南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

C/C++ Program Design

CS205

Week 14

Prof. Shiqi Yu (于仕琪)

<yusq@sustech.edu.cn>

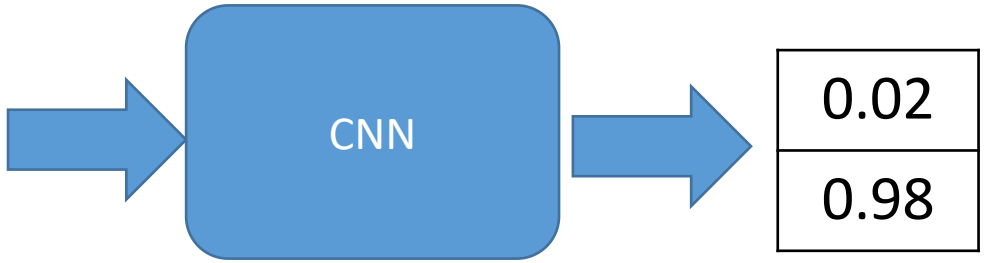
Prof. Feng Zheng(郑锋)

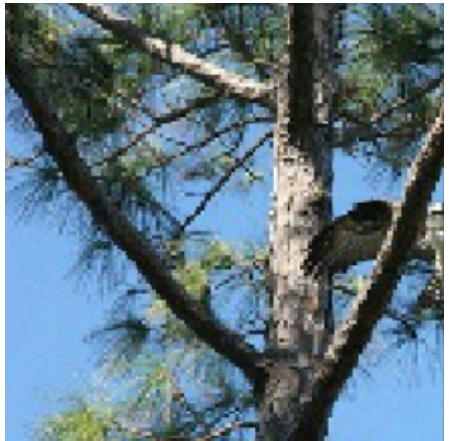
<zhengf@sustech.edu.cn>

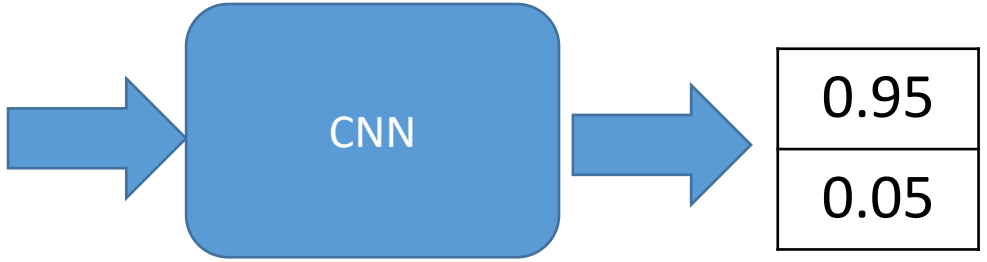
CNN for face classification

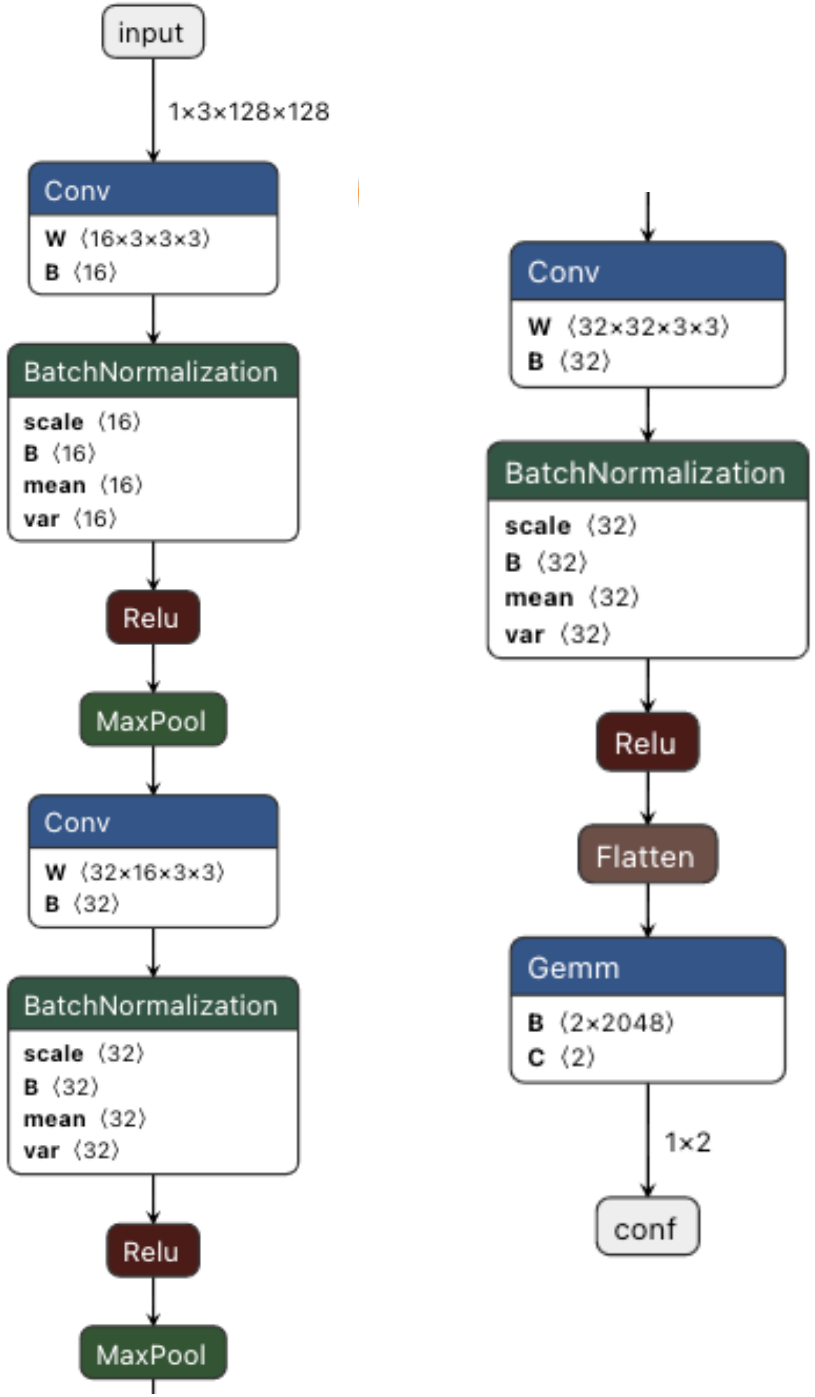
Pre-tained model

- https://github.com/ShiqiYu/SimpleCNNbyCPP









Model

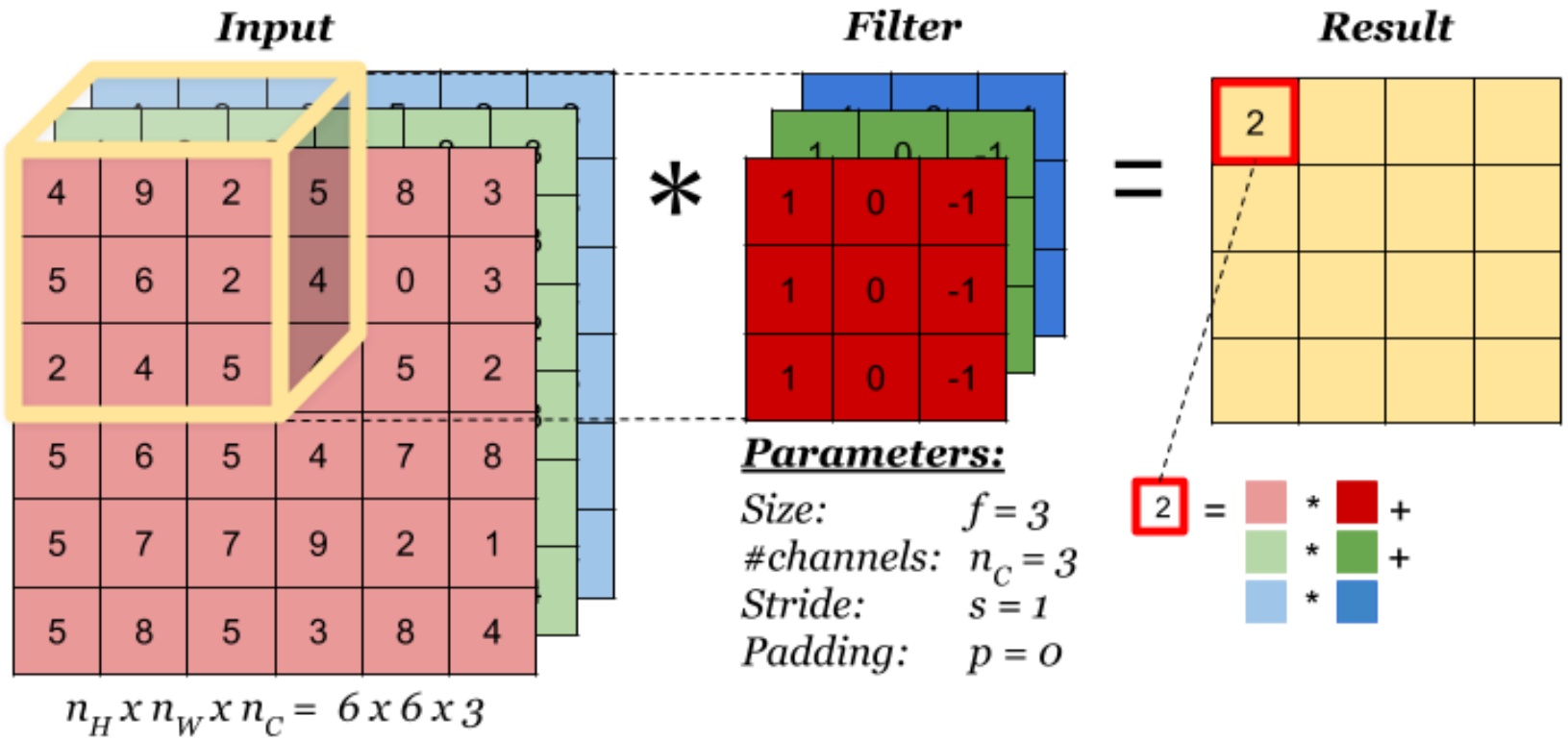
- 3 Convolutional layers (Conv+BN+ReLU)
- 2 MaxPool
- 1 Full connected layer

```
self.backbone = nn.Sequential(
    ConvBNReLU(3, 16, 3, 2, 1),     # downsampled by 2, 128 -> 64
    nn.MaxPool2d(2, 2),             # downsampled by 2, 64 -> 32
    ConvBNReLU(16, 32, 3, 1),       # keep
    nn.MaxPool2d(2, 2),             # downsampled by 2, 32 -> 16
    ConvBNReLU(32, 32, 3, 2, 1)     # downsampled by 2, 16 -> 8
)

self.classifier = nn.Sequential(
    nn.Linear(in_features=32*8*8,
              out_features=num_cls,
              bias=True)
)
```

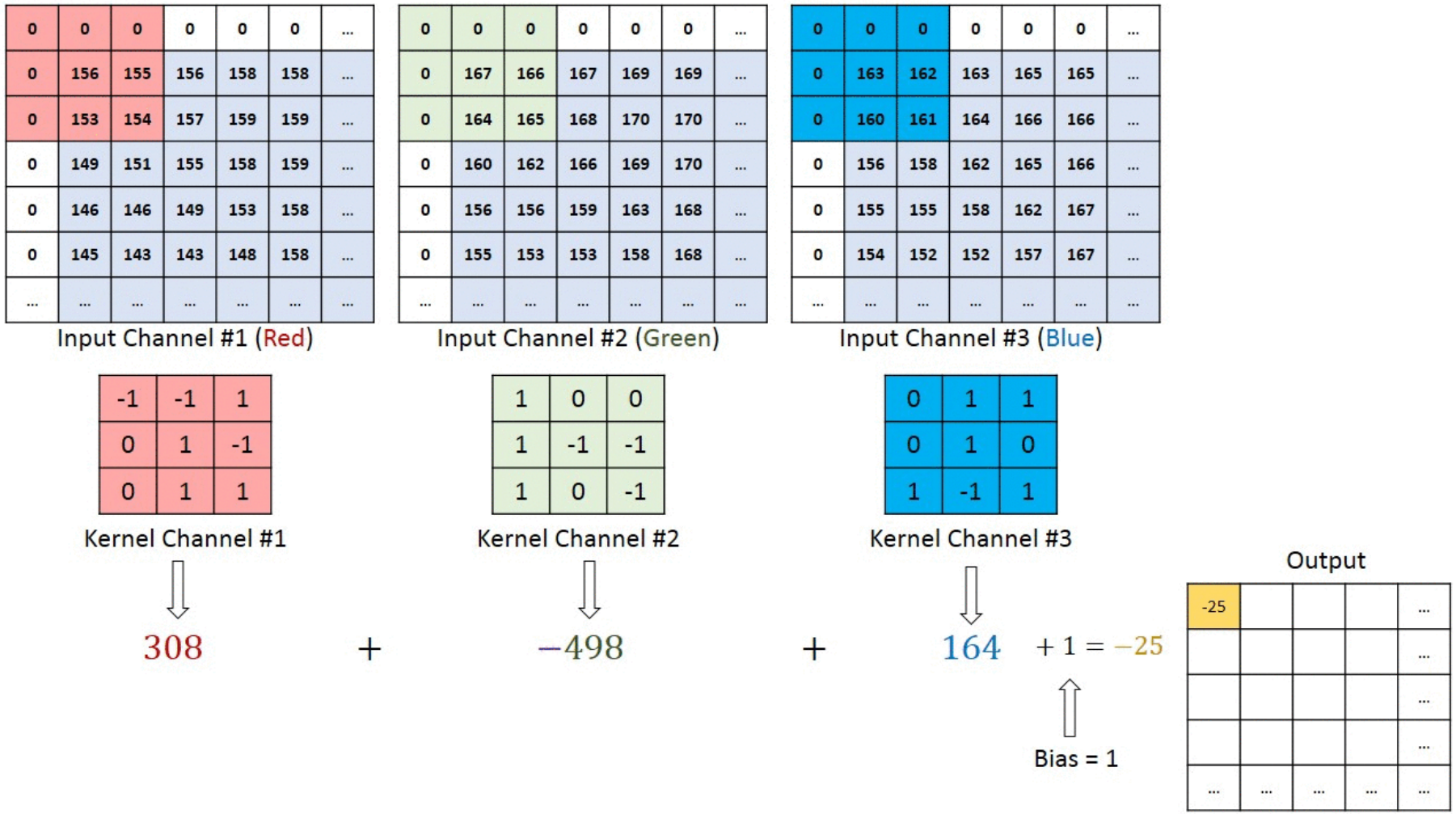
conv

- Multiple filters (kernels) can create multiple output channels



https://www.zybuluo.com/hongchenzimo/note/1086311

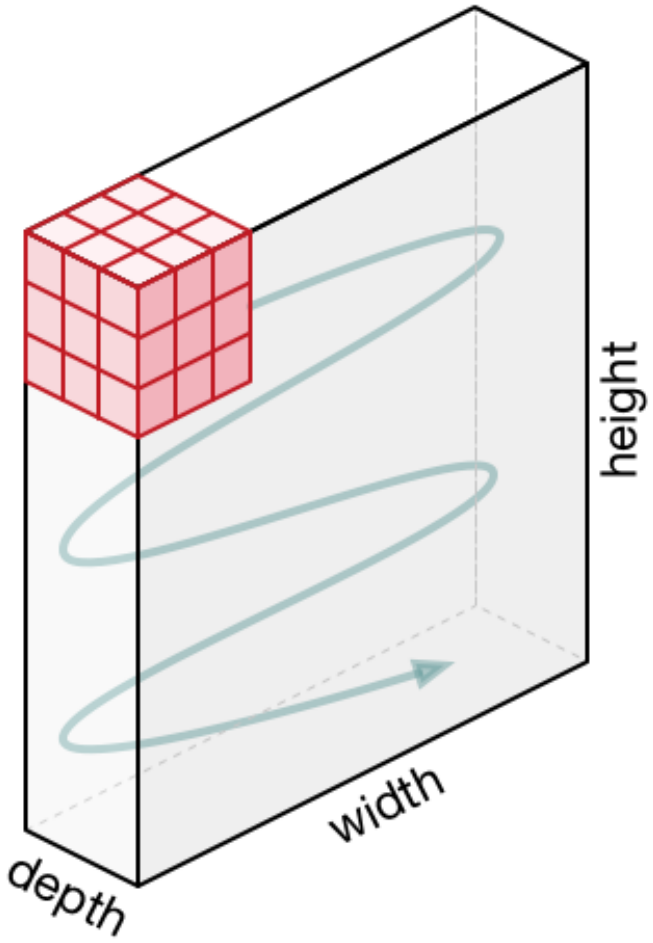
conv

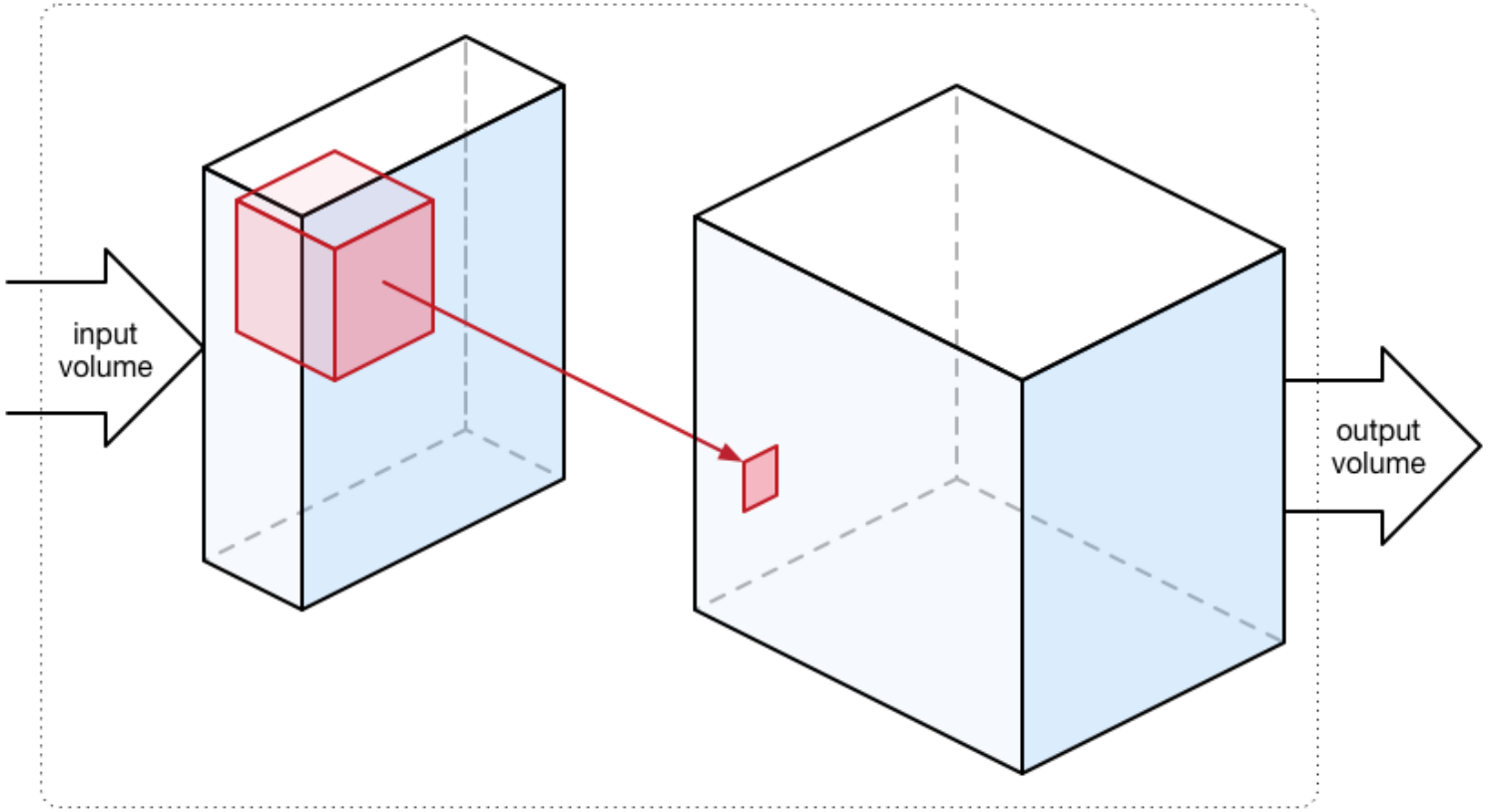


https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

conv

- A convolutional kernel create a channel in the output data.





https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Batch normalization

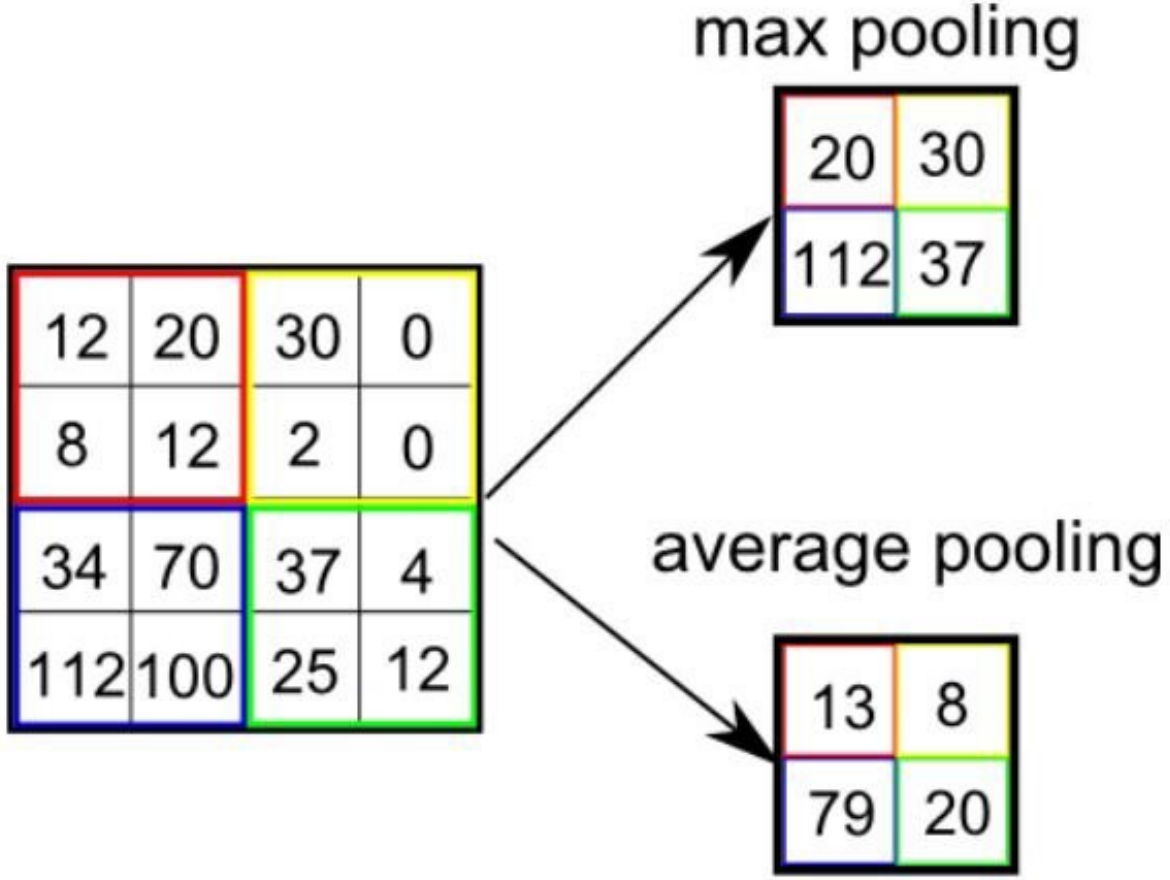
- The BN layers have been merged into conv layers

ReLU: Rectified Linear Unit

```
if (x < 0)
    x = 0;
```

max pooling

- For each channel

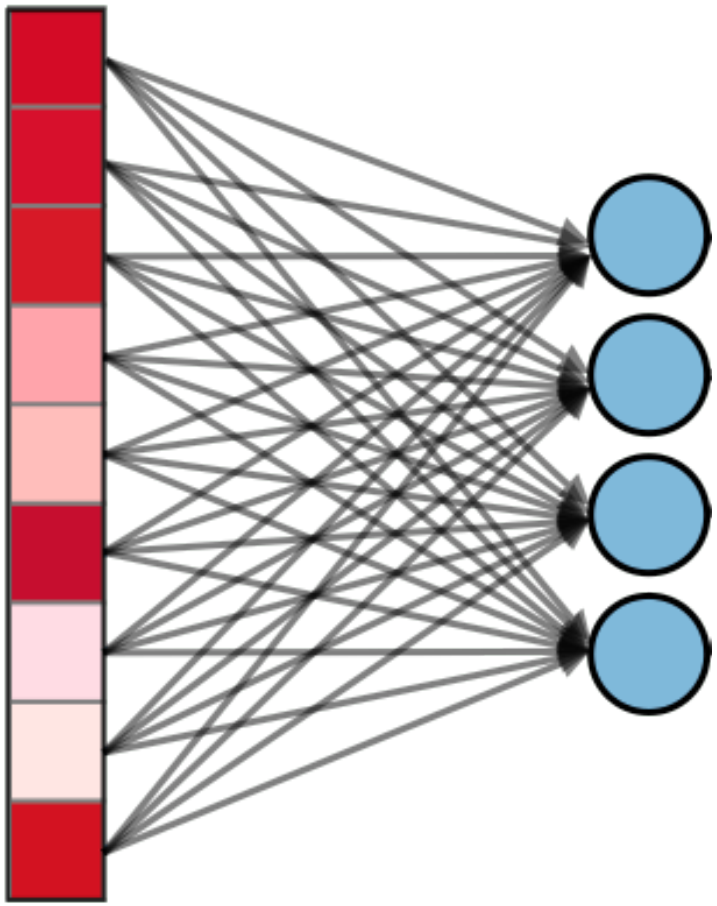


- If the input is C x H x W, the output will be C x H/2 x W/2

Flatten

- The data blob is flatted into a 2048-length vector from 32x8x8

FC: Fully-Connected Layer



- If the input is L and the output is N (2 in the model, 4 in left figure), the size of weights is NxL (2x2048 in the model)

$$output_{2x1} = weight_{2x2048} * input_{2048x1} + bias_{2x1}$$

Softmax

- To output the confidence vector (n=2 in the model)

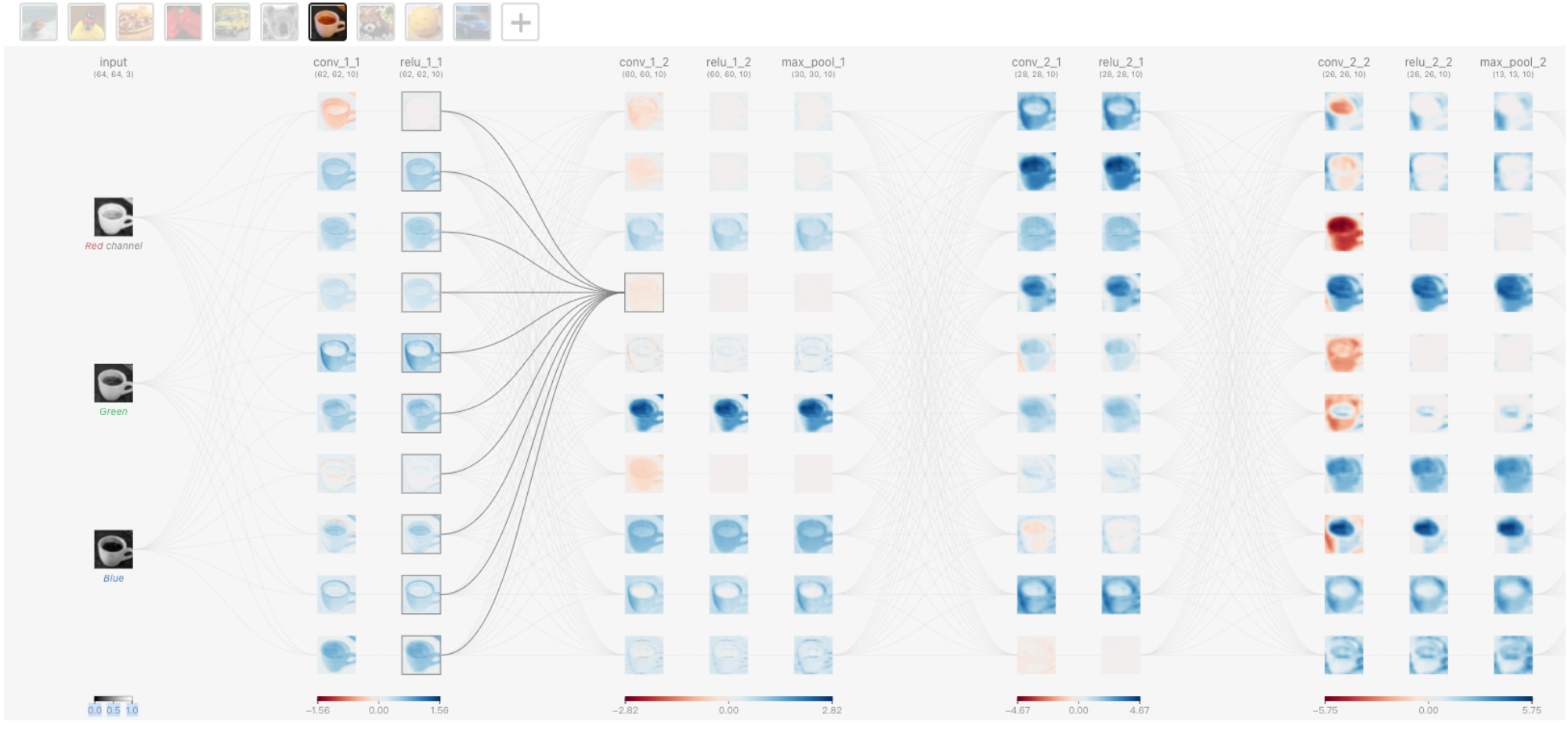
for $x \in \mathbb{R}^n$

$$p_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

$$p = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix}$$

CNN Explainer

- https://poloclub.github.io/cnn-explainer/



The slides are based on the book <Stephen Prata, C++ Primer Plus, 6th Edition, Addison-Wesley Professional, 2011>

Exceptions

Rudimentary Options

- An example: harmonic mean of two numbers

```
2.0 * x * y / (x + y)
```

- Calling abort(): program example error1.cpp
  - Send a message such as "abnormal program termination" to the standard error stream and terminate the program
  - Return an implementation-dependent value that indicates failure to the operating system
- Returning an error code: program error2.cpp
  - Return values to indicate a problem
  - We have used it in the previous examples

Throw-Catch Mechanism

- An exceptional circumstance arises while a program is running
- Exception mechanism provide a way to transfer control from one part of a program to another
  - Throwing an exception
    - ✓ throw keyword indicates the throwing of an exception
    - ✓ A throw statement, in essence, is a jump (other jump operators??)
  - Catching an exception with a handler
    - ✓ catch keyword indicates the catching of an exception
    - ✓ Followed by a type declaration that indicates the type of exception
  - Using a try block
    - ✓ A try block identifies a block of code for which particular exceptions will be activated
    - ✓ Followed by one or more catch blocks
- See program example error3.cpp

Throw-Catch Mechanism

- Can we use more complex types? YES!
- Using objects as exceptions
  - Advantage: use different exception types to distinguish among different functions and situations that produce exceptions
  - An object can carry information with it, and you can use this information to help identify the conditions that caused the exception to be thrown
  - A catch block could use that information to decide which course of action to pursue
- See program example error4.cpp
  - Geometric and harmonic means

More Exception Features of Throw-Catch Mechanism

- Differences to the normal function
  - One difference
    - ✓ A return statement: transfer execution to the calling function
    - ✓ A throw: transfer execution to the first function having a try-catc
  - Second difference
    - ✓ The compiler always creates a copy when throwing an exception
- The exception class
  - Define an exception class that C++ uses as a base class
  - One virtual member function is named what(), and it returns a string

```
class problem {...};
...
void super() throw (problem)
{
    ...
    if (oh_no)
    {
        problem oops;    // construct object
        throw oops;      // throw it
    ...
}
...
try {
    super();
}
catch(problem & p)
{
// statements
}
```

```
#include <exception>
class bad_hmean : public std::exception
{
public:
    const char * what() { return "bad arguments to hmean()"; }
...
};
```

More Exception Features

- The stdexcept exception classes
  - The stdexcept header file defines several more exception classes
  - logic_error and runtime_error classes
    - ✓ logic_error family: domain_error, invalid_argument, length_error, out_of_bounds
    - ✓ runtime_error family: range_error, overflow_error, underflow_error

```
namespace std {
    class logic_error;
    class domain_error;
    class invalid_argument;
    class length_error;
    class out_of_range;
    class runtime_error;
    class range_error;
    class overflow_error;
    class underflow_error;
}
```

Class std::logic_error

```
namespace std {
  class logic_error : public exception {
  public:
    explicit logic_error(const string& what_arg);
    explicit logic_error(const char* what_arg);
  };
}
```

Class std::domain_error

```
namespace std {
  class domain_error : public logic_error {
  public:
    explicit domain_error(const string& what_arg);
    explicit domain_error(const char* what_arg);
  };
}
```

More Exception Features

- The bad_alloc exception and new
  - Have new throw a bad_alloc exception
  - new returned a null pointer when it couldn't allocate the memory
- See program example newexcp.cpp