Discrimination-of-substandard-and-falsified-formulations-from-genuine-drugs-using-NIR-and-ML

CNN models

CNN architectures are consisting of an input layer, convolution layers, pooling, fully connected and an output layer. 1D filters (`Conv1D`) on the convolution layers are used in 1D CNN model for one-dimensional spectral and 2D filters (`conv2D`) is used in two-dimensional modeling.

1) 1D CNN:

The 1D CNN model was trained with an initial learning rate of 0.01 and Adam optimizer. The data is divided into 80% train and 20% test set. Various combinations of number of convolutional layers and fully connected layers ranging from 2 to 4 were tried to see the performance and below model performed better than other models.

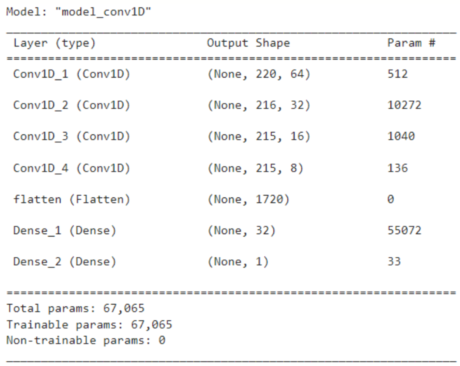
Details of the CNN model is described below. The CNN was implemented in Python using Karas library. Model performance is evaluated using ‘mse’ mean squared error and RMSE on a test dataset.

i. Model name: 1D_CNN_4CL_2DL

Type	Filter size	Kernel	activation	Units
Convolutional	64	7	relu	-
Convolutional	32	5	relu	-
Convolutional	16	2	relu	-
Convolutional	8	1	relu	
Flatten	-	-	-	-
Fully connected			relu	32
Fully connected	-	-	linear	1

ReLU, rectified linear unit

1D_CNN_4CL_2DL Model description:



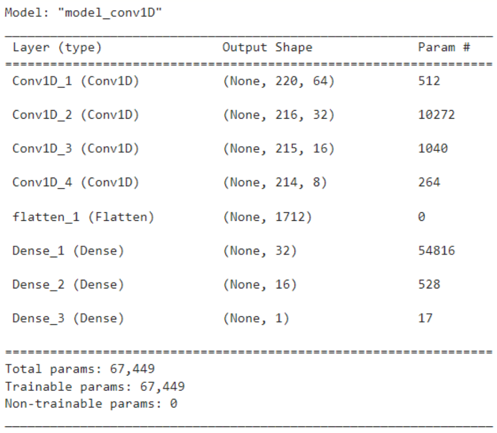
```
Model: "model_conv1D"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 Conv1D_1 (Conv1D)           (None, 220, 64)           512

 Conv1D_2 (Conv1D)           (None, 216, 32)           10272

 Conv1D_3 (Conv1D)           (None, 215, 16)           1040

 Conv1D_4 (Conv1D)           (None, 215, 8)            136

 flatten (Flatten)           (None, 1720)              0

 Dense_1 (Dense)             (None, 32)                55072

 Dense_2 (Dense)             (None, 1)                 33

=================================================================
Total params: 67,065
Trainable params: 67,065
Non-trainable params: 0
_________________________________________________________________
```

ii. Model name: 1D_CNN_4CL_3DL

Type	Filter size	Kernel	activation	Units
Convolutional	64	7	relu	-
Convolutional	32	5	relu	-
Convolutional	16	2	relu	-
Convolutional	8	2	relu	
Flatten	-	-	-	-
Fully connected			relu	32
Fully connected	-	-	relu	16
Fully connected	-	-	linear	1

ReLU, rectified linear unit

1D_CNN_4CL_3DL Model description:



```
Model: "model_conv1D"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 Conv1D_1 (Conv1D)           (None, 220, 64)           512

 Conv1D_2 (Conv1D)           (None, 216, 32)           10272

 Conv1D_3 (Conv1D)           (None, 215, 16)           1040

 Conv1D_4 (Conv1D)           (None, 214, 8)            264

 flatten_1 (Flatten)         (None, 1712)              0

 Dense_1 (Dense)             (None, 32)                54816

 Dense_2 (Dense)             (None, 16)                528

 Dense_3 (Dense)             (None, 1)                 17

=================================================================
Total params: 67,449
Trainable params: 67,449
Non-trainable params: 0
_________________________________________________________________
```

2) 2D CNN:

The 2D CNN model was trained with an initial learning rate of 0.01 and Adam optimizer. The data is divided into 80% train and 20% test set. Data was arranged in 2*2 matrix of (15*15) features. Various combinations of number of convolutional layers and fully connected layers ranging from 2 to 4 were tried to see the performance and below model performed better than other models.

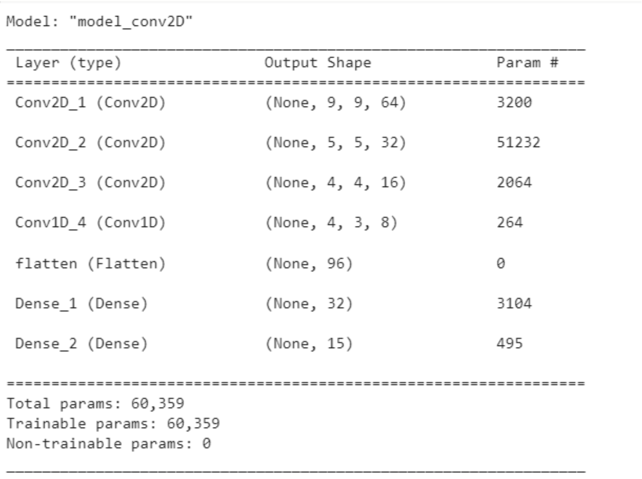
Details of the CNN model is described below. The CNN was implemented in Python using Karas library. Model performance is evaluated using ‘mse’ mean squared error and RMSE on a test dataset.

i. Model name: 2D_CNN_4CL_2DL

Type	Filter size	Kernel	activation	Units
Convolutional	64	7	relu	-
Convolutional	32	5	relu	-
Convolutional	16	2	relu	-
Convolutional	8	2	relu	
Flatten	-	-	-	-
Fully connected			relu	32
Fully connected	-	-	linear	15

ReLU is rectified linear unit

Model Description: 2D_CNN_4CL_2DL



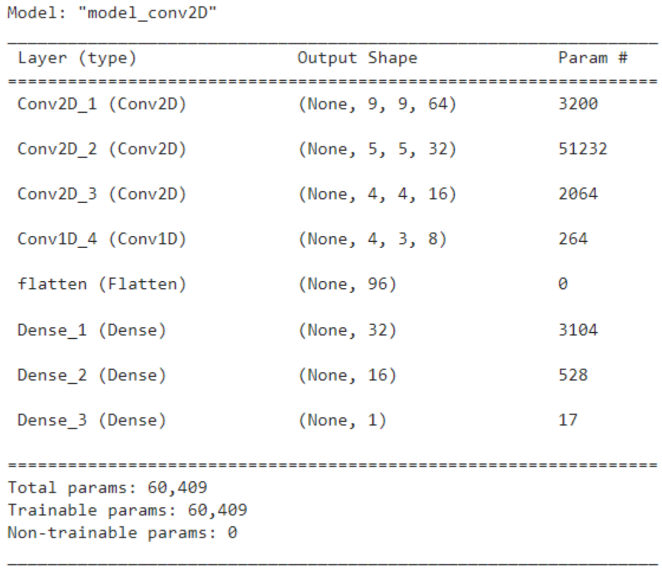
```
Model: "model_conv2D"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 Conv2D_1 (Conv2D)           (None, 9, 9, 64)          3200

 Conv2D_2 (Conv2D)           (None, 5, 5, 32)          51232

 Conv2D_3 (Conv2D)           (None, 4, 4, 16)          2064

 Conv1D_4 (Conv1D)           (None, 4, 3, 8)           264

 flatten (Flatten)           (None, 96)                0

 Dense_1 (Dense)             (None, 32)                3104

 Dense_2 (Dense)             (None, 15)                495

=================================================================
Total params: 60,359
Trainable params: 60,359
Non-trainable params: 0
_________________________________________________________________
```

ii. Model name: 2D_CNN_4CL_3DL

Type	Filter size	Kernel	activation	Units
Convolutional	64	7	relu	-
Convolutional	32	5	relu	-
Convolutional	16	2	relu	-
Convolutional	8	2	relu	
Flatten	-	-	-	-
Fully connected			relu	32
Fully connected	-	-	relu	16
Fully connected	-	-	linear	1

ReLU, rectified linear unit

2D_CNN_4CL_3DL Model description:



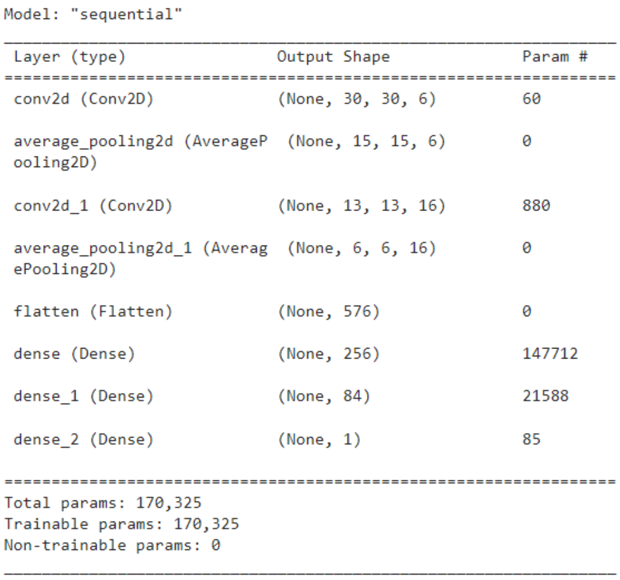
```
Model: "model_conv2D"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 Conv2D_1 (Conv2D)           (None, 9, 9, 64)          3200

 Conv2D_2 (Conv2D)           (None, 5, 5, 32)          51232

 Conv2D_3 (Conv2D)           (None, 4, 4, 16)          2064

 Conv1D_4 (Conv1D)           (None, 4, 3, 8)           264

 flatten (Flatten)           (None, 96)                0

 Dense_1 (Dense)             (None, 32)                3104

 Dense_2 (Dense)             (None, 16)                528

 Dense_3 (Dense)             (None, 1)                 17

=================================================================
Total params: 60,409
Trainable params: 60,409
Non-trainable params: 0
_________________________________________________________________
```

iii. Model name: 2D_CNN_LeNet

Type	Filter size	Kernel	activation	Units
Convolutional	16	3	relu	-
Average Pooling	-	-	-	-
Convolutional	16	3	relu	-
Average Pooling	-	-	-	-
Flatten	-	-	-	-
Fully connected			relu	256
Fully connected	-	-	relu	84
Fully connected			linear	1

ReLU, rectified linear unit

Model description: 2D_CNN_LeNet



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 30, 30, 6)         60

 average_pooling2d (AverageP  (None, 15, 15, 6)        0
 ooling2D)

 conv2d_1 (Conv2D)           (None, 13, 13, 16)        880

 average_pooling2d_1 (Averag  (None, 6, 6, 16)         0
 ePooling2D)

 flatten (Flatten)           (None, 576)               0

 dense (Dense)               (None, 256)               147712

 dense_1 (Dense)             (None, 84)                21588

 dense_2 (Dense)             (None, 1)                 85

=================================================================
Total params: 170,325
Trainable params: 170,325
Non-trainable params: 0
_________________________________________________________________
```