The model used is CRNN with recurrent dropout

Brief of Model:

Preprocessing:

- The audio was pre-emphasized with transfer function H(z) = 1-az^-1 with a = 0.97.
- Using librosa library, we obtain the mel frequency spectrogram. We used n_fft=1024, as a result, the number of frames collected from data was 862 frames.
- Parameter n_mels=40 represents the number of filters in mel frequency range.
- The matrix 862x40 was then taken to log scale to remove small noises.
- Then the matrix was normalized to range [-1, 1].
- The matrix is then reshape to 862x40x1 to be used 2D convolution.
- Lastly, we shuffled the data and separate train-test in ratio of 8-2.

Model: The model is CRNN which is described in the paper : E. Çakır, S. Adavanne, G. Parascandolo, K. Drossos, and T. Virtanen, "Convolutional recurrent neural networks for bird audio detection". We feed the network with size 862 x 40 x 1

Detail architecture:

> 4 x (Conv2D(96 features, (5,5), activation="relu", padding="same") + MaxPooling2D + BatchNormalization + Dropout(0.25))
> The max pooling is done on the mel axis, in the following factors: 8, 2, 2, 2
> After 4 convolution layers, the shape of output is (862x1x96). Then it would be reshaped to be 862x96 to feed to RNN;
> Next is 2 layers of GRU(96, return_sequences=true, recurrent_dropout=0.25)
> Then we use a temporal max pooling to merge 862x96 matrix into matrix size 1x96.
> The matrix now is flatten and connected to a Dense node with size is 1, activation function is sigmoid.
> The loss function chosen is "binary_crossentropy", metric=[AUC].
> The optimizer chosen was Adam(learning_rate=0.0008)

The number of epochs chosen was 15. We have realized that the model is vulnerable to be overfitting. The performance was reduced when we attempted to increase the number of epochs. Also, the way we shuffle the data also affect the result significantly. So we keep a seed to make the model reproducible.

We have tried different models before conclude with this one. VGG-16 was tried, but the result was below 60. We also tried and adjusted the model from SpeechLab_UKY_3 in DCASE2018, but the result was also below 60. This might happen due to the amount of train data (we have less), and the way we shuffle and preprocess the data. The preprocessing is so important: adding pre-emphasis step increase the result about 2! When we come to the model in our paper, recurrent_dropout is not supported by CuDNN kernel, so we had tried to use regularizer L2 to two GRU layers. The result was okay, but could not surpass what we have. Moreover, at some points we actually normalized data to range [0, 1] and use LeakyReLU instead, but the results was not as good as our best.