conv_layer_utils.py

```python
from nndl.layers import *
from cs231n.fast_layers import *

"""
This code was originally written for CS 231n at Stanford University
(cs231n.stanford.edu).  It has been modified in various areas for use in the
ECE 239AS class at UCLA.  This includes the descriptions of what code to
implement as well as some slight potential changes in variable names to be
consistent with class nomenclature.  We thank Justin Johnson & Serena Yeung for
permission to use this code.  To see the original version, please visit
cs231n.stanford.edu.
"""


def conv_relu_forward(x, w, b, conv_param):
  """
  A convenience layer that performs a convolution followed by a ReLU.

  Inputs:
  - x: Input to the convolutional layer
  - w, b, conv_param: Weights and parameters for the convolutional layer

  Returns a tuple of:
  - out: Output from the ReLU
  - cache: Object to give to the backward pass
  """
  a, conv_cache = conv_forward_fast(x, w, b, conv_param)
  out, relu_cache = relu_forward(a)
  cache = (conv_cache, relu_cache)
  return out, cache


def conv_relu_backward(dout, cache):
  """
  Backward pass for the conv-relu convenience layer.
  """
  conv_cache, relu_cache = cache
  da = relu_backward(dout, relu_cache)
  dx, dw, db = conv_backward_fast(da, conv_cache)
  return dx, dw, db


def conv_relu_pool_forward(x, w, b, conv_param, pool_param):
  """
  Convenience layer that performs a convolution, a ReLU, and a pool.

  Inputs:
  - x: Input to the convolutional layer
  - w, b, conv_param: Weights and parameters for the convolutional layer
  - pool_param: Parameters for the pooling layer

  Returns a tuple of:
  - out: Output from the pooling layer
  - cache: Object to give to the backward pass
  """
  a, conv_cache = conv_forward_fast(x, w, b, conv_param)
  s, relu_cache = relu_forward(a)
  out, pool_cache = max_pool_forward_fast(s, pool_param)
  cache = (conv_cache, relu_cache, pool_cache)
  return out, cache


```

```
63  def conv_relu_pool_backward(dout, cache):
64    """
65    Backward pass for the conv-relu-pool convenience layer
66    """
67    conv_cache, relu_cache, pool_cache = cache
68    ds = max_pool_backward_fast(dout, pool_cache)
69    da = relu_backward(ds, relu_cache)
70    dx, dw, db = conv_backward_fast(da, conv_cache)
71    return dx, dw, db
```