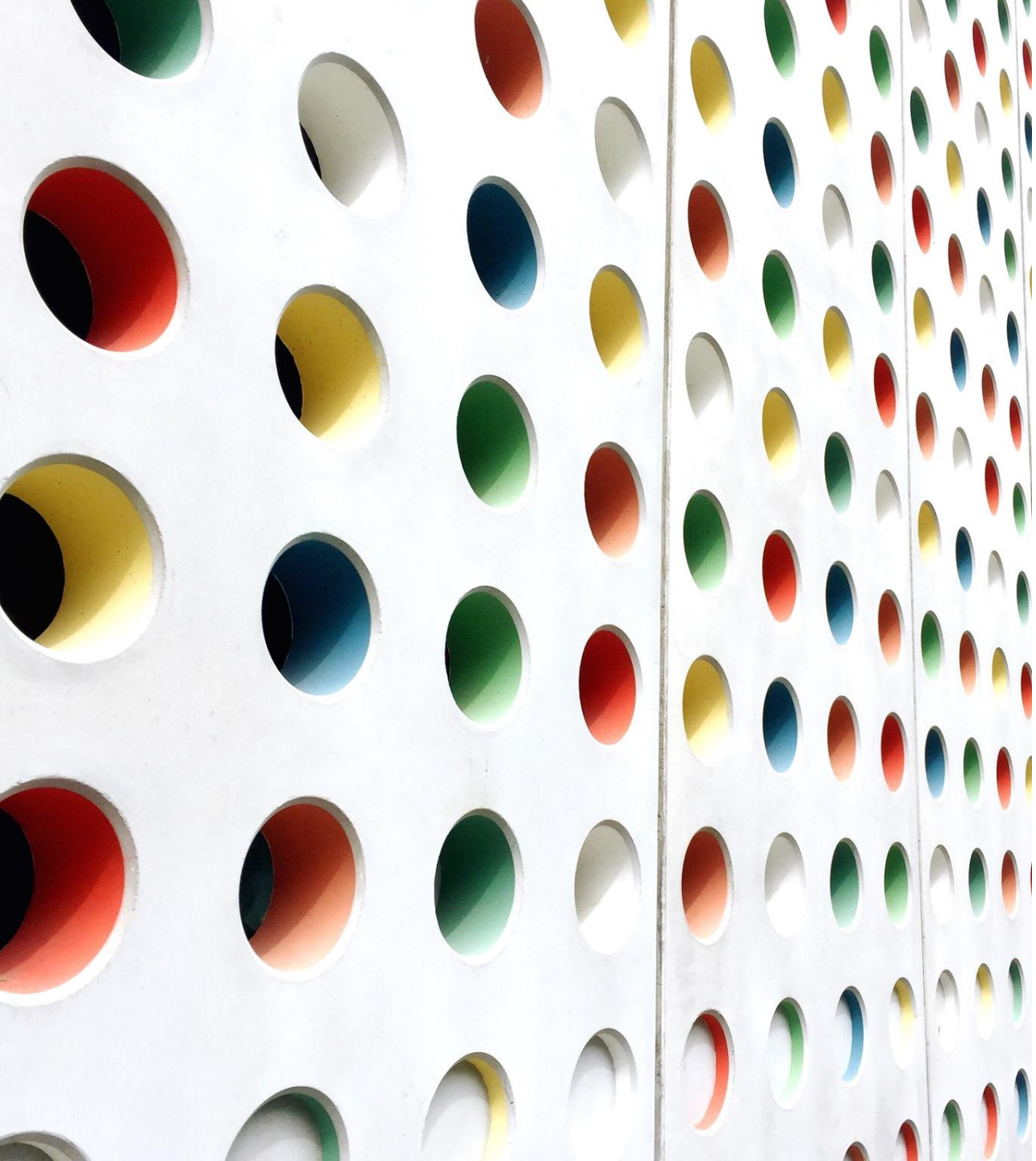


Smartphone Detection of Human Activity

VICTOR SHIH

ORION FERNANDES

Data Set Information

The experiments were carried out with a group of 30 volunteers within an age bracket of 19-48 years. They performed a protocol of activities composed of six basic activities: three static postures (standing, sitting, lying) and three dynamic activities (walking, walking downstairs and walking upstairs). The experiment also included postural transitions that occurred between the static postures. These are: stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie, and lie-to-stand. All the participants were wearing a smartphone (Samsung Galaxy S II) on the waist during the experiment execution. We captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz using the embedded accelerometer and gyroscope of the device. The experiments were video-recorded to label the data manually. The obtained dataset was randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.

Attribute Information

The dataset is then divided in two parts and they can be used separately.

I) Inertial Sensor Data

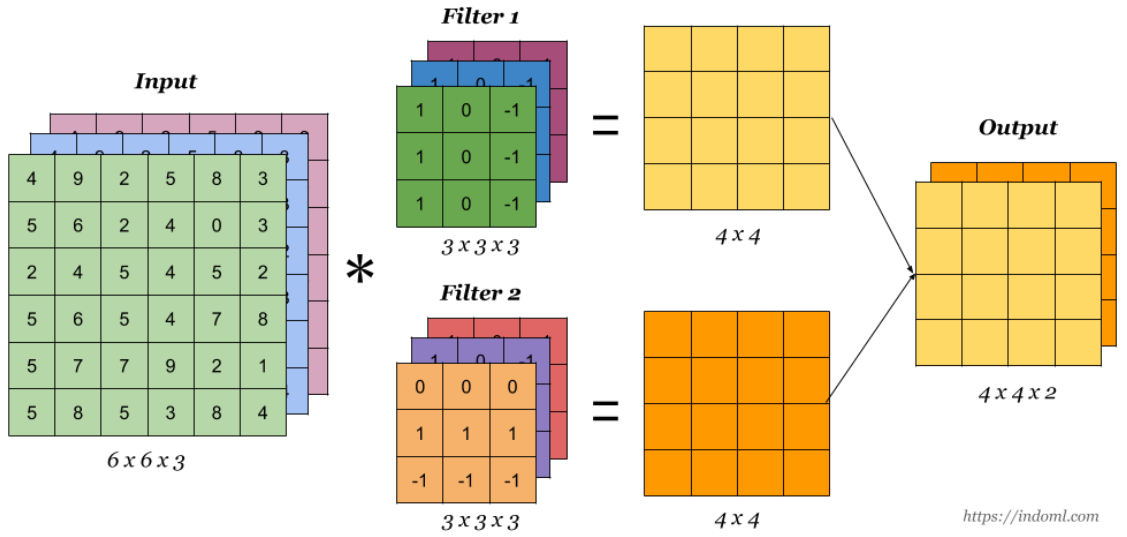
- Raw triaxial signals from the accelerometer and gyroscope of all the trials with participants.
- The labels of all the performed activities.

II) Records of activity windows. Each one composed of:

- A 561-feature vector with time and frequency domain variables.
- Its associated activity label.
- An identifier of the subject who carried out the experiment.

What is a CNN Model ?

CNN is **a type of neural network model** which allows us to extract higher representations for the image content. Unlike the classical image recognition where you define the image features yourself, CNN takes the image's raw pixel data, trains the model, then extracts the features automatically for better classification.



Model Construction

- **1D Convolution (temporal convolution):** This layer creates a convolution kernel that is convolved with the layer input over a single spatial (or temporal) dimension to produce a tensor of outputs. It's great for time data sequences of telemetry data from an accelerometer or gyroscope.
- **Dropout:** This is a regularization layer that essentially throws away a percentage of the weights.
- **MaxPooling1D**: Down samples the input representation by taking the maximum value over a spatial window size.
- **Flatten:** Flattens input without affecting batch size.
- **Dense:** A neural network layer that is connected deeply, which means each neuron in the dense layer receives input from all neurons of its previous layer.

Model Specifications

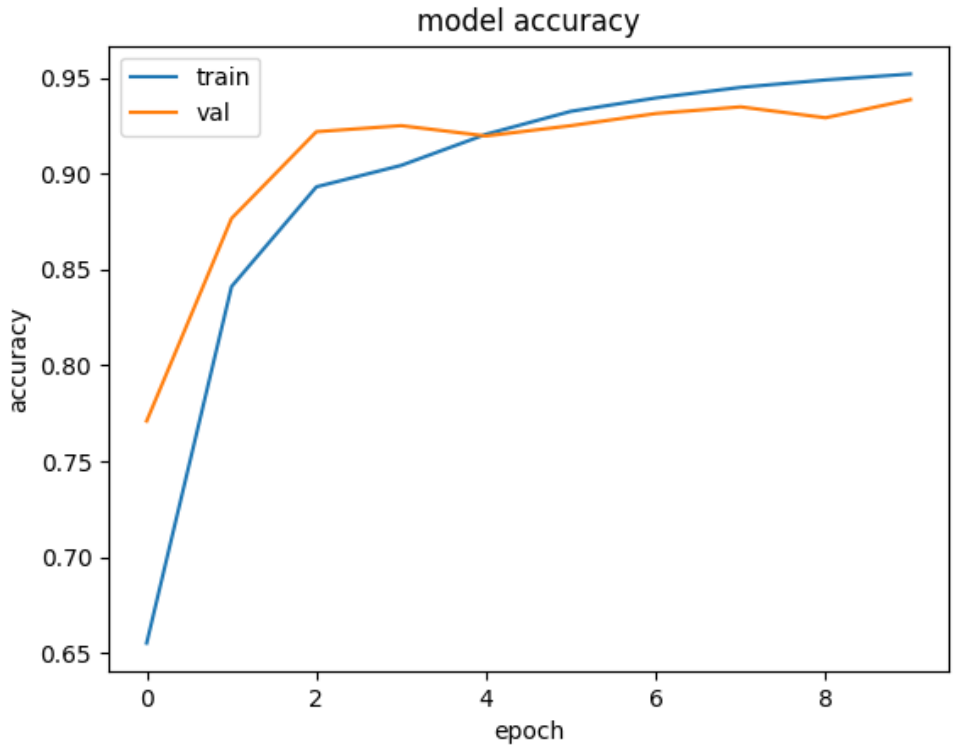
- Activation Function for 1D Convolution – ‘relu’
- Used Cross Entropy for Loss.
- Model fit over 10 epochs

Model Architecture

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 561, 1)]          0
conv1d (Conv1D)              (None, 561, 32)           128
conv1d_1 (Conv1D)            (None, 561, 32)           3104
batch_normalization (BatchN  (None, 561, 32)           128
ormalization)
activation (Activation)      (None, 561, 32)           0
dropout (Dropout)            (None, 561, 32)           0
conv1d_2 (Conv1D)            (None, 561, 32)           3104
conv1d_3 (Conv1D)            (None, 561, 32)           3104
batch_normalization_1 (Batc  (None, 561, 32)           128
hNormalization)
activation_1 (Activation)    (None, 561, 32)           0
dropout_1 (Dropout)          (None, 561, 32)           0
flatten (Flatten)            (None, 17952)             0
dense (Dense)                (None, 64)                1148992
dropout_2 (Dropout)          (None, 64)                0
dense_1 (Dense)              (None, 32)                2080
dropout_3 (Dropout)          (None, 32)                0
dense_2 (Dense)              (None, 12)                396
=================================================================
Total params: 1,161,164
Trainable params: 1,161,036
Non-trainable params: 128
```

Results

Accuracy : train:95.2%, test:93.9%



Loss: train:0.136, test:0.185