya memerlukan memori GPU sebesar 0.646 GB atau 0.43 kali dari ResNet50, 0.2 kali dari EfficientNet, dan 0.53 kali dari FullConv. Sebagai perbandingan dengan FullConv dimana lapisan yang kedua sampai kelima berupa *convolution*, ShuffleNet menghasilkan

tingkat akurasi yang sama 86.93% (tabel 6), namun dengan memori GPU yang dihabiskan 1:1.88 dengan FullConv. Lebih lanjut, ShuffleNet melakukan deteksi paling cepat yaitu sebesar 0.0027 detik (Gambar 7). Dibandingkan dengan FullConv, ShuffleNet mendeteksi lebih cepat dengan perbandingan 1:15.55.

## 5. KESIMPULAN

Model CNN ShuffleNet mampu menghasilkan akurasi deteksi COVID-19 dengan akurasi sedikit dibawah dari model yang parameternya lebih banyak seperti EfficientNet dan ResNet50. Namun demikian, model yang diajukan mempunyai jumlah parameter yang jauh lebih sedikit 18.55 kali dari EfficientNet dan 22.36 kali dari ResNet50. Selain itu, ShuffleNet menghabiskan memori GPU paling sedikit sebesar 0.646 GB serta waktu deteksi tercepat sebesar 0.0027 detik, sehingga memungkinkan untuk diaplikasikan pada perangkat keras maupun aplikasi *mobile*. Model yang diajukan telah dilatih dan divalidasi pada dataset sebanyak 1125 citra sinar-x dada dengan 3 kategori sehingga bisa dikatakan valid dan akan lebih baik lagi apabila lebih banyak dilatih dengan data baru. Model yang telah dilatih ini apabila telah diaplikasikan pada perangkat lunak maupun keras dapat menjadi pelengkap uji spesimen atau RT-PCR yang berguna untuk membantu pendeteksian COVID-19 secara masal.

## DAFTAR PUSTAKA

- A. BERNHEIM, X. MEI, et al. (2020). Chest CT findings in coronavirus disease-19 (COVID-19): relationship to duration of infection. *Radiology*, <https://doi.org/10.1148/radiol.20200463>. In press.
- A. NARIN, C. KAYA, Z. PAMUK. (2020). Automatic Detection of Coronavirus Disease (COVID- 19) Using X-Ray Images and Deep Convolutional Neural Networks. *arXiv preprint arXiv:2003.10849*.
- C. HUANG, Y. WANG, et al. (2020). Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *Lancet* 395 (10223), p.497–506.
- DENG, JIA, et al. (2009). Imagenet: A large-scale hierarchical image database. 2009 IEEE conference on computer vision and pattern recognition.
- E.E.D. HEMDAN, M.A. SHOUMAN, M.E. KARAR. (2020). COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images. *arXiv preprint arXiv:2003.11055*.
- F. PAN, T. YE, et al. (2020). Time course of lung changes on chest CT during recovery from 2019 novel coronavirus (COVID-19)

- pneumonia. Radiology, <https://doi.org/10.1148/radiol.2020200370>. In press.
- F. WU, S. ZHAO, B. YU, et al. (2020). A new coronavirus associated with human respiratory disease in China. *Nature* 579 (7798), p.265–269.
- H.E, KAIMING, et al. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- H. SHI, X. HAN, et al. (2020). Radiological findings from 81 patients with COVID-19 pneumonia in Wuhan, China: a descriptive study. *Lancet Infect. Dis.* 24 (4), p.425–434.
- IOFFE, SERGEY, dan CHRISTIAN SZEGEDY. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.
- J.F.W. CHAN, S. YUAN, et al. (2020). A familial cluster of pneumonia associated with the 2019 novel coronavirus indicating person-to-person transmission: a study of a family cluster. *Lancet* 395 (10223), p.514–523.
- J.P. COHEN. (2020). COVID-19 Image Data Collection. <https://github.com/ieee8023/COVID-chestxray-dataset>.
- L. LAN, D. XU, G. YE, C. XIA, S. WANG, Y. LI, H. XU. (2020). Positive RT-PCR test results in patients recovered from COVID-19. *Jama* 323 (15), p.1502–1503.
- L. WANG dan A. WONG. (2020). COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest Radiography Images. *arXiv preprint arXiv:2003.09871*.
- M.A, NINGNING, et al. (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. *Proceedings of the European Conference on Computer Vision (ECCV)*.
- M.L. HOLSHUE, C. DEBOLT, et al. (2020). First case of 2019 novel coronavirus in the United States. *N. Engl. J. Med.* 328, p.929–936.
- NAIR, VINOD, dan GEOFFREY E. HINTON. (2010). Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th international conference on machine learning (ICML-10)*.
- P.K. SETHY dan S.K. BEHERA. (2020). Detection of Coronavirus Disease (COVID-19) Based on Deep Features.
- S.H. YOON, K.H. LEE, et al. (2020). Chest radiographic and CT findings of the 2019 novel coronavirus disease (COVID-19): analysis of nine patients treated in Korea. *Korean J. Radiol.* 21 (4), p.494–500.
- S. WANG, B. KANG, J. MA, X. ZENG, M. XIAO, J. GUO, B. XU. (2020). A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19). *medRxiv*.
- TAN, MINGXING, dan QUOC V. LE. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*.
- T. SINGHAL. 2020. A review of coronavirus disease-2019 (COVID-19). *Indian J. Pediatr.* 87, p.281–286.
- W. KONG dan P.P. AGARWAL. (2020). Chest imaging appearance of COVID-19 infection. *Radiology: Cardiothoracic Imaging* 2 (1), e200028.
- WORLD HEALTH ORGANIZATION. (2020). Pneumonia of Unknown Cause—China. *Emergencies Preparedness, Response, Disease Outbreak News, World Health Organization (WHO)*.
- W. ZHAO, Z. ZHONG, X. XIE, Q. YU, J. LIU. (2020). Relation between chest CT findings and clinical conditions of coronavirus disease (COVID-19) pneumonia: a multicenter study. *Am. J. Roentgenol.* 214 (5), p.1072–1077.
- X. WANG, Y. PENG, L. LU, Z. LU, M. BAGHERI, R.M. SUMMERS. (2017). Chestx-ray8: hospitalscale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.2097–2106.
- Y. LI dan L. XIA. (2020). Coronavirus Disease 2019 (COVID-19): role of chest CT in diagnosis and management. *Am. J. Roentgenol.*, p.1–7.
- Y. SONG, S. ZHENG, L. LI, X. ZHANG, X. ZHANG, Z. HUANG, Y. CHONG. (2020). Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. *medRxiv*.
- ZHANG, RENYI, et al. (2020). Identifying airborne transmission as the dominant route for the spread of COVID-19. *Proceedings of the National Academy of Sciences*.
- ZHANG, XIANGYU, et al. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- Z. WU dan J.M. MCGOOGAN. (2020). Characteristics of and important lessons from the coronavirus disease 2019 (COVID-19) outbreak in China: summary of a report of 72 314 cases from the Chinese Center for Disease Control and Prevention. *Jama* 323 (13), p.1239–1242.
- Z.Y. ZU, M.D. JIANG, P.P. XU, W. CHEN, Q.Q. NI, G.M. LU, L.J. ZHANG. (2020). Coronavirus disease 2019 (COVID-19): a perspective from China. *Radiology*, <https://doi.org/10.1148/radiol.2020200490>. In press.