
Dense Net

Kelompok 7:

- Nouchka Indra Dewa - C14170021
 - Jason Lucky Harrysiswanto - C14170129
 - Leonardo Yurion Tungribali - C14170153
 - Eric Pranoto - C14180139
-

Pendahuluan

- DenseNet adalah jenis convolutional neural network yang memanfaatkan dense connection antar layer melalui Dense Block, dimana semua layer berhubungan (dengan mencocokkan ukuran fitur map) secara langsung satu sama lain.
- Dense net menghubungkan setiap layer dengan layer lain dengan feed forward fashion.
- Jika dibandingkan dengan convolutional network yang sebelum-sebelumnya, CNN sebelumnya punya L layers dengan L connection antara 1 layer dengan tiap layer dan subsequent layernya sendiri.
- Kalo Dense net punya $\frac{L(L+1)}{2}$ direct connection.

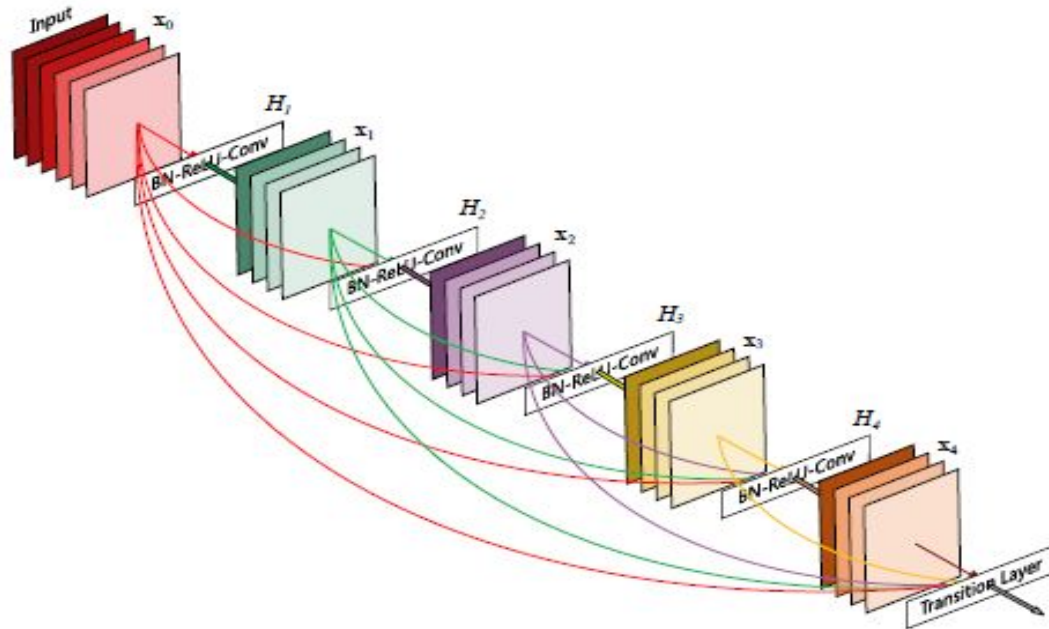


Figure 1: A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.

Berbeda dengan ResNet, DenseNet tidak pernah combine fiturnya melalui penjumlahan sebelum dimasukkan ke dalam layer, melainkan DenseNet combine fiturnya dengan cara concatenation dengan rumus $L(L+1)/2$ tadi.

How does it Works?

- Sebuah kemungkinan efek counter-intuitive dari pattern connectivity yang rapat ini adalah membutuhkan lebih sedikit parameter daripada traditional CNN. Hal ini dikarenakan tidak perlu relearn fitur maps yang berlebihan.
 - Arsitektur dari traditional feed-forward dapat dilihat sebagai sebuah algoritma dengan state yang diteruskan dari layer ke layer. Di tiap layer, state dari layer sebelumnya akan dibaca lalu diteruskan ke layer selanjutnya. Dengan demikian, state tersebut akan berubah dan perubahan tersebut juga ikut diteruskan dengan informasi baru yang diperlukan untuk mengembangkan state tersebut.
 - Berbeda dengan ResNet, arsitektur dari DenseNet ini secara eksplisit akan membedakan mana informasi yang harus ditambahkan ke network dan mana yang harus disimpan.
 - Layer dari DenseNet sangatlah sempit (misal 12 fitur per layer), menambahkan hanya 1 set kecil fitur map ke “collective knowledge” dari network dan menyimpan sisa fitur map yang tidak berubah.
 - Pengklasifikasian yang terakhir adalah membuat decision berdasarkan semua fitur map yang sudah ada dalam network.
-

-
- Selain efisiensi parameter yang lebih baik, 1 keuntungan besar dari DenseNet ini adalah adanya peningkatan aliran informasi dan gradien melalui network, sehingga lebih memudahkan untuk di-train.
 - Tiap layer memiliki akses langsung ke gradien dari loss function dan original input signal, mengarah ke implicit deep supervision. Hal ini membantu training dari deeper network arsitektur.
 - Pengamatan selanjutnya adalah bahwa kerapatan connections memiliki regularizing effect, yang mana dapat mengurangi overfitting pada tasks dengan ukuran training set yang lebih kecil.
 - Setelah dievaluasi, DenseNet memiliki 4 highly competitive benchmark datasets, yaitu CIFAR-10, CIFAR-100, SVHN, dan ImageNet.
-

Related Work

- Eksplorasi dari arsitektur network telah menjadi bagian dari penelitian mengenai neural network. Pada dasarnya, DenseNet menggunakan potensi dari suatu network dari fitur reuse, yielding condensed models yang cenderung mudah untuk di-train dan memiliki efisiensi parameter yang tinggi.
 - Dengan meng-concatenate fitur maps yang dipelajari dari berbagai layer, variasi input dari lapisan berikutnya akan semakin tinggi, maka efisiensinya pun juga otomatis meningkat.
 - Hal ini menjadi perbedaan yang besar antara DenseNet dengan ResNet. Jika dibandingkan dengan Inception Networks yang juga meng-concatenate fitur dari layer-layer yang berbeda pun, bisa dikatakan bahwa DenseNet lebih sederhana dan lebih efisien.
-

Dense Connectivity

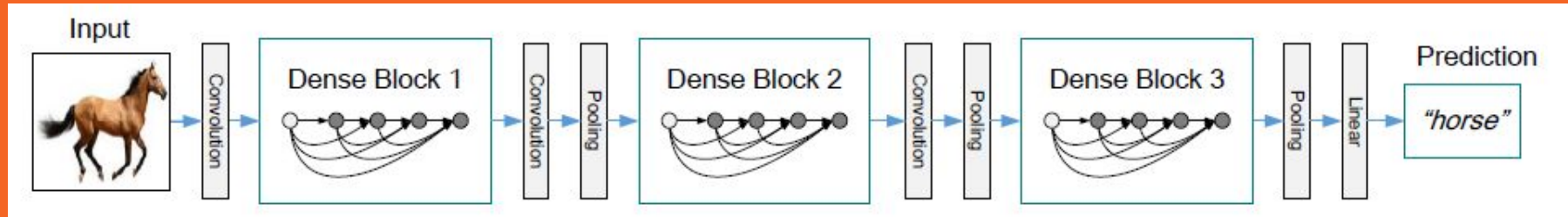
- Untuk meningkatkan informasi flow antar layer, maka dibuatlah connectivity pattern yang berbeda, yaitu direct connections dari layer manapun ke semua layer berikutnya.
- Gambar 1 tadi mengilustrasikan layout dari DenseNet yang dihasilkan secara skematis. Akibatnya, layer ke l akan menerima fitur maps dari semua layer sebelumnya, $\langle x_0, x_1, \dots, x_{l-1} \rangle$ sebagai input:

$$\mathbf{x}_l = H_l([\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{l-1}]),$$

- Dimana $[\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{l-1}]$ mengacu pada concatenation dari fitur maps yang dihasilkan pada layer 0 sampai $l-1$.
 - Karena rapatnya Connectivity tersebut, maka dinamakan Dense Convolutional Network (DenseNet).
 - Untuk kemudahan implementasi, beberapa input dari Hl di persamaan tersebut digabungkan menjadi sensor tunggal.
-

$$\mathbf{x}_\ell = H_\ell([\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{\ell-1}]),$$

- HI sendiri didefinisikan sebagai fungsi gabungan dari 3 consecutive operation: yaitu Batch Normalization (BN), dilanjutkan dengan Rectified Linear Unit (ReLU), dan sebuah 3x3 convolutional.
 - Namun, fungsi tersebut akan tidak bekerja jika ukuran fitur maps berubah. Namun demikian, bagian penting dari convolutional network adalah down-sampling layers yang dapat mengubah ukuran fitur maps.
 - Untuk memfasilitasi down-sampling tersebut, DenseNet membagi network menjadi beberapa connected densely dense blocks, seperti gambar berikut ini
-



- Layers di antara blocks disebut sebagai transiton layers. Layer inilah yang melakukan convolution dan pooling.
- Layer tersebut terdiri dari BN layer dan sebuah 1x1 convolutional layer, diikuti dengan 2x2 average pooling layer.

$$\mathbf{x}_\ell = H_\ell([\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{\ell-1}]),$$

- Jika setiap fungsi H_ℓ menghasilkan fitur map sebanyak k , maka layer ke- ℓ memiliki $k_0 + k(\ell-1)$ input fitur maps, dimana k_0 merupakan jumlah channel pada input layer.
 - Sebuah perbedaan penting antara DenseNet dengan network arsitektur lain adalah bahwa DenseNet dapat memiliki layer yang sangat sempit (misal $k=12$). Kita mengacu pada hyperparameter k sebagai growth rate network.
 - Setiap layer memiliki akses menuju semua fitur maps sebelumnya pada block masing-masing dan ke “collective knowledge” pada network.
 - Dalam state ini, setiap layer bisa menambahkan k fitur maps miliknya. Growth rate lah yang akan mengatur berapa banyak informasi baru dari setiap layer yang dikontribusikan ke global state.
 - Saat global state sudah terbentuk, maka bisa diakses dari manapun dalam network dan tidak perlu ada replikasi dari layer ke layer.
-

-
- Meskipun tiap layer hanya menghasilkan k output fitur maps, tapi biasanya juga memiliki banyak input. Telah tercatat juga bahwa convolution 1×1 dapat dikenal dengan bottleneck layer sebelum 3×3 convolution untuk mengurangi jumlah input fitur maps, dengan demikian akan meningkatkan efisiensi komputasi.
 - Design ini efektif secara khusus untuk DenseNet dan network DenseNet dengan Bottleneck layer, seperti BN-ReLU-Conv (1×1)-BN-ReLU-Conv (3×3) versi dari H1 sebagai DenseNet-B.
 - Untuk menambah kekompakan model, kita bisa mengurangi jumlah fitur maps pada transition layer. Jika sebuah dense block mengandung m fitur maps, maka kita bisa biarkan layer tersebut untuk generate $\lfloor \theta m \rfloor$ output fitur maps, dimana $0 < \theta \leq 1$ sebagai compression factor.
 - Saat $\theta = 1$, jumlah fitur maps pada transition layers tidak berubah. Kita menyebut DenseNet dengan $\theta < 1$ sebagai DenseNet-C, dan set $\theta = 0.5$.
 - Ketika bottleneck dan transition layers dengan $\theta < 1$ digunakan, maka model tersebut kita namakan DenseNet-BC.
-

-
- Di semua set data (kecuali ImageNet), DenseNet yang digunakan memiliki 3 dense block yang memiliki jumlah layer yang sama.
 - Sebelum memasuki dense block pertama, sebuah convolution dengan 16 (atau 2x growth rate DenseNet-BC) output channels diterapkan pada input images.
 - Untuk convolutional layers dengan kernel 3x3, tiap sisi input memiliki zero-padded by one pixel agar ukuran fitur map tetap fixed.
 - Convolution yang digunakan adalah 1x1, dilanjutkan dengan 2x2 average pooling sebagai transition layer antara 2 dense block yang berdekatan.
 - Sebuah global average pooling digunakan pada dense block terakhir, juga softmax classifier.
 - Ukuran fitur maps pada 3 dense block adalah 32x32, 16x16, dan 8x8.
 - Struktur basic DenseNet dikonfigurasi dengan $\{L=40, k=12\}$, $\{L=100, k=12\}$, dan $\{L=100, k=24\}$.
 - Untuk DenseNet-BC, network konfigurasinya $\{L=100, k=12\}$, $\{L=250, k=24\}$, dan $\{L=190, k=40\}$.
-

- Ada pula percobaan dengan ImageNet menggunakan struktur DenseNet-BC dengan dense block pada 224x224 input image. Convolutional layer awal terdiri dari 2k convolutions dengan ukuran 7x7 with stride 2.
- Jumlah fitur maps pada semua layer lain juga menggunakan setting k. Exact network configuration dalam ImageNet kurang lebih sbg berikut

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

Table 1: DenseNet architectures for ImageNet. The growth rate for all the networks is $k = 32$. Note that each “conv” layer shown in the table corresponds the sequence BN-ReLU-Conv.

Datasets

CIFAR

- Kedua dataset dari CIFAR terdiri dari colored natural image dengan 32x32 pixels.
 - CIFAR-10 (C10) terdiri dari gambar yang diambil dari 10 class, sedangkan CIFAR-100 (C100) terdiri dari gambar yang diambil dari 100 class.
 - Training dan test set nya berisi 50.000 dan 10.000 images masing-masing dan memberikan 5000 training images sebagai validation set.
 - Menggunakan sebuah skema standard data augmentation (mirroring/shifting) yang mayoritas digunakan untuk kedua datasets ini [11, 13, 17, 22, 28, 20, 32, 34].
 - Untuk preprocessing, normalisasi data menggunakan channel mean dan varians.
 - Untuk final run, menggunakan semua 50.000 training images dan report final test error di akhir training.
-

SVHN

- Merupakan singkatan dari Street View House Numbers, terdiri dari 32x32 colored digit images.
 - Ada 73,257 images untuk training set, 26,032 images untuk test set, dan 531,131 images untuk additional training.
 - Dengan mengikuti common practice [7, 13, 20, 22, 30], semua training data digunakan tanpa adanya data augmentation, dan sebuah validation set dengan 6000 images dipisahkan dari training set.
 - Model training yang dipilih adalah model dengan validation error terendah.
-

ImageNet

- ILSVRC 2012 classification dataset terdiri dari 1.2 juta images untuk training, 50.000 untuk validation, dari 1000 classes.
 - Untuk training images menggunakan skema data augmentation yang sama, seperti pada [8, 11, 12], dan apply single-crop atau 10-crop dengan ukuran 224x224 saat test time.
 - Setelah [11, 12, 13], jika ada classification error, maka direport pada validation set.
-

Contoh perbandingan error rates antara DenseNet dan ResNet

Model	top-1	top-5
DenseNet-121	25.02 / 23.61	7.71 / 6.66
DenseNet-169	23.80 / 22.08	6.85 / 5.92
DenseNet-201	22.58 / 21.46	6.34 / 5.54
DenseNet-264	22.15 / 20.80	6.12 / 5.29

Table 3: The top-1 and top-5 error rates on the ImageNet validation set, with single-crop / 10-crop testing.

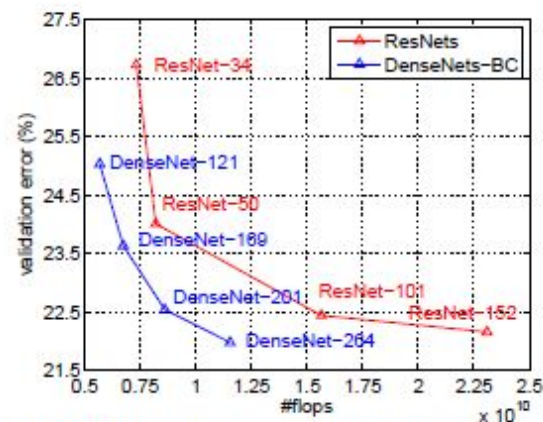
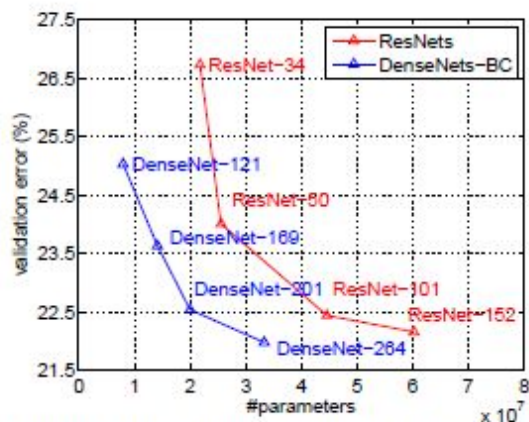


Figure 3: Comparison of the DenseNets and ResNets top-1 error rates (single-crop testing) on the ImageNet validation dataset as a function of learned parameters (*left*) and FLOPs during test-time (*right*).

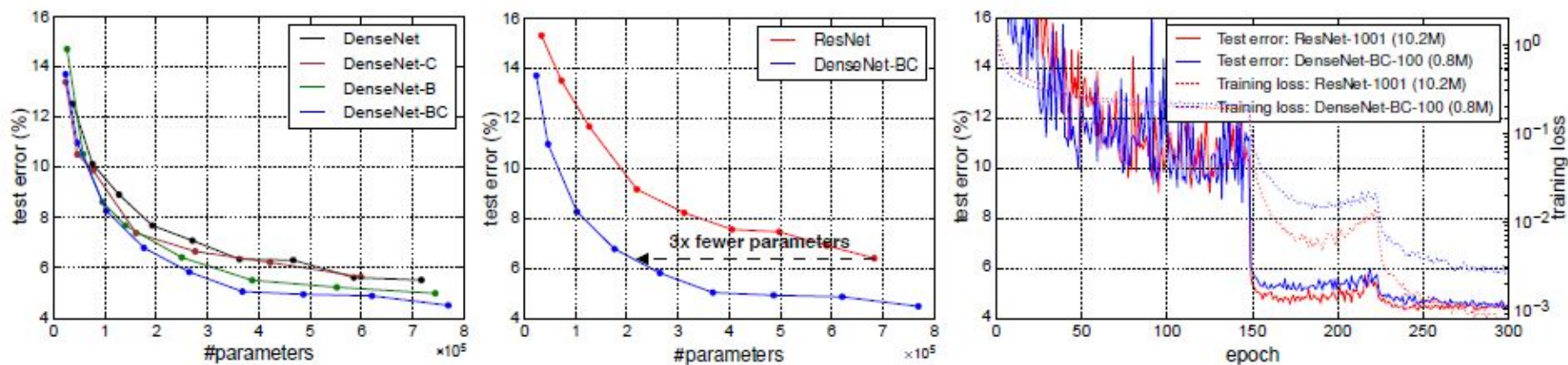


Figure 4: *Left:* Comparison of the parameter efficiency on C10+ between DenseNet variations. *Middle:* Comparison of the parameter efficiency between DenseNet-BC and (pre-activation) ResNets. DenseNet-BC requires about 1/3 of the parameters as ResNet to achieve comparable accuracy. *Right:* Training and testing curves of the 1001-layer pre-activation ResNet [12] with more than 10M parameters and a 100-layer DenseNet with only 0.8M parameters.

Terima Kasih
