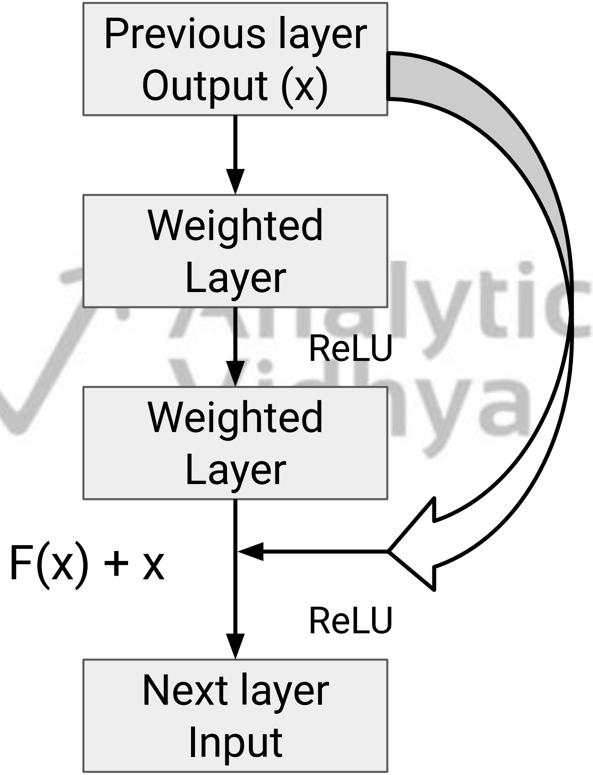
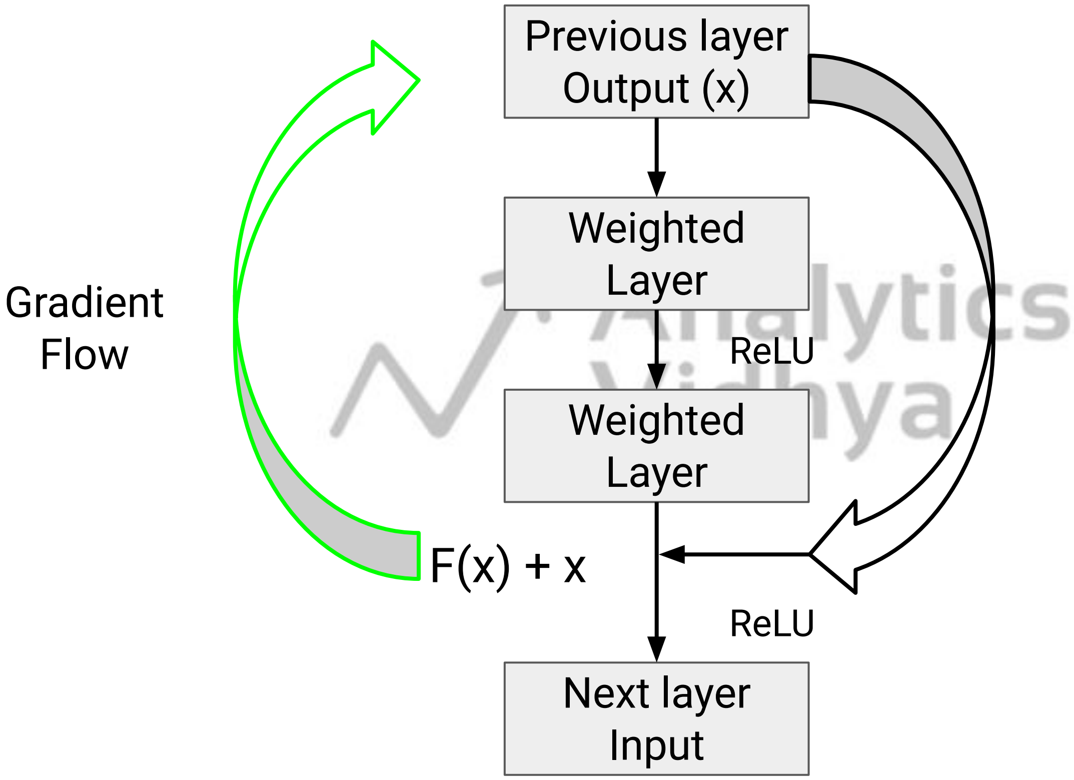
DenseNet

ResNet



Residual Block

ResNet



Residual Block

DenseNet

DenseNet

- Proposed in 2018

Densely Connected Convolutional Networks

Gao Huang*
Cornell University
gh349@cornell.edu

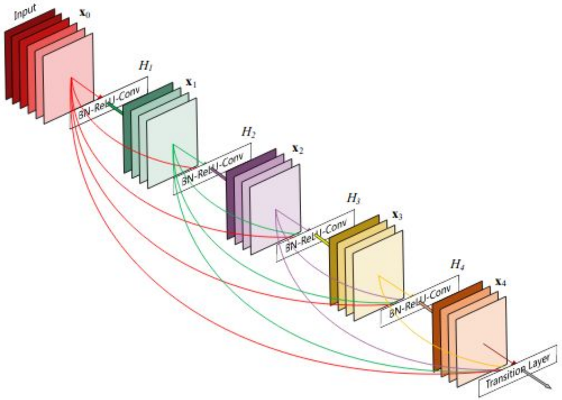
Zhuang Liu*
Tsinghua University
liuzhuang13@mails.tsinghua.edu.cn

Laurens van der Maaten
Facebook AI Research
lvdmaaten@fb.com

Kilian Q. Weinberger
Cornell University
kqw4@cornell.edu

Abstract

Recent work has shown that convolutional networks can be substantially deeper, more accurate, and efficient to train if they contain shorter connections between layers close to the input and those close to the output. In this paper, we embrace this observation and introduce the Dense Convolutional Network (DenseNet), which connects each layer to every other layer in a feed-forward fashion. Whereas traditional convolutional networks with L layers have L connections—one between each layer and its subsequent layer—our network has $\frac{L(L+1)}{2}$ *direct connections. For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs*



DenseNet

- Proposed in 2018
- Architectural details:
  - Mosty 3 X 3 convolutions are used

DenseNet

- Proposed in 2018
- Architectural details:
  - Mosty 3 X 3 convolutions are used
  - Batch normalization and ReLU used before conv layers in dense blocks

DenseNet

- Proposed in 2018
- Architectural details:
    - Mosty 3 X 3 convolutions are used
    - Batch normalization and ReLU used before conv layers in dense blocks
    - 1 X 1 convolution used to reduce the size of feature map

DenseNet

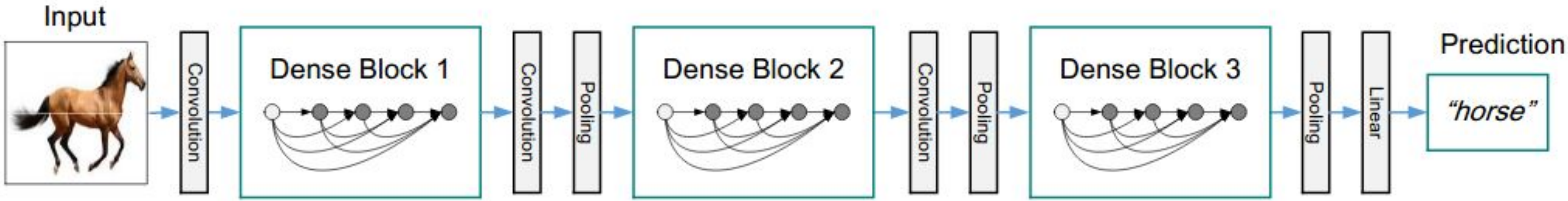
- Proposed in 2018
- Architectural details:
  - Mosty 3 X 3 convolutions are used
  - Batch normalization and ReLU used before conv layers in dense blocks
  - 1 X 1 convolution used to reduce the size of feature map
- Connects each layer to all the subsequent layer

DenseNet

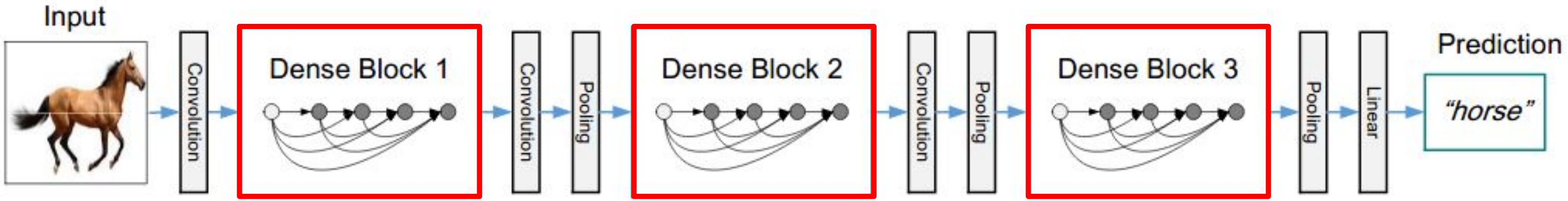
- Proposed in 2018
- Architectural details:
  - Mosty 3 X 3 convolutions are used
  - Batch normalization and ReLU used before conv layers in dense blocks
  - 1 X 1 convolution used to reduce the size of feature map
- Connects each layer to all the subsequent layer
- Do not add features using element-wise addition, instead it concatenates the features

Architecture: DenseNet

Architecture: DenseNet

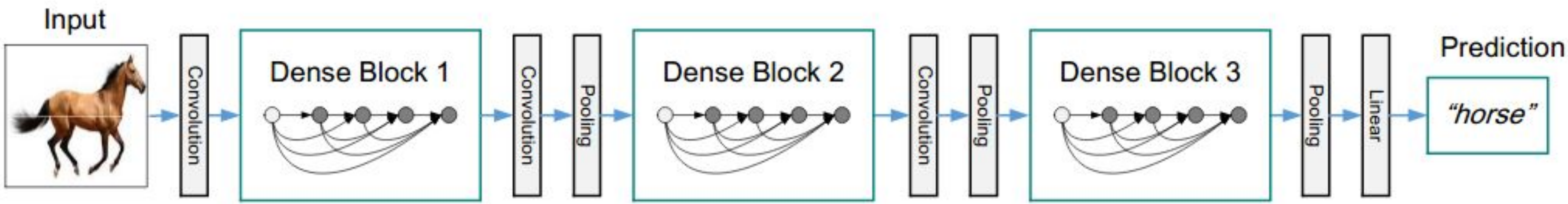


Architecture: DenseNet



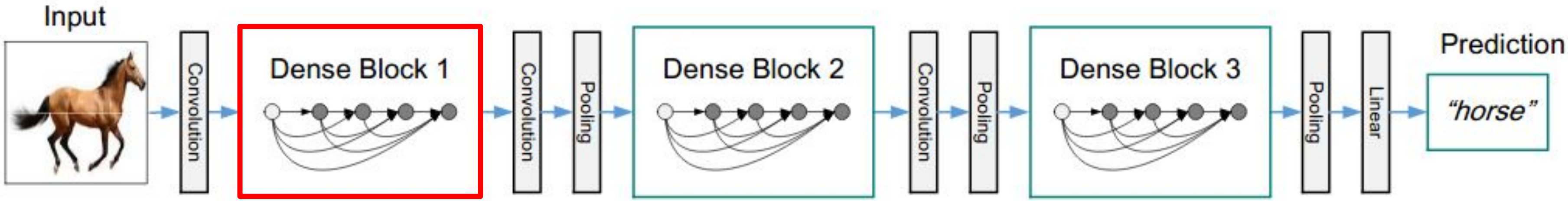
Analytics Vidhya

Architecture: DenseNet

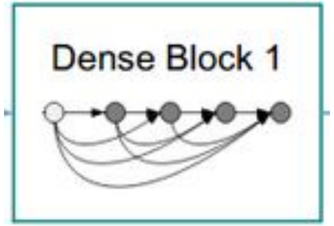


Analytics Vidhya

Architecture: DenseNet

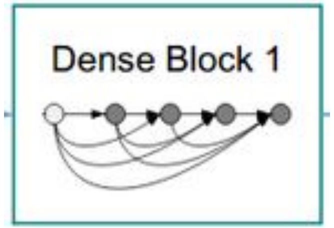


Understanding Dense Blocks



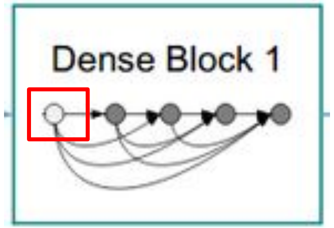
Understanding Dense Blocks

Input
(56, 56, 64)

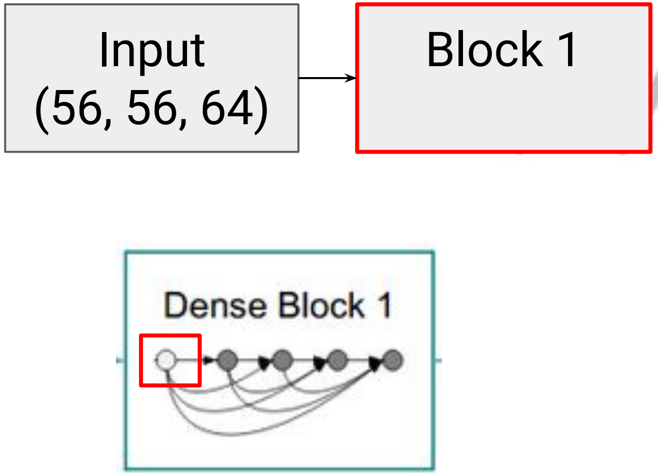


Understanding Dense Blocks

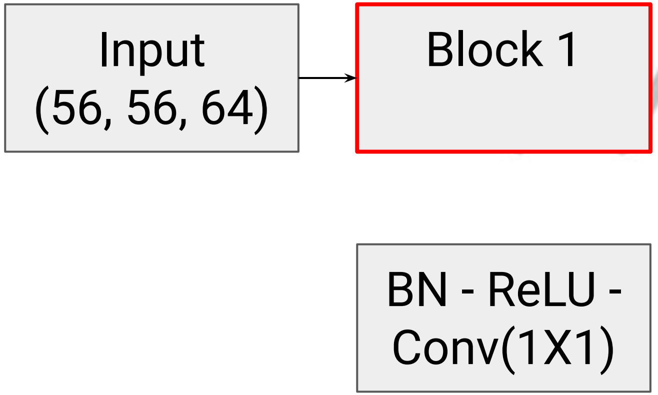
Input
(56, 56, 64) → Block 1



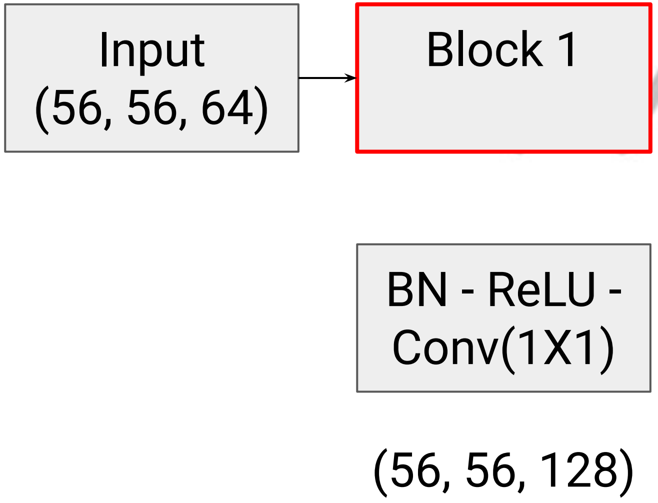
Understanding Dense Blocks



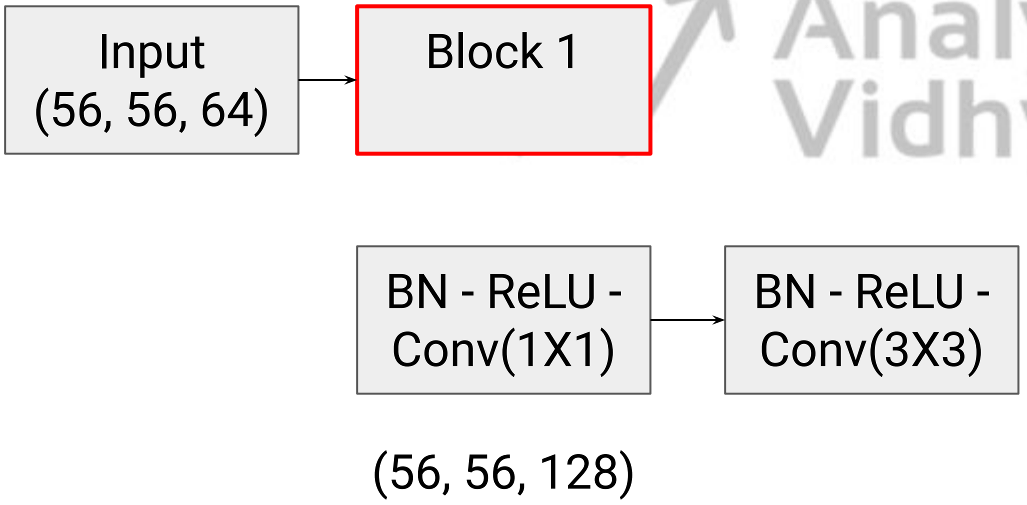
Understanding Dense Blocks



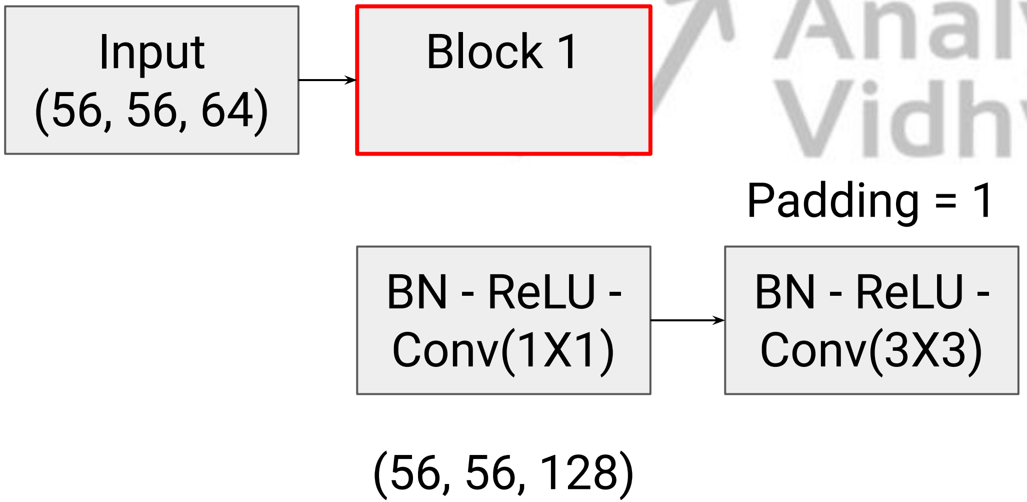
Understanding Dense Blocks



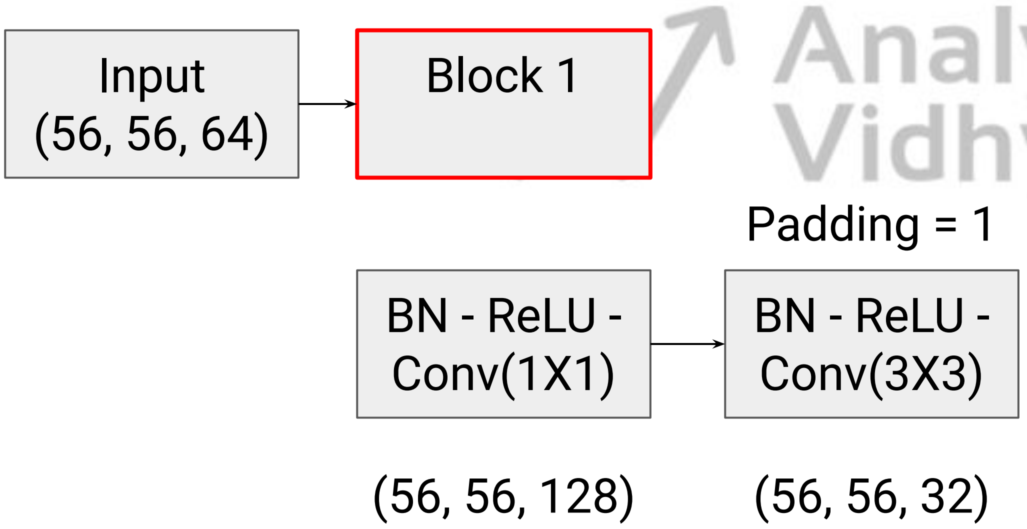
Understanding Dense Blocks



Understanding Dense Blocks



Understanding Dense Blocks

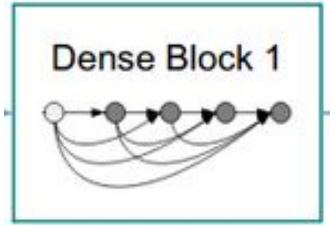


Understanding Dense Blocks

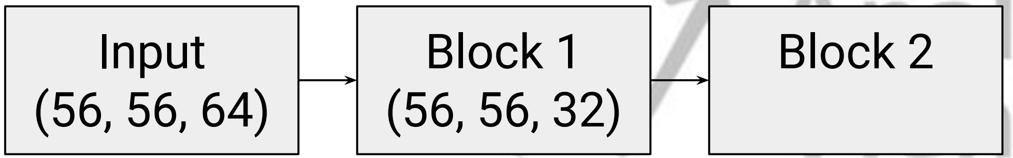
Input
(56, 56, 64)

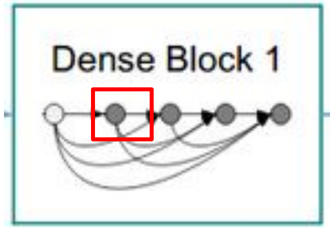
→

Block 1
(56, 56, 32)

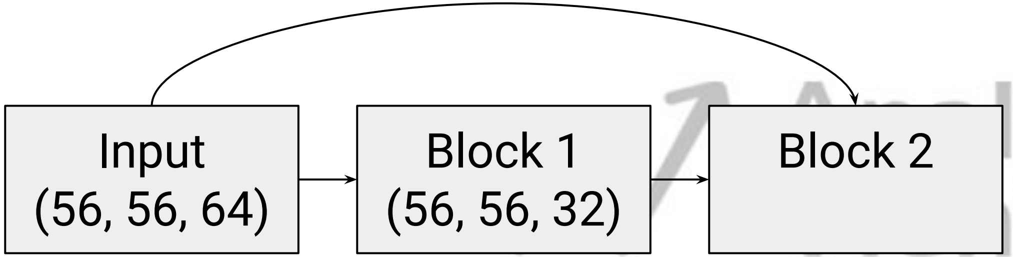


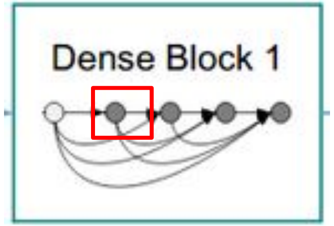
Understanding Dense Blocks



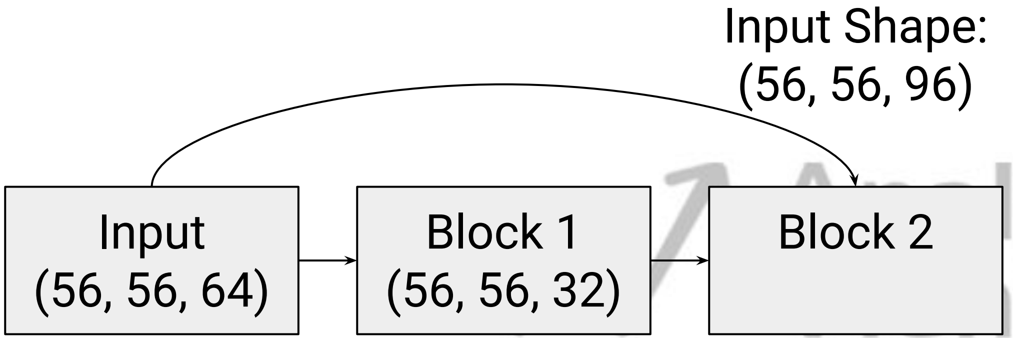


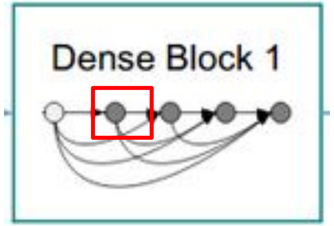
Understanding Dense Blocks



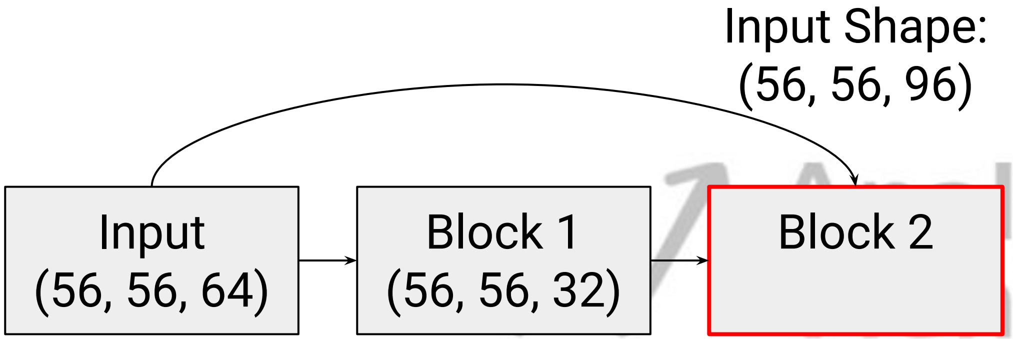


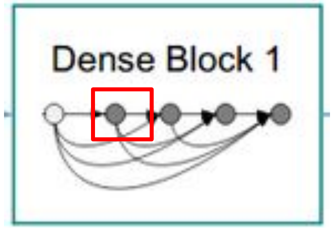
Understanding Dense Blocks



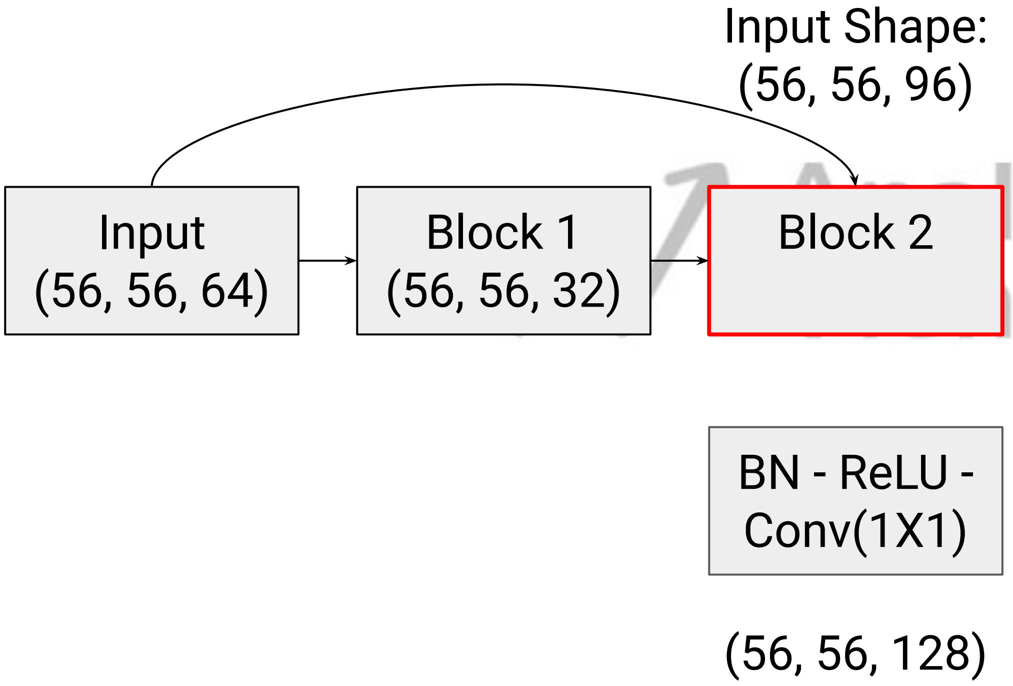


Understanding Dense Blocks

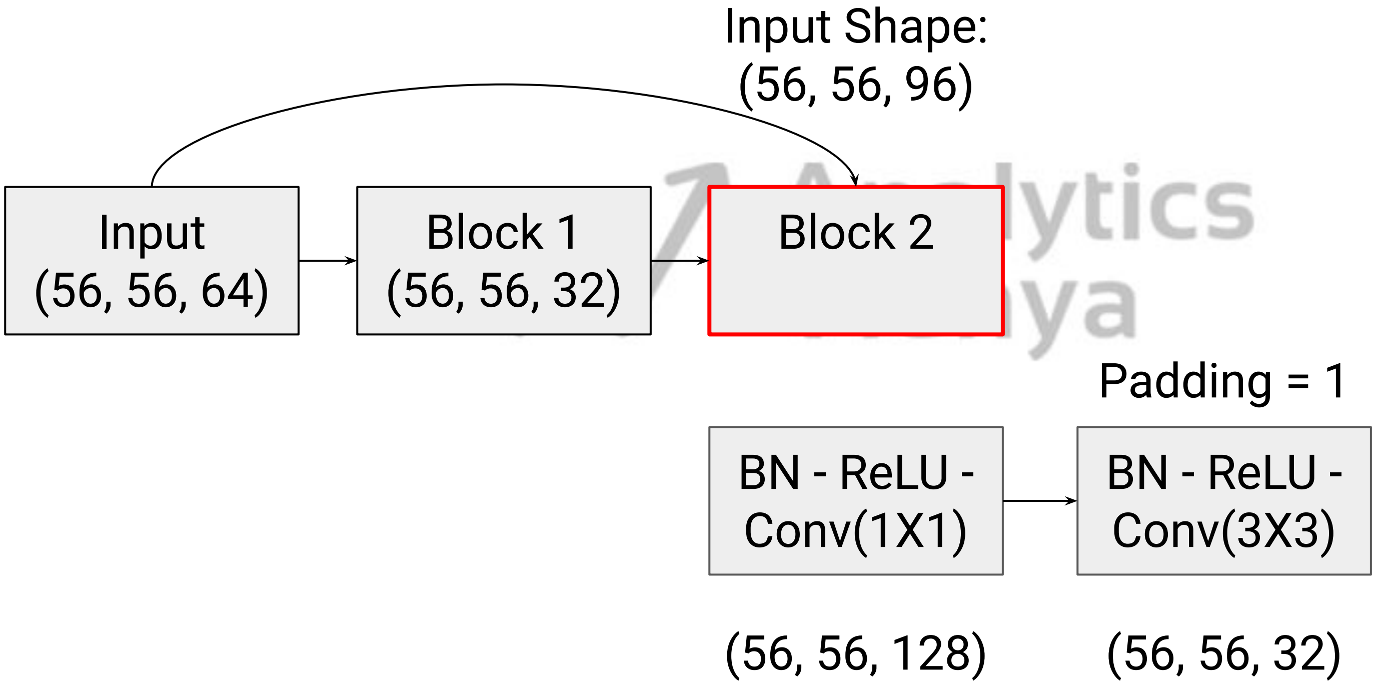




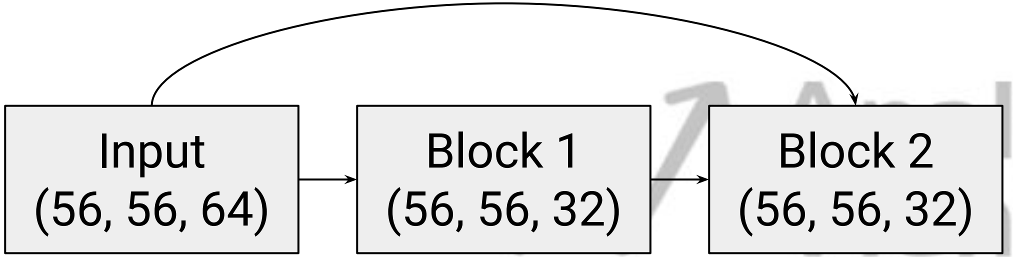
Understanding Dense Blocks

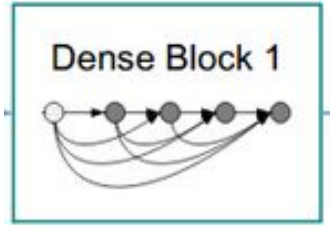


Understanding Dense Blocks

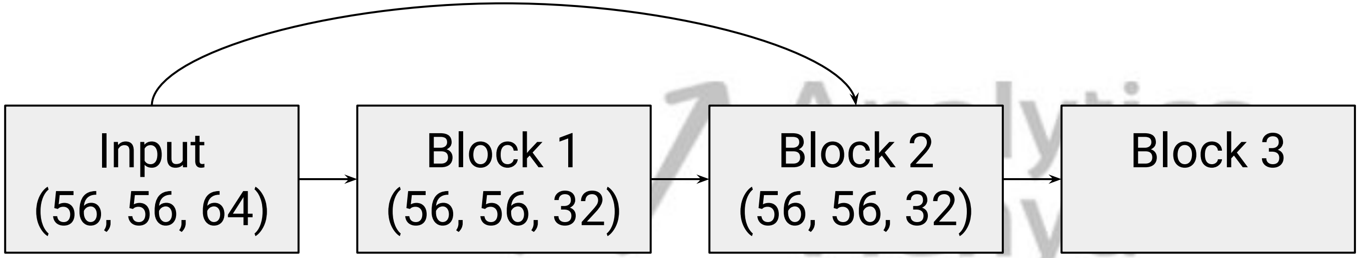


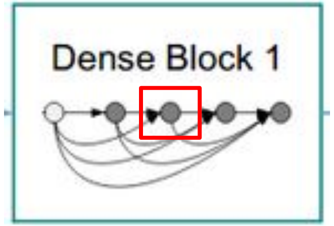
Understanding Dense Blocks



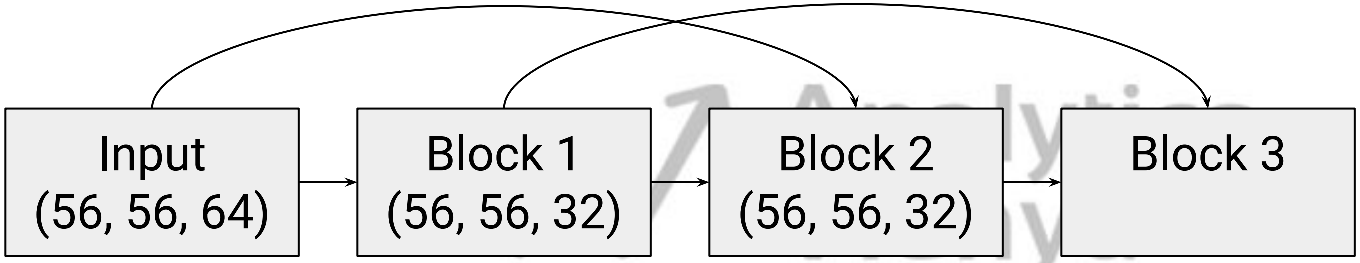


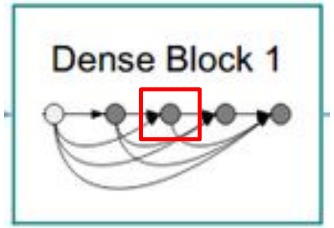
Understanding Dense Blocks



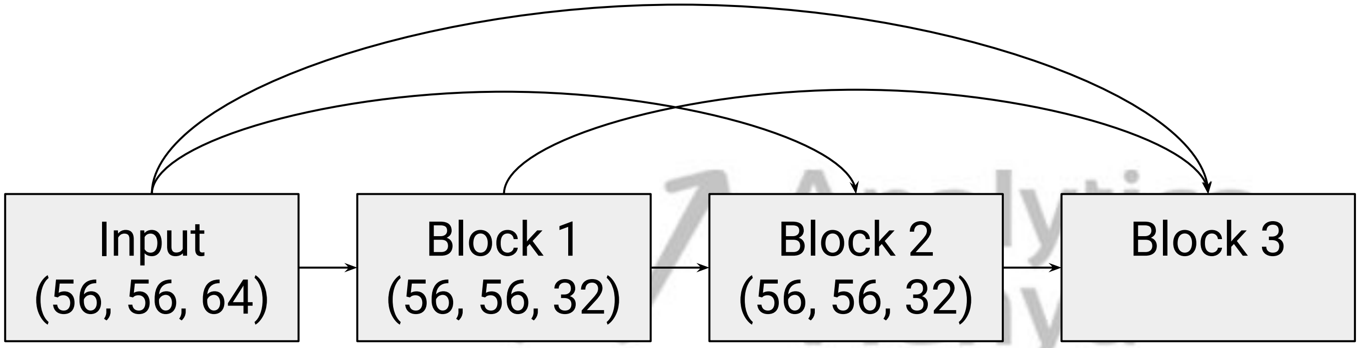


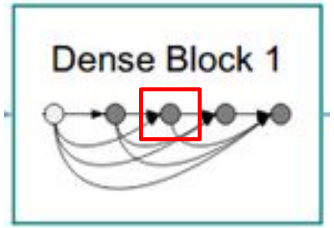
Understanding Dense Blocks



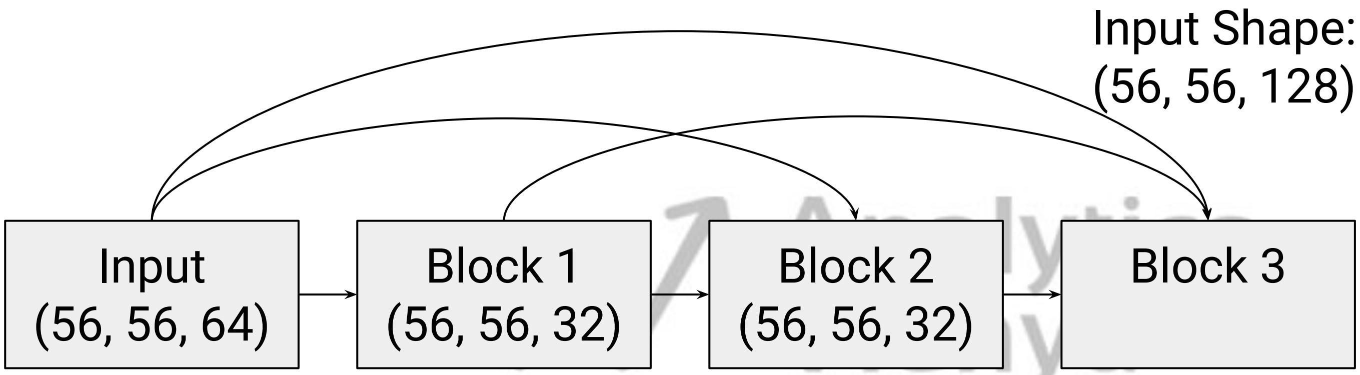


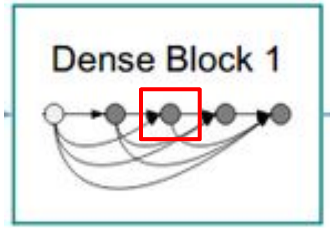
Understanding Dense Blocks



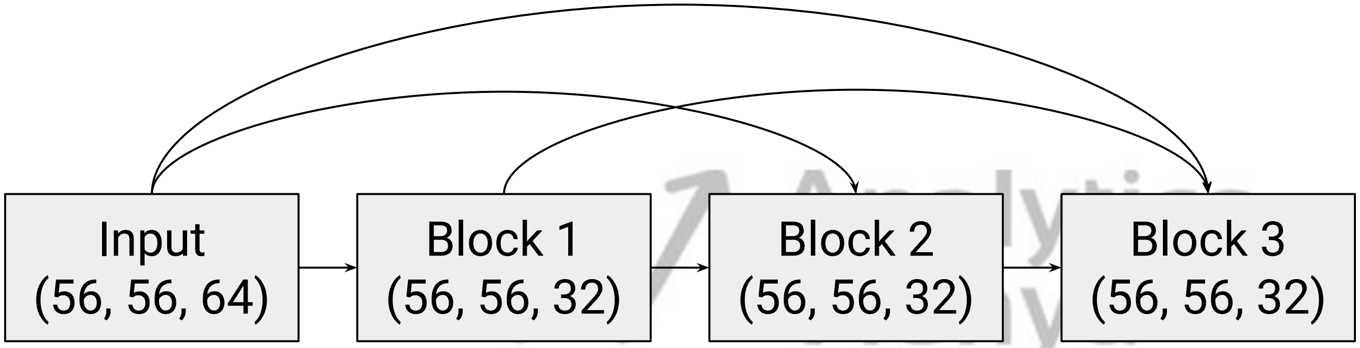


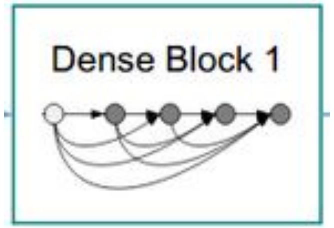
Understanding Dense Blocks



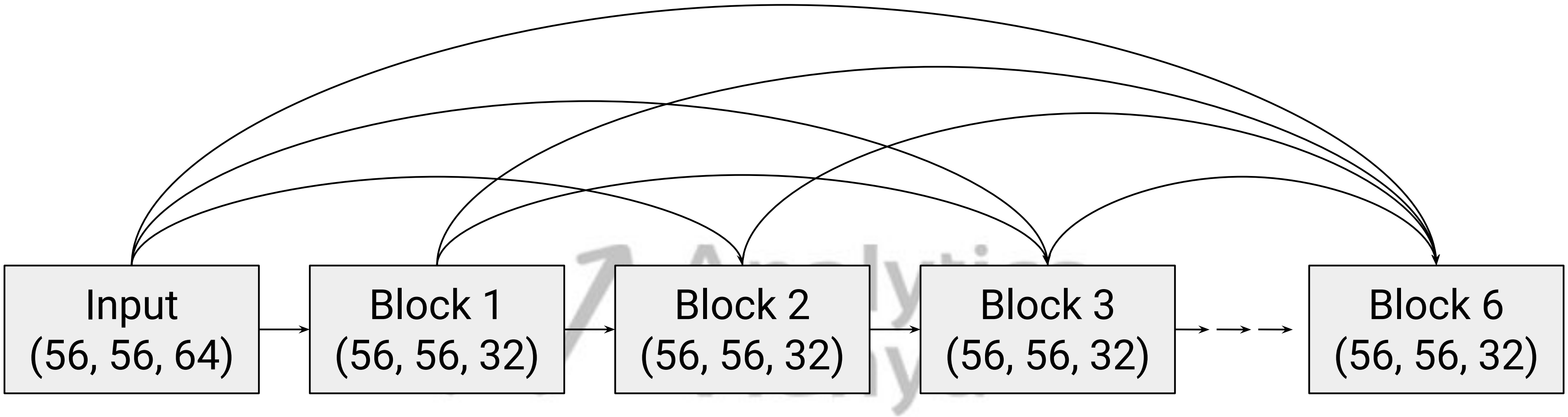


Understanding Dense Blocks

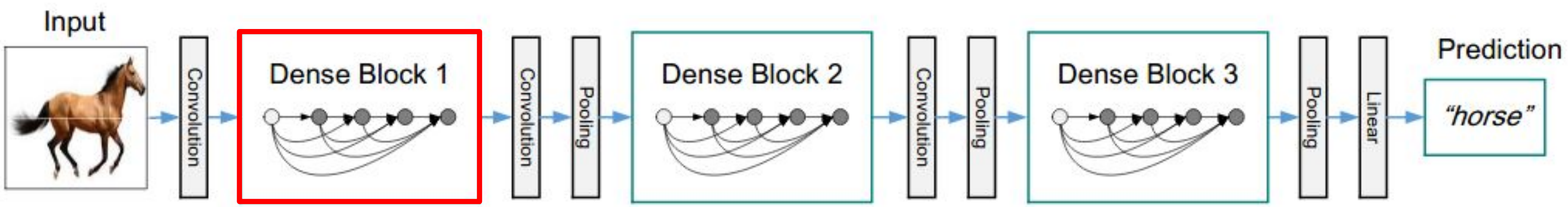


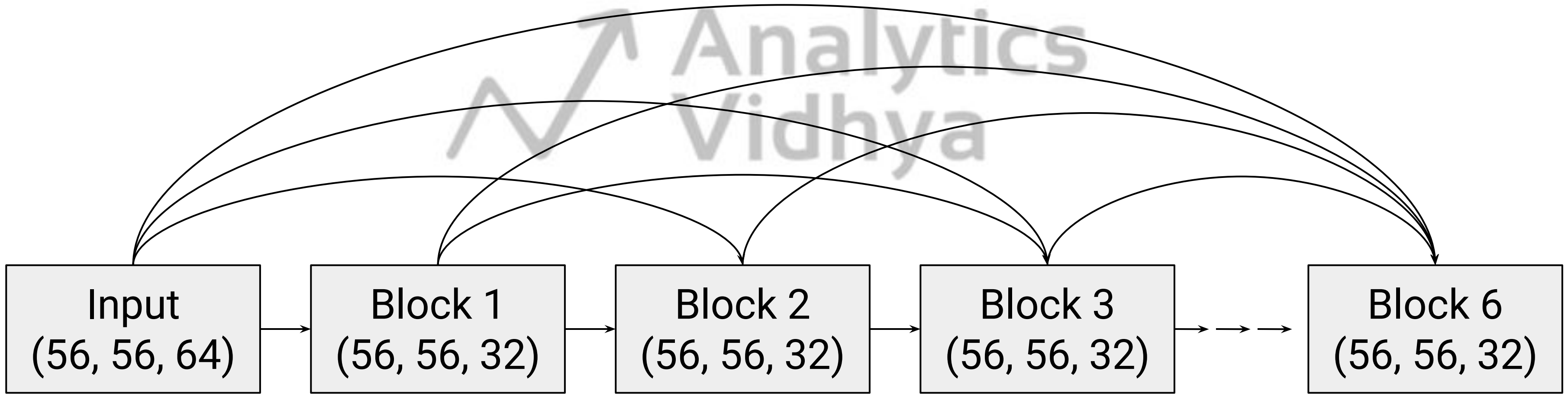


Understanding Dense Blocks

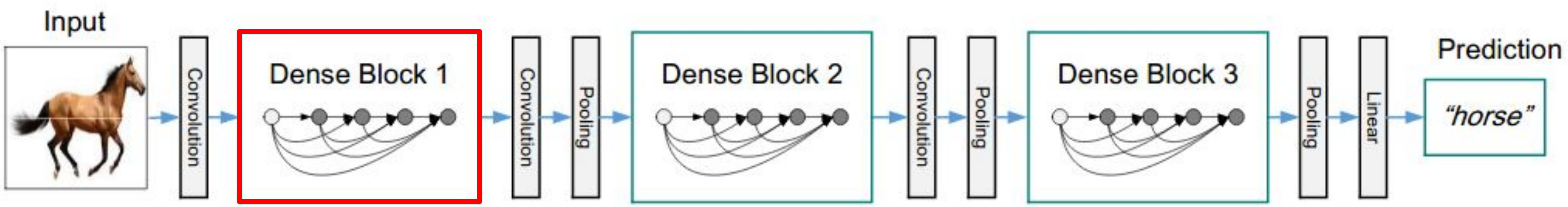


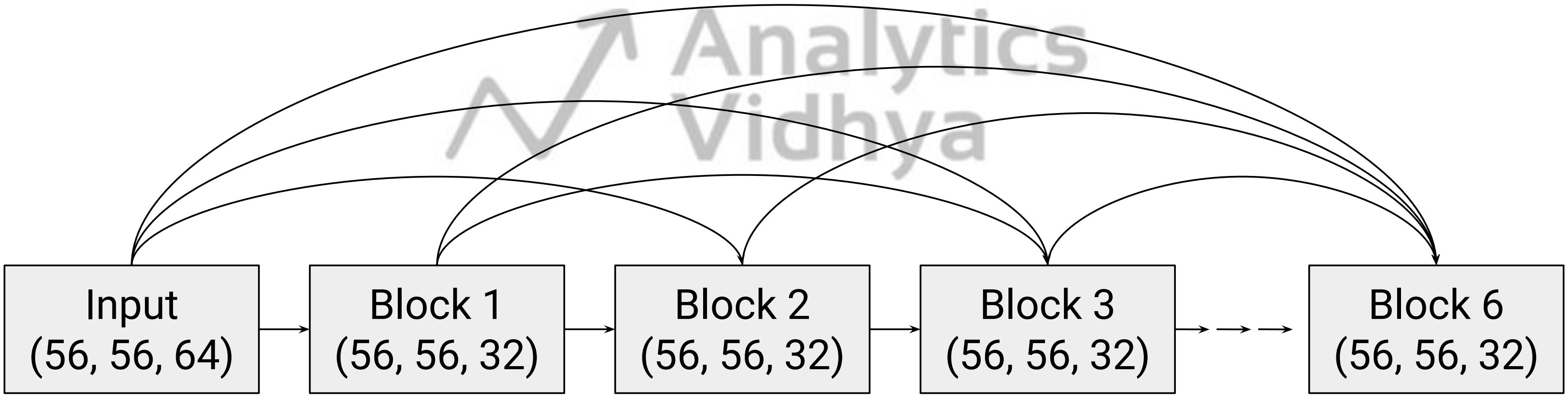
Architecture: DenseNet





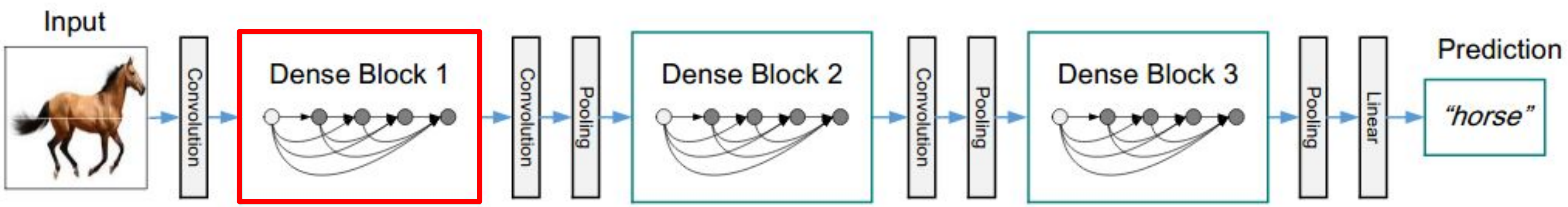
Architecture: DenseNet

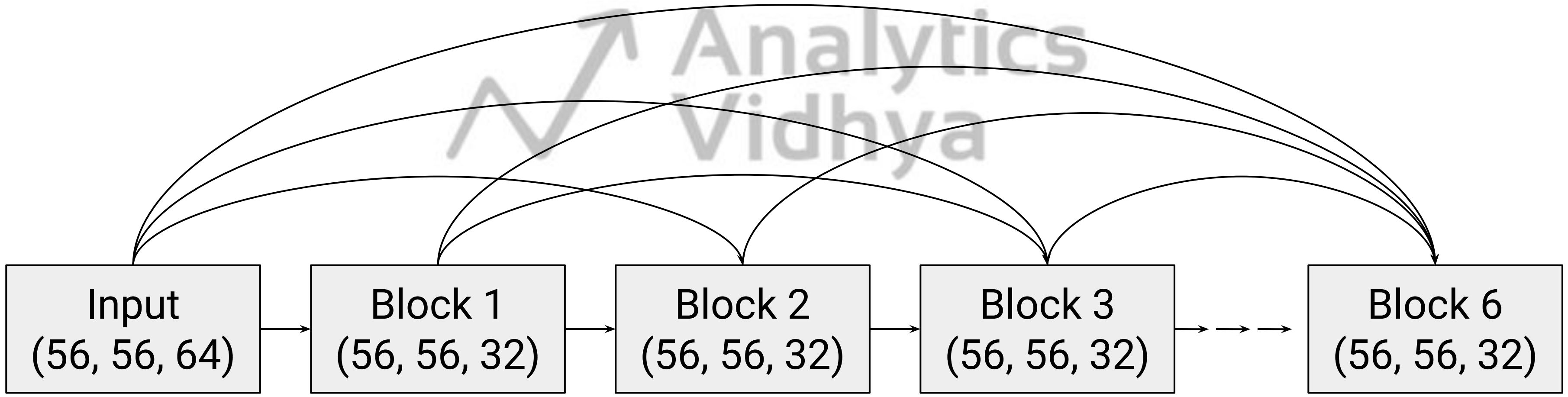




Output shape: (56, 56, 64 + 6*32)

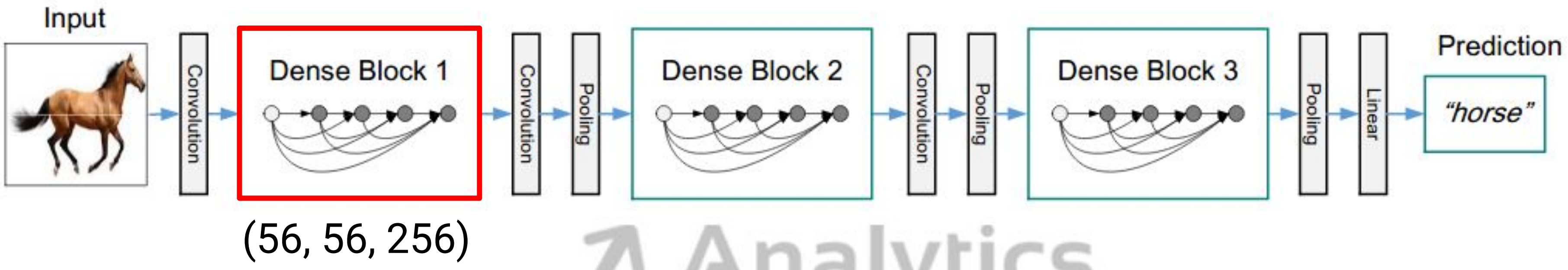
Architecture: DenseNet



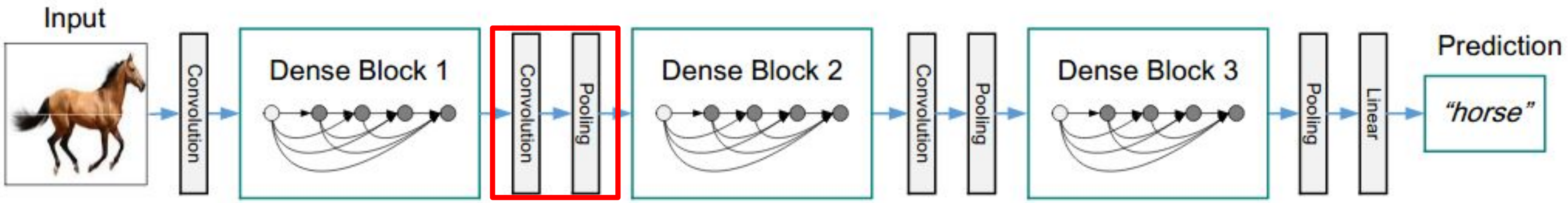


Output shape: (56, 56, 64 + 6*32) = (56, 56, 256)

Architecture: DenseNet

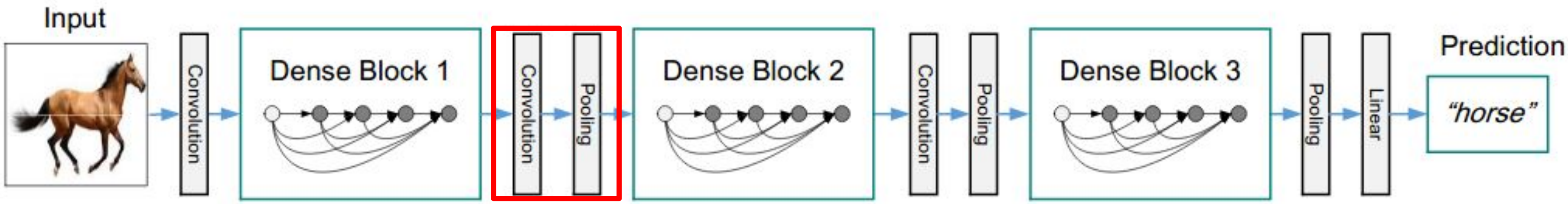


Architecture: DenseNet



(56, 56, 256)

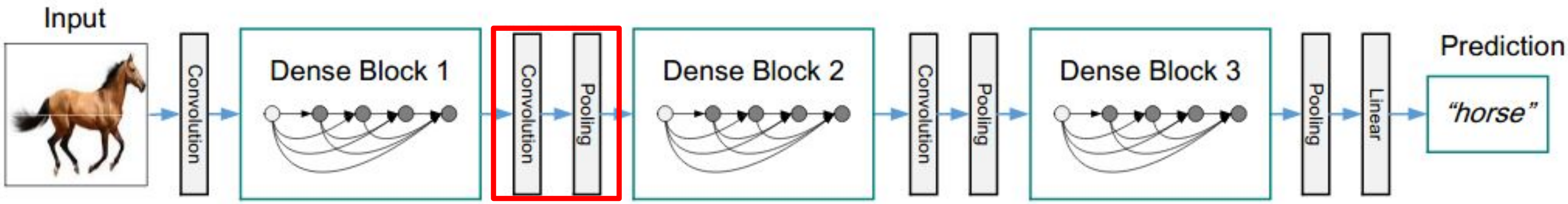
Architecture: DenseNet



(56, 56, 256)

Transition Layers

Architecture: DenseNet

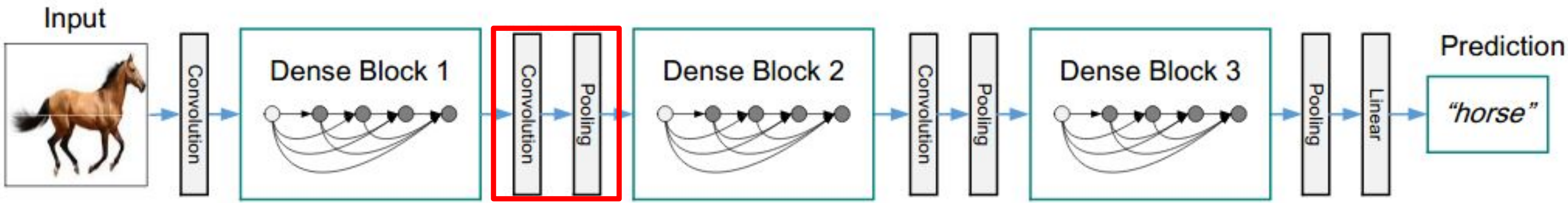


(56, 56, 256)

Transition Layers

Input
(56, 56, 256)

Architecture: DenseNet

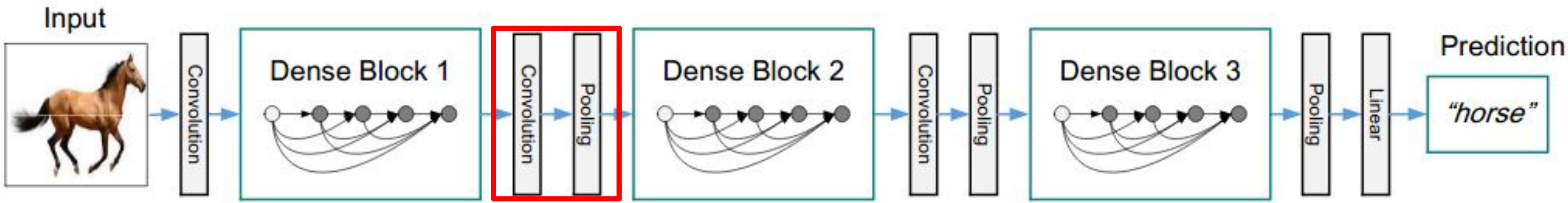


(56, 56, 256)

Transition Layers

Input (56, 56, 256) → BN - ReLU - Conv(1X1)

Architecture: DenseNet



(56, 56, 256)

Transition Layers

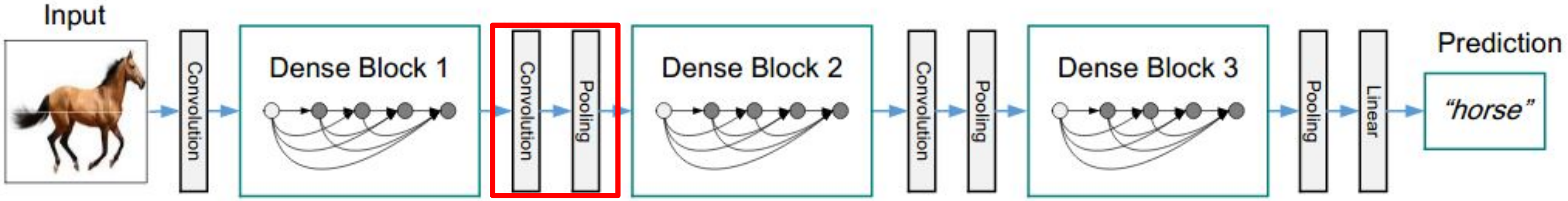
Input
(56, 56, 256)

→

BN - ReLU - Conv(1X1)

(56, 56, 128)

Architecture: DenseNet



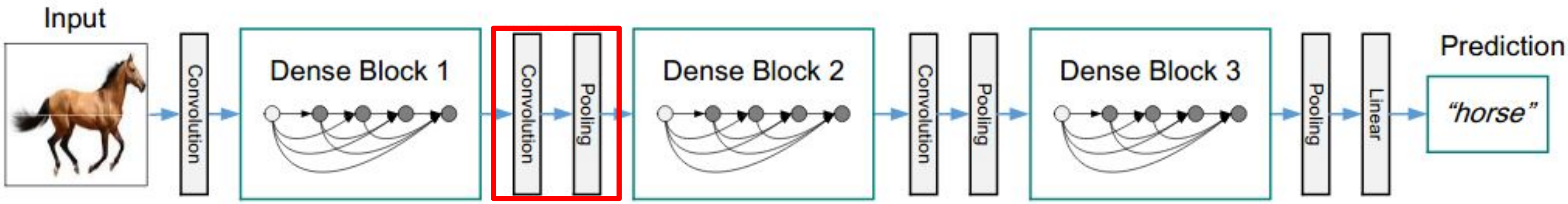
(56, 56, 256)

Transition Layers

Input (56, 56, 256) → BN - ReLU - Conv(1X1) → Average Pooling (2X2, stride = 2)

(56, 56, 128)

Architecture: DenseNet



(56, 56, 256)

Transition Layers

Input (56, 56, 256) → BN - ReLU - Conv(1X1) → Average Pooling (2X2, stride = 2)

(56, 56, 128)

(28, 28, 128)

Architecture: DenseNet

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			

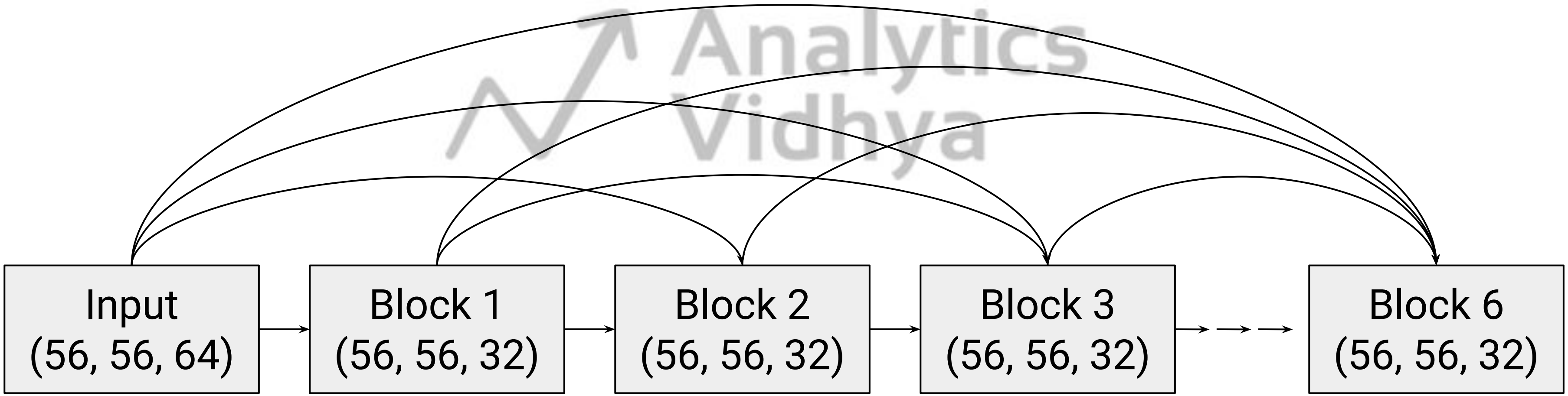
Architecture: DenseNet

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			

Advantages of Dense Blocks

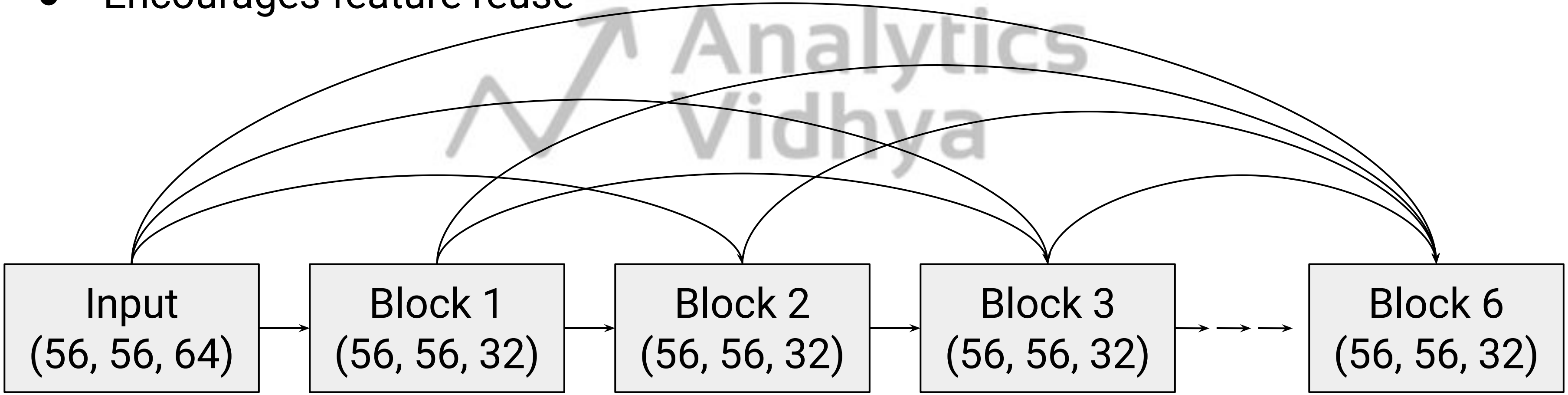
Advantages of Dense Blocks

- Resolves the issue of vanishing gradients



Advantages of Dense Blocks

- Resolves the issue of vanishing gradients
- Encourages feature reuse



Advantages of Dense Blocks

- Resolves the issue of vanishing gradients
- Encourages feature reuse
- Reduces number of parameters

Comparing versions of DenseNet

Comparing versions of DenseNet

Model	Top-5 Error
DenseNet-121	7.71%

Comparing versions of DenseNet

Model	Top-5 Error
DenseNet-121	7.71%
DenseNet-169	6.85%

Comparing versions of DenseNet

Model	Top-5 Error
DenseNet-121	7.71%
DenseNet-169	6.85%
DenseNet-201	6.34%

Comparing versions of DenseNet

Model	Top-5 Error
DenseNet-121	7.71%
DenseNet-169	6.85%
DenseNet-201	6.34%
DenseNet-264	6.12%

Thank You