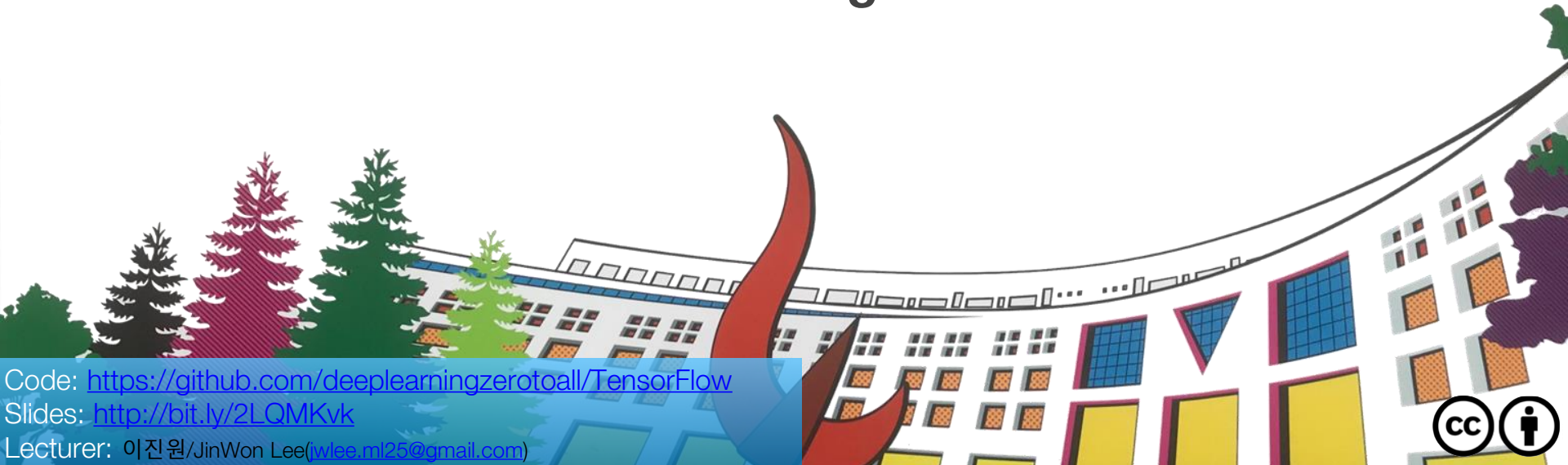
ML/DL for Everyone Season2

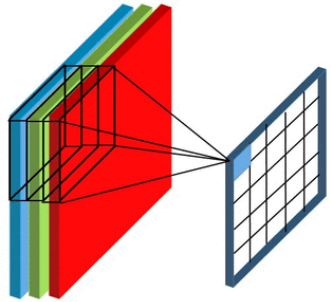
with TensorFlow

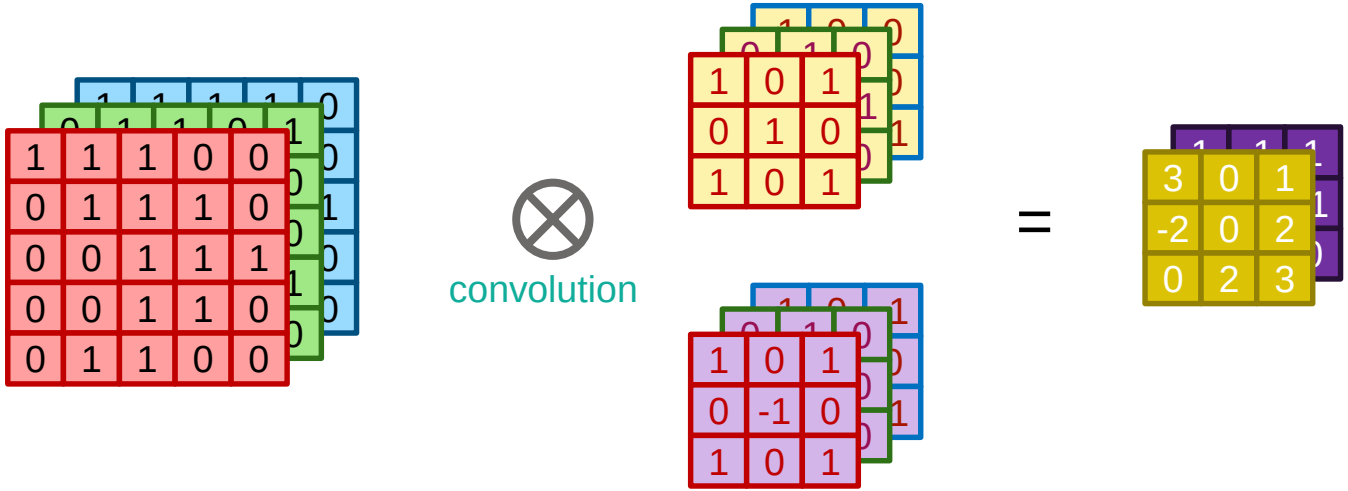
Lab 11-0 CNN Basics
Pooling



Code: https://github.com/deeplearningzerotoall/TensorFlow
Slides: http://bit.ly/2LQMKvk
Lecturer: 이진원/JinWon Lee(jwlee.ml25@gmail.com)

2D Convolution Layer – Multi Channel, Many Filters

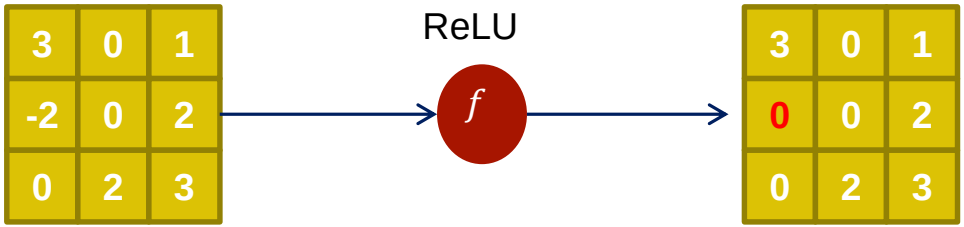


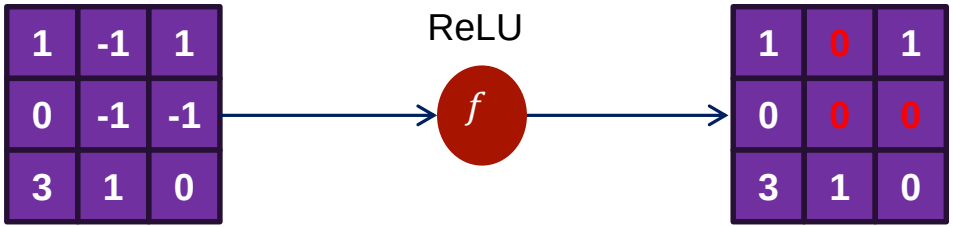


https://predictiveprogrammer.com/famous-convolutional-neural-network-architectures-1/

Activation Function

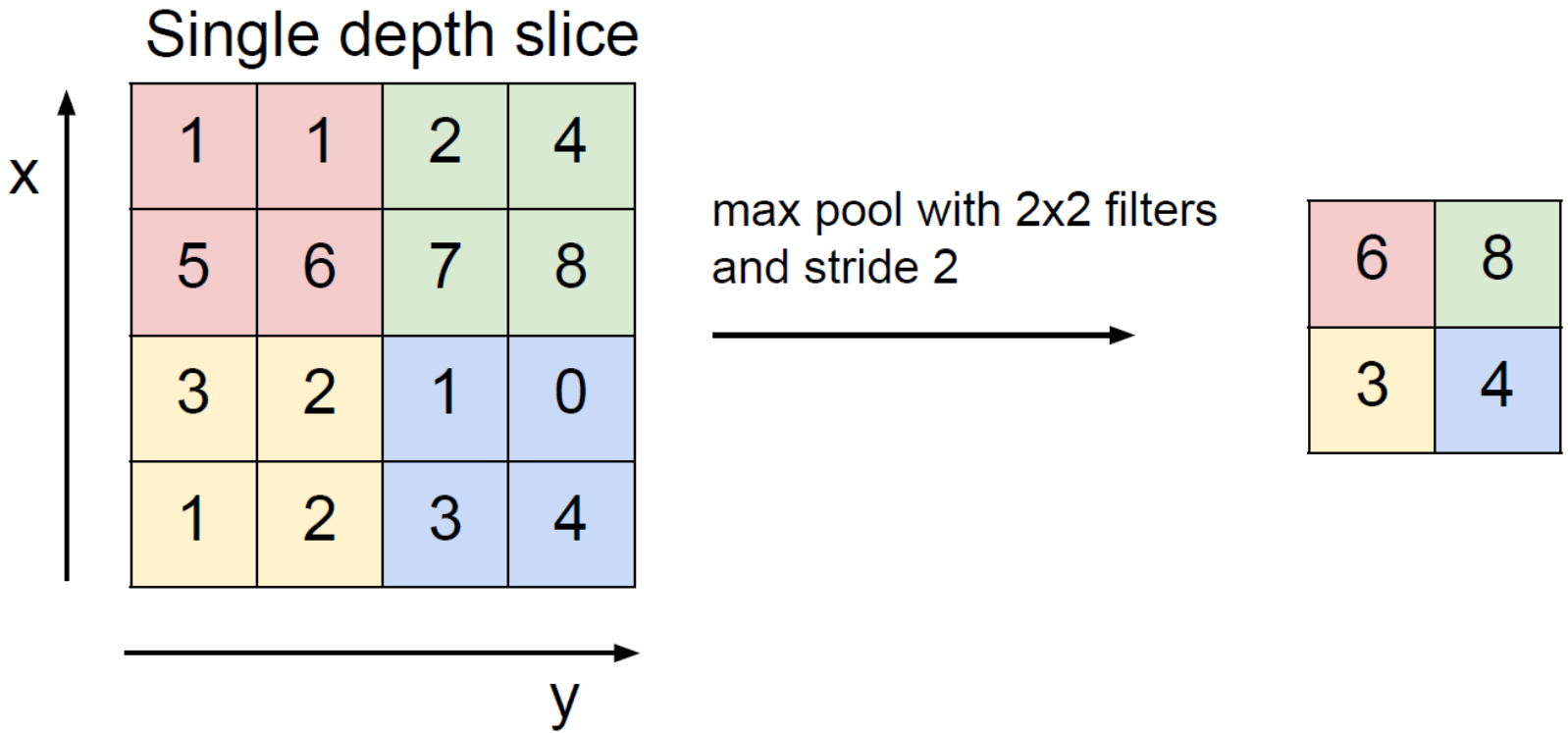
- ReLU





Pooling

- Max Pooling or Average Pooling



Pooling (max pooling, 2x2 filter, stride 1)

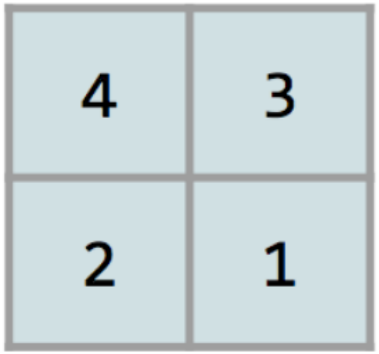
tf.keras.layers.MaxPool2D

```
__init__(
    pool_size=(2, 2),
    strides=None,
    padding='valid',
    data_format=None,
    **kwargs
)
```

tf.keras.layers.MaxPool2D

- `pool_size`: integer or tuple of 2 integers, factors by which to downscale (vertical, horizontal). (2, 2) will halve the input in both spatial dimension. If only one integer is specified, the same window length will be used for both dimensions.
- `strides`: Integer, tuple of 2 integers, or None. Strides values. If None, it will default to `pool_size`.
- `padding`: One of `"valid"` or `"same"` (case-insensitive).
- `data_format`: A string, one of `channels_last` (default) or `channels_first`. The ordering of the dimensions in the inputs. `channels_last` corresponds to inputs with shape `(batch, height, width, channels)` while `channels_first` corresponds to inputs with shape `(batch, channels, height, width)`. It defaults to the `image_data_format` value found in your Keras config file at `~/.keras/keras.json`. If you never set it, then it will be "channels_last".

Max Pooling



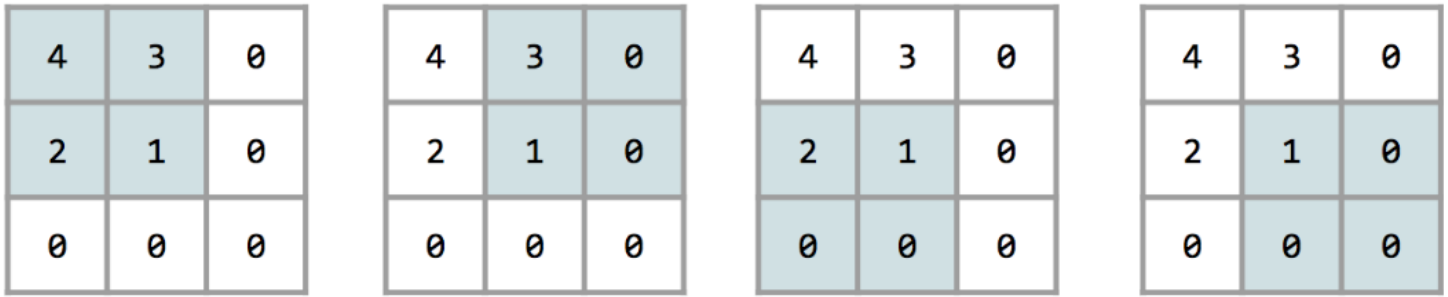

```
image = tf.constant([[[[4],[3]],
                      [[2],[1]]]], dtype=np.float32)
pool = keras.layers.MaxPool2D(pool_size=(2,2), strides=1,
       padding='VALID')(image)
print(pool.shape)
print(pool.numpy())
```

```
(1, 1, 1, 1)
[[[[4.]]]]
```

Max Pooling

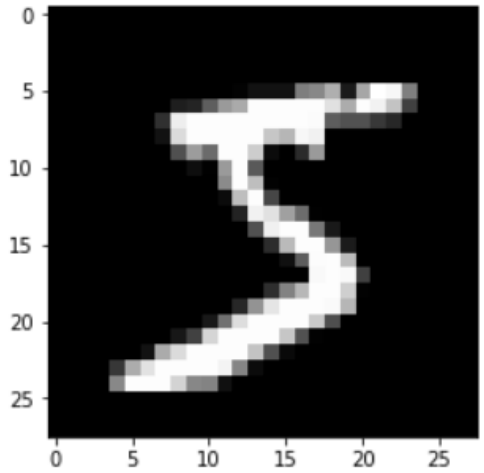
```
image = tf.constant([[[[4],[3]],
                      [[2],[1]]]], dtype=np.float32)
pool = keras.layers.MaxPool2D(pool_size=(2,2), strides=1,
       padding='SAME')(image)
print(pool.shape)
print(pool.numpy())
```

```
(1, 2, 2, 1)
[[[[4.]
   [3.]]

  [[2.]
   [1.]]]]
```



Loading MNIST Data

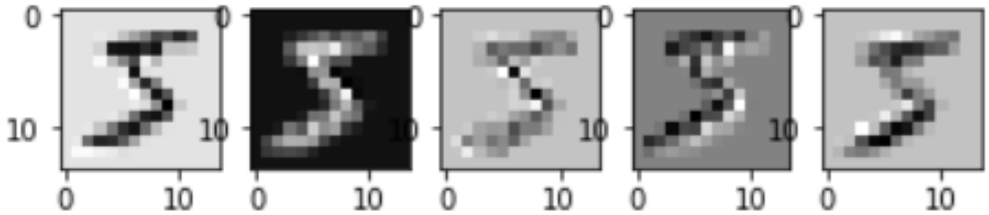
```
mnist = keras.datasets.mnist
class_names = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']

(train_images, train_labels), (test_images, test_labels) =
mnist.load_data()

train_images = train_images.astype(np.float32) / 255.
test_images = test_images.astype(np.float32) / 255.

img = train_images[0]
plt.imshow(img, cmap='gray')
plt.show()
```



Convolution Layer – Output Feature Maps

```
img = img.reshape(-1,28,28,1)
img = tf.convert_to_tensor(img)
weight_init = keras.initializers.RandomNormal(stddev=0.01)
conv2d = keras.layers.Conv2D(filters=5, kernel_size=3, strides=(2, 2),
       padding='SAME', kernel_initializer=weight_init)(img)
print(conv2d.shape)
feature_maps = np.swapaxes(conv2d, 0, 3)
for i, feature_map in enumerate(feature_maps):
   plt.subplot(1,5,i+1), plt.imshow(feature_map.reshape(14,14),
cmap='gray')
plt.show()
```

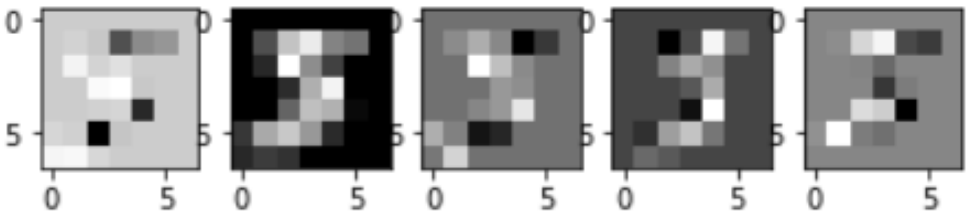
```
(1, 14, 14, 5)
```



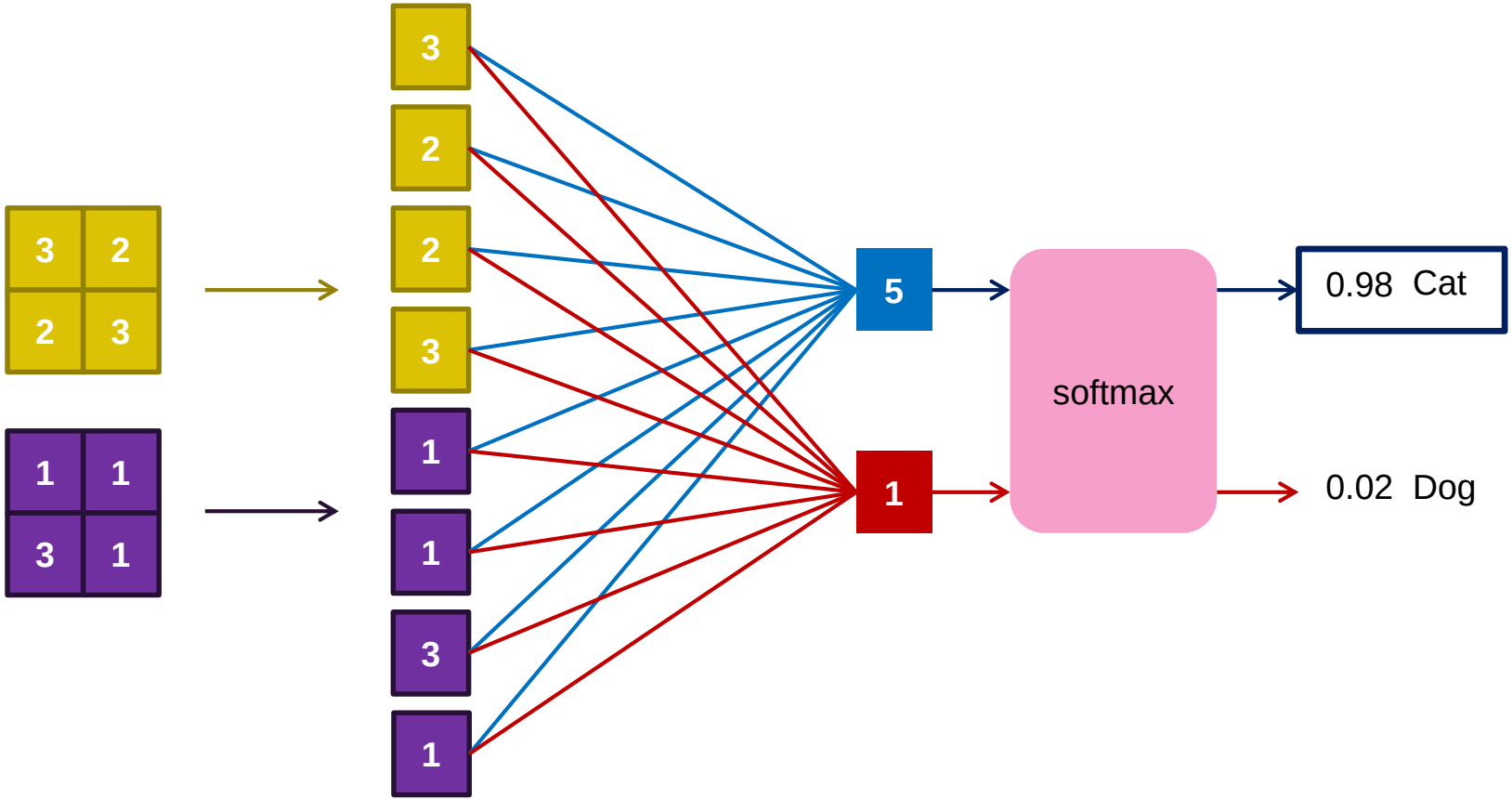
Pooling Layer – Output Feature Maps

```
pool = keras.layers.MaxPool2D(pool_size=(2, 2), strides=(2, 2),
        padding='SAME')(conv2d)
print(pool.shape)

feature_maps = np.swapaxes(pool, 0, 3)
for i, feature_map in enumerate(feature_maps):
   plt.subplot(1,5,i+1), plt.imshow(feature_map.reshape(7, 7),
cmap='gray')
plt.show()
```

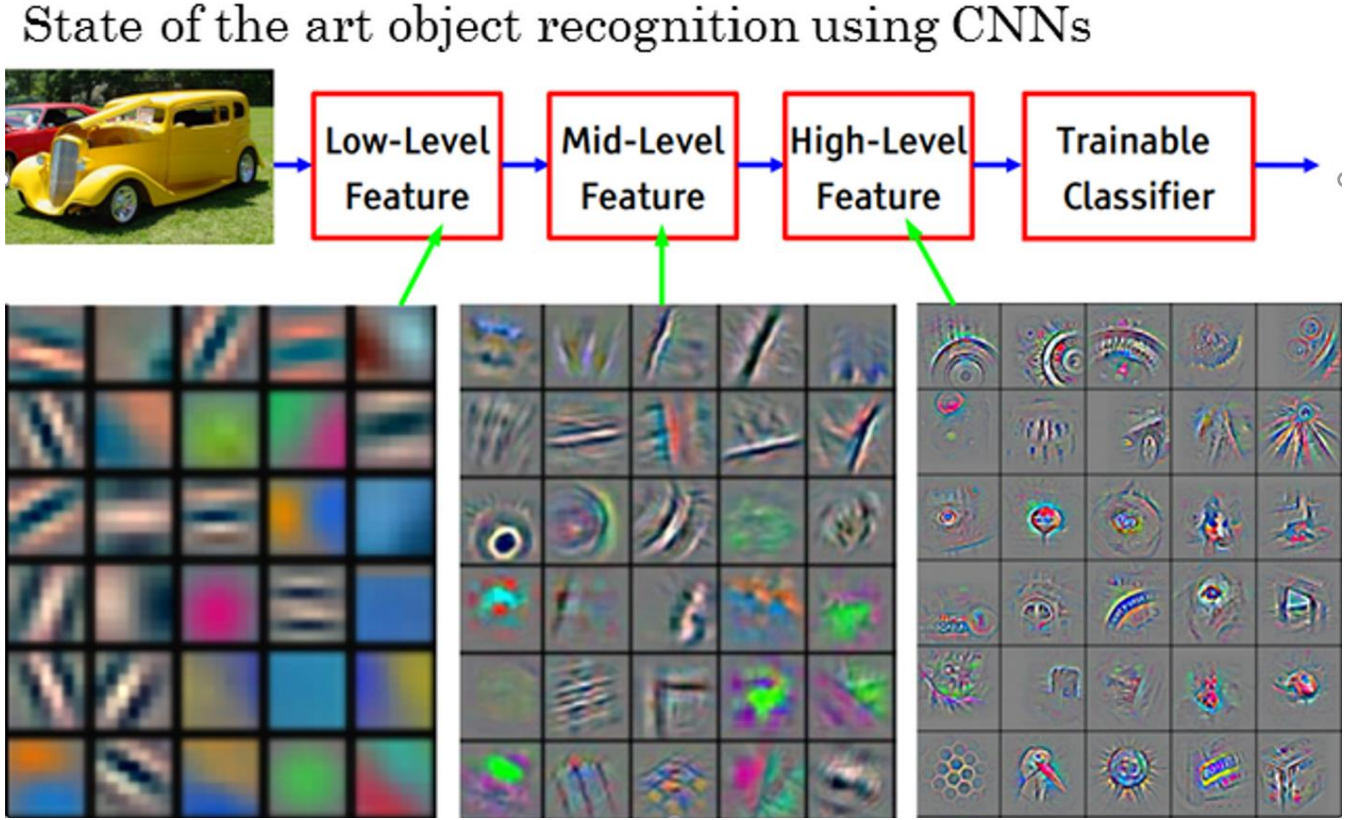
```
(1, 7, 7, 5)
```



Fully Connected(Dense) Layer



Convolutional Neural Network



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

What's Next?

- CNN with MNIST Dataset using tf.keras Squential APIs