Computer Vision: Assignment 3

魏家博 (Chia-Po Wei)

Department of Electrical Engineering

National Sun Yat-sen University

Outline

3a) Image Filtering with Convolutional Layers

3b) Single-Input Single-Output (SISO) and Single-Input Multi-Output (SIMO) CNN Models

Steps for Assignment 3a

1. Open assignment_3a_template.ipynb
2. Use below to create inputs

   ```
   inputs = tf.keras.Input((None, None, 1))
   ```

3. Use below to create outputs

   ```
   outputs = Conv2D(1, kernel_size, stride, padding, name='conv1')(inputs)
   ```

   - `kernel_size` is set to 5 throughout this assignment.
   - You need to determine `stride` and `padding` such that the output shape of Conv2D is the same as the input shape.

4. Use below to create the model

   ```
   filter_net = tf.keras.Model(XXX)
   ```

   - You need to determine XXX.

5. Use below to set the weights and bias of Conv2D

   ```
   filter_net.get_layer('conv1').set_weights([weights, bias])
   ```

   - `bias` is set to `np.array([0])`
   - `weights` is a numpy array of shape `(kernel_size, kernel_size, 1, 1)`
   - Each entry of `weights` has the value `1/kernel_size**2`

Steps for Assignment 3a

6. Use below to get img
   ```
   img = cv2.imread('Winona.jpg', cv2.IMREAD_GRAYSCALE)
   ```
7. `img` is of shape (772, 564)
   - Use `np.expand_dims()` to change its shape to (1, 772, 564)
   - Denote the result by `img_batch`. Note that `img_batch` is of shape (1, 772, 564).
8. Use below to get img_out
   ```
   img_out = filter_net(img_batch)
   ```
   - Reshape `img_out` such that its shape is (772, 564)
   - Use `np.array()` to convert the dtype of `img_out` to `np.uint8`
9. Use `plt.imshow(img_out, cmap='gray')` to display `img` and `img_out` as shown in Figure 1.

[Question 1] If `kernel_size` in Steps 3 and 5 is changed from 5 to 11, how will `img_out` change (more blurry or more clear)? Why?

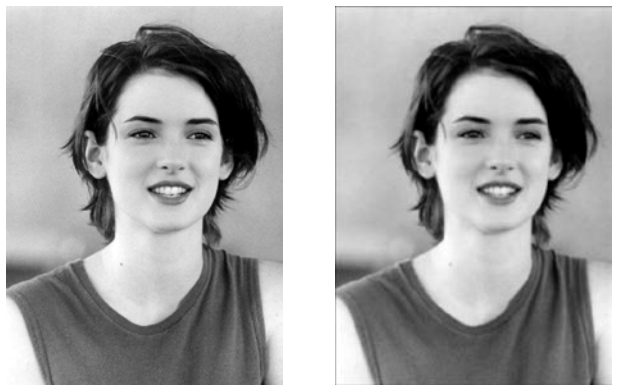

Figure 1

Steps for Assignment 3a

10. Let `img` be defined as in Step 6. Note that `img` is of shape (772, 564).
    - `pad = int((kernel_size-1) / 2)`
    - Create the zero array `img_pad` of shape (772+2*pad, 564+2*pad)
    - `img_pad` is depicted as the dashed rectangle in Figure 2.
    - Assign `img` to the region of solid rectangle in Figure 2.
        - Hint: `img_pad[XXX, YYY] = img`
        - You need to determine XXX and YYY.
11. Repeat Step 2 to Step 5 with the following adjustment:
    - In Step 3, set `stride=1` and `padding='valid'`
12. Change the shape of `img_pad` to (1, 772+2*pad, 564+2*pad) as done in Step 7, and denote the result by `img_batch`.
13. Repeat Steps 8 and 9.

[Question 2] Let Step 2 to Step 9 be denoted as Method 1, and Step 10 to Step 13 be denoted as Method 2.

Why Methods 1 and 2 lead to the same results?

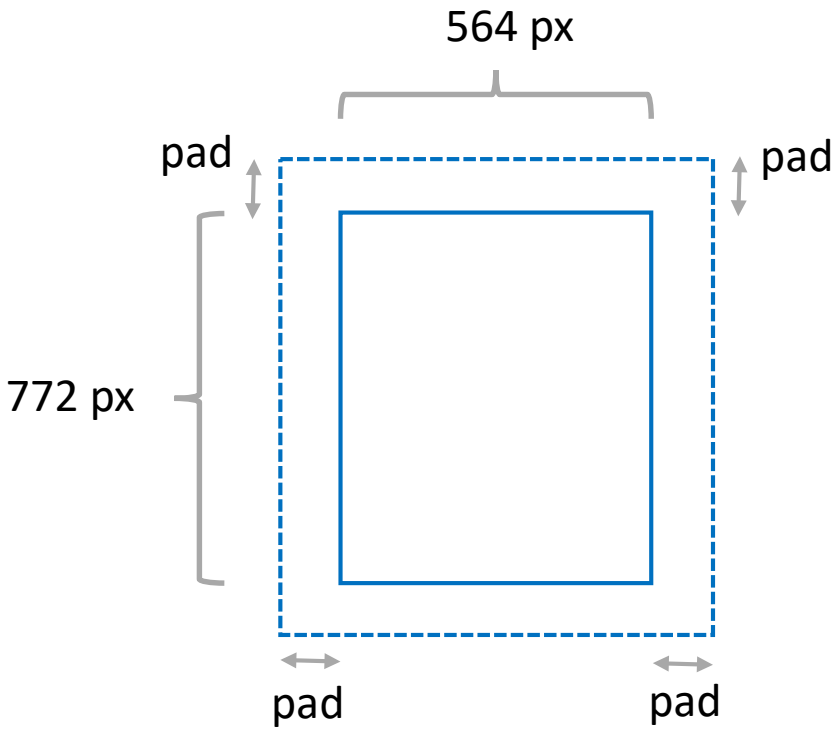

Figure 2

Steps for Assignment 3b

1. Open assignment_3b_template.ipynb
2. Create the CNN model `net48_siso()` that satisfies the following requirements:
   - The input shape of the model is (48, 48, 3).
   - The output shape of the model is (1, 1, 2).
   - It is **prohibited** to use any fully connected layers, Dense().
   - You might need to use `Conv2D()` and `MaxPooling2D()` many times.
   - The kernel_size of `Conv2D()` must satisfy 1 <= kernel_size <= 7
   - The strides of `Conv2D()` must satisfy 1 <= strides <= 2
   - The pool_size of `MaxPooling2D()` must satisfy 1 <= pool_size <= 2
   - The strides of `MaxPooling2D()` must satisfy 1 <= strides <= 2
   - The output of the model is defined by
     - `x = Conv2D(K, F, S, P, activation='softmax', name='cls_output')(x)`
     - You need to determine K, F, S, P such that the output shape is (1,1,2).
     - The choices K, F, S, P depends on previous layers, and thus the choices are not unique.
     - The name of the output layer should be `cls_output`.
     - The activation function of the output layer should be `softmax`. The activation function of other `Conv2D` layers should use `relu`.

Steps for Assignment 3b

3. Create the CNN model `net48_simo()` that satisfies the following requirements:
   - This model needs to satisfy all the requirements in Step 2 of Assignment 4b.
   - This model has two outputs. The first output is the same as that in Step 2 of Assignment 4b.
   - The second output of the model is defined by
     - `x2 = Conv2D(K, F, S, P, name='reg_output')(x)`
     - You need to determine K, F, S, P such that the output shape is (1,1,4).
     - The choices K, F, S, P depends on previous layers, and thus the choices are not unique.
     - The name of the second output layer should be `reg_output`.
     - The second output layer **is not** followed by any activations.
     - You can change the variable name `x2`.
4. Use below to return the model

   `return tf.keras.Model(inputs = inputs, outputs = XXX)`
   - You need to determine XXX.
   - Hint: XXX is a list or a tuple that contains the two outputs.

Steps for Assignment 3b

[Question 3] After defining model48_simo(), consider below

```
img = np.random.randn(1, 48, 48, 3)
outputs = model48_simo.predict(img)
cls = np.reshape(outputs[0], (2,))
reg = np.reshape(outputs[1], (4,))
```

Why `cls[0] + cls[1]` is always equal to one regardless of the values of img?