Testing and Implementation of *Sound Event Localization and Detection of Overlapping Sources Using Convolutional Recurrent Neural Networks*

Giuliano Martinelli 1915652 Nicoló Palmiero 1918734

March, 2019

Contents

1 Introduction

In this paper we will report on the reproduction of the results achieved by *Sharath Adavanne, Archontis Politis, Joonas Nikunen and Tuomas Virtanen* as described in the paper *Sound Event Localization and Detection of Overlapping Sources Using Convolutional Recurrent Neural Networks.*
This report aims to implement and test its results in order to have a deeper understanding on the theme and in neural networks in general.

Sound event localization and detection (SELD) is the task of identifying the temporal activities of each sound event, estimating their respective spatial location, and further associating textual labels with the sound events.
In other words this model, that we will start to refer as SELDnet, is able to learn produce two different outputs:
The first is the Sound Event Detection(SED) performed as a multi-class classification.
The second output is the estimation of 3D Cartesian coordinates of the Direction of Arrival(DOA) using multi-class regression.
The structure selected is characterized by the use of an hybrid approach, combining Recurrent with Convolutional layers. More about the architecture of this model are reported in the section about Architecture.
One of the principal issues of sound event detection and localization is due to the fact that in the real world sounds overlap very often, so it is crucial for the model to be trained with this kind of sound events, in order to obtain good results even in case of realistic registrations. For this reason the tests have been performed with different datasets having from 1 to 3 overlapping sound events.

In order to reproduce the results obtained by the paper, we started implementing the model on our machines using some pieces of codes referenced by the paper. Those were incompatible with current versions of python, and it took some time in order for us to make the code suitable for our purposes.
After noticing that preprocessing and training were taking too much time and resources, we opted for a different solution involving Google Colab, which allowed us to use a GPU instead of a CPU in order to speed up processing data.
At the end of this migration on the notebook we settled up the new environment in order to run the model with the support of Google Drive. We then needed to implement and adapt an evaluation process that was more suitable to reproduce the metrics that were reported on the paper, and we opted for a cross-evaluation between the different splits of each dataset.
Once implemented and fitted the model over the different dataset splits, we were able to reproduce the metrics that were cited in the paper. In addition we measured also different common metrics as mean and standard deviation

or confidence intervals.
In order to evaluate graphically the results, we focused on reproducing the same plots that were presented in the original article. This has shown to be a very laborious process that but showed that the results were reproducing in a correct way the ones reported on the paper.

2 Datasets

We decided to use two different datasets that were available online and used in the original paper, ANSIM and RESIM. Those two datasets have a common structure consisting in 3 different sets. Each set consist of a series of audio files with respective ground truth classification and a different number of overlapping sounds, varying from 1 to 3 audio sources that overlaps during the recording.
Their main difference is due to the environment in which they were generated: ANSIM is located in an an-echoic environment while RESIM present more noisy data being recorded in an echoic ambient. In particular RESIM simulates a moderately reverberant room of size $10 \times 8 \times 4$m.
In order to visualize dataset distribution, we have been able to plot some graphs that can explain how are they balanced.
Since Ansim and Resim differ only because of the presence of eco in a reverberant ambient, we will report the results of the distribution of Ansim dataset with max 3 overlapping sound for each frame, that is in practice equivalent to the one of Resim dataset.

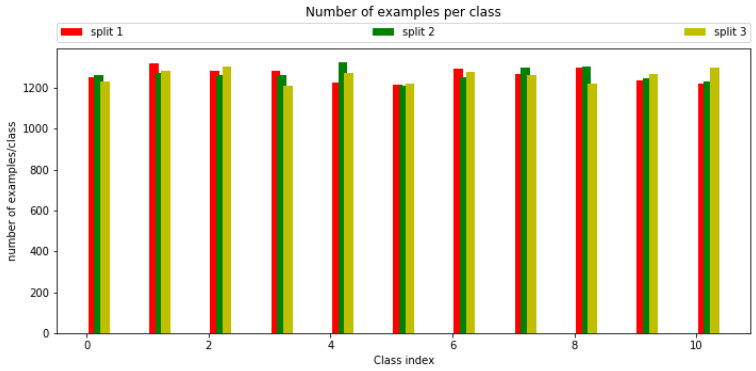


Figure 1: number of samples per class ANSIM overlap 3

In the graphic above we can evaluate the number of each class samples for each split, denoted with a different color. We can appurate that this dataset is balanced when it comes to number of classes represented in it.

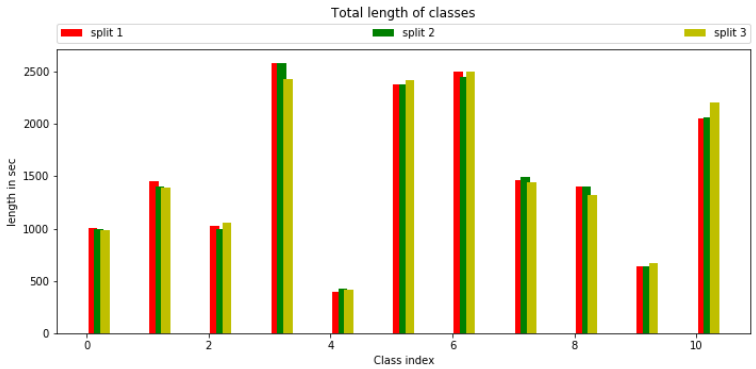


Figure 2: Length in seconds of ANSIM overlap 3 classes

In this plot, we can show how the length of the audio sample changes between the splits. those audio samples are registered in sec, and they don't differ too much from one split to another.

2.1 ANSIM

ANSIM - Ambisonic, An-echoic and Synthetic Impulse Response dataset consists of spatially located sound events in an an-echoic environment. It comprises three subsets: no temporally overlapping sources (O1), maximum two temporally overlapping sources (O2) and maximum three temporally overlapping sources (O3).
Each of the subsets consists of three cross-validation splits with 240 training and 60 testing samples. The dataset is generated using 11 sound event classes such as speech, coughing, door slam, page-turning, phone ringing and keyboard. Each of these sound classes has 20 examples, of which 16 are randomly chosen for the training set and the rest four for the testing set.
In order to generate Direction of Arrival, samples are randomly placed in a spatial grid of 10° resolution along azimuth and elevation, such that two overlapping sound events are separated by 10° and the elevation is in the range of $[-60^\circ, 60^\circ]$. In order to have a variability of amplitude, the sound events are randomly placed at a distance of 1 to 10 m from the microphone.

2.2 RESIM

RESIM - Ambisonic, Reverberant and Synthetic Impulse Response dataset has the same structure and classes of ANSIM dataset, with the only difference being that the sound events are spatially placed within a room using the image source method. It also comprises three subsets: no temporally over-

lapping sources (O1), maximum two temporally overlapping sources (O2) and maximum three temporally overlapping sources (O3). The three cross-validation splits of each of the three subsets is generated for a moderately reverberant room of size $10 \times 8 \times 4$m in order to add noise and generate more realistic data.

3 Feature extraction

In order to use data the first operation that we have to do is *Feature extraction*. This consists in preprocessing the data, obtaining the spectrogram of the audio tracks that will be learned by the neural network.
The spectrogram is extracted from each of the C channels of the multi-channel audio using an M-point discrete Fourier transform (DFT) on Hamming window of length $M = 512$ and 50% overlap. The phase and magnitude of the spectrogram are then extracted and used as separate features. Only the $M/2$ positive frequencies without the zeroth bin are used.
The output of the feature extraction block is a feature sequence of T frames, with an overall dimension of $T \times M/2 \times 2C$, where the $2C$ dimension consists of C magnitude and C phase components.
In terms of practical data dimensions, feature extraction is able to expand our dataset splits from 2 GBs to over 45GBs. Once this operation is done preprocessed data is ready to be fed to the first convolutional layer of our neural network.

4 Architecture

We can use the following figure, shown in the original paper, in order to explain efficiently the structure of the Convolutional Recurrent Neural Network.
The features generated by feature extraction are fed directly to the neural network using DataGenerators.
In the proposed architecture the features in the spectrogram are learned using multiple layers of 2D CNN.
Each CNN layer has $P = 64$ nodes with filters of $3 \times 3 \times 2D$ dimensional receptive fields acting with a rectified linear unit (ReLU) activation.
The use of filter kernels spanning all the channels allows the CNN to learn relevant inter-channel features required for localization, whereas the time and frequency dimensions of the kernel allows learning relevant features suitable for both the DOA and SED tasks.
After each layer of CNN, the output activations are normalized using batch normalization, and the dimensionality is reduced using max-pooling, thereby keeping the sequence length T unchanged.

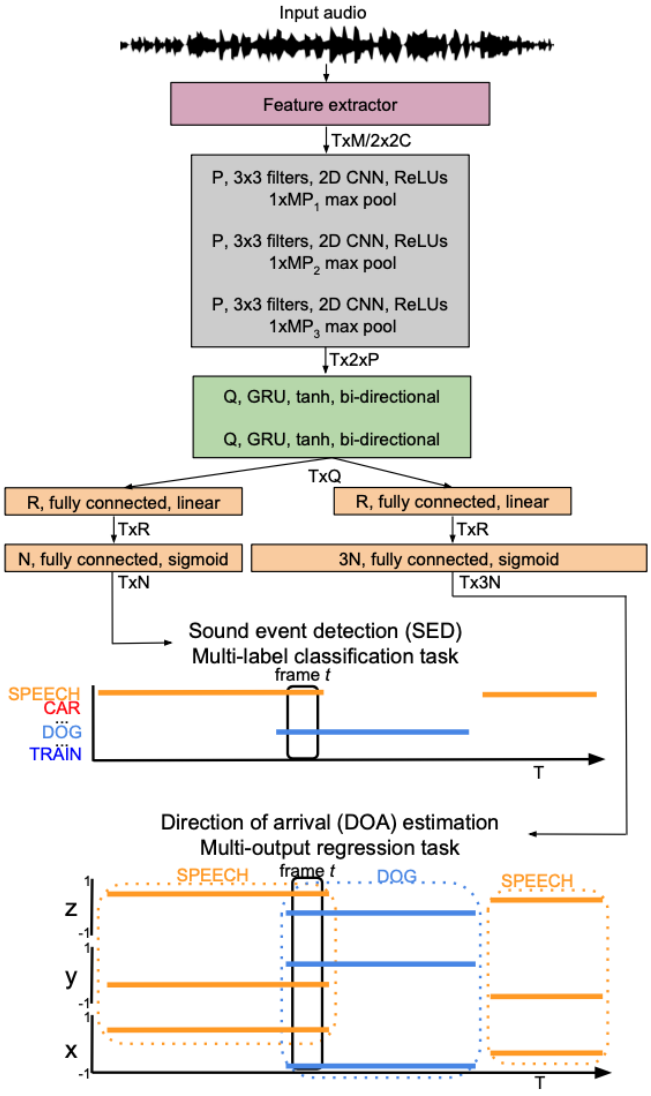


Figure 3: Neural Network architecture

The output of the last CNN layer is introduced into the bidirectional RNN layers. Those are used to learn the temporal context information from the CNN output activations. Specifically each layer has $Q = 128$ nodes of Gated Recurrent Units (GRU) with tanh activation.

At the end of this recurrent layer, the output is separated in two different branches of Fully Connected Layers. The first FC layer in both branches contains $R = 128$ nodes with linear activation. The last Fully Connected layer in the SED branch consists of N nodes with sigmoid activation, each corresponding to one of the N sound event classes to be detected while the last FC layer in the DOA branch consists of 3N nodes with tanh activation.

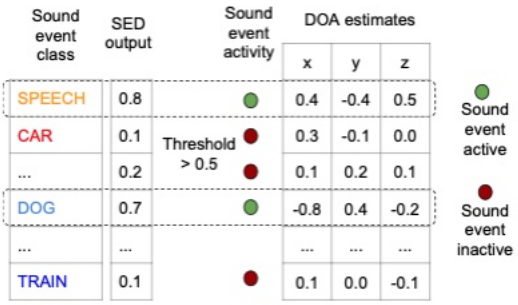


Figure 4: Neural network output

The output of the two Fully Connected Layers can be reported using the following image. We refer to the above architecture as SELDnet.
The SED output of the SELDnet is in the continuous range of [0,1] for each class, while the DOA output is in the continuous range of $[-1, 1]$ for each coordinate.

5 Training

We train the SELDnet with a weighted combination of MSE and binary cross-entropy loss for 200 epochs using Adam optimizer.
We decided to implement 80 20 division of the data between training and testing. Early stopping is used to control the network from over-fitting to training data.

```
37/37 [==============================] - 35s 945ms/step
SED Metrics: ER_overall: 0.283224818077365, F1_overall: 0.8396537878111893
DOA Metrics: doa_loss_gt: 0.4060365070828753, doa_loss_pred: 0.34829429458004835, good_pks_ratio: 0.6692554370777027
epoch_cnt: 68, time: 747.42s, tr_loss: 0.38, val_loss: 1.11, F1_overall: 0.84, ER_overall: 0.28, doa_error_gt: 0.41,
saved model for the best_epoch: 57 with best_metric: 0.21339723782931397,
```

Figure 5: One caption from the training phase

The training is stopped if the SELD score that was previously reported does not improve on the test split for 10 epochs. This score measures all the relevant data and depends on the errors produced by SED and DOA prediction. more about the SELD score in the metrics subsection.
The network was implemented using Keras library with GPU accelerated TensorFlow backend.

6 Evaluation

6.1 Metrics

SELDnet is evaluated using different metrics for SED and DOA estimation. For SED we error rate (ER) and F-score calculated at intervals of one second. The F-score is calculated as

$$F = \frac{2 \cdot \sum_{k=1}^{K} TP(k)}{2 \cdot \sum_{k=1}^{K} TP(k) + \sum_{k=1}^{K} FP(K) + \sum_{k=1}^{K} FN(K)} \tag{1}$$

where the number of true positives TP(k) is the total number of sound event classes that were active in both ground truth and predictions for the k-th one-second interval. The number of false positives in a segment FP(k) is the number of sound event classes that were active in the prediction but were inactive in the ground truth. Similarly, false negatives FN(k) is the number of sound event classes inactive in the predictions but active in the ground truth. Instead, the error rate is calculated as

$$ER = \frac{\sum_{k=1}^{K} S(k) + \sum_{k=1}^{K} D(k) + \sum_{k=1}^{K} I(k)}{\sum_{k=1}^{K} N(k)} \tag{2}$$

where, for each one-second interval k, N(k) is the total number of active sound event classes in the ground truth. S(k), D(k) and I(k) are the number of substitutions, deletions, and insertions in an interval given by

$$S(k) = min(FN(k), FP(k)) \tag{3}$$

$$D(k) = max(0, FN(k) - FP(k)) \tag{4}$$

$$I(k) = max(0, FP(k) - FN(k)) \tag{5}$$

The predicted DOA estimates (x_E, y_E, z_E) are evaluated with respect to the ground truth (x_G, y_G, z_G) used to synthesize the data, utilizing the central angle $\sigma \in [0,\ 180]$. The σ is the angle formed by (x_E, y_E, z_E) and (x_G, y_G, z_G) at the origin in degrees, and is given by

$$\sigma = 2 \cdot arcsin(\frac{\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}}{2}) \cdot \frac{180}{\pi} \tag{6}$$

where, $\Delta x = x_G - x_E$, $\Delta y = y_G - y_E$, and $\Delta z = z_G - z_E$. The DOA error is then calculated as:

$$DOAerror = \frac{1}{M} \cdot \sum_{m=1}^{M} \sigma((x_G^m, y_G^m, z_G^m), (x_E^m, y_E^m, z_E^m)) \tag{7}$$

where M is the total number of DOA estimates across the entire dataset, and $\sigma((x_G^m, y_G^m, z_G^m), (x_E^m, y_E^m, z_E^m))$ is the angle between m-th estimated and ground truth DOAs. Additionally, in order to account for time frames where the number of estimated and ground truth DOAs are unequal, we report the frame recall, calculated as $TP/(TP + FN)$ in percentage, where true positives TP is the total number of time frames in which the number of DOAs predicted is equal to ground truth, and false negatives FN is the total number of frames where the predicted and ground truth DOA are unequal. The DOA estimation method is jointly evaluated using the DOA error and the frame recall. During the training of SELDnet, we perform early stopping based on the combined SELD score calculated as

$$SELD\ score = (SED\ score + DOA\ score)/2 \tag{8}$$

where

$$SED\ score = (ER + (1 - F))/2 \tag{9}$$

$$DOAscore = DOA\ error/180 + (1 - frame\ recall)/2 \tag{10}$$

6.2 Cross-validation

In order to obtain more accurate results from the model that we trained with the different datasets, we decided to implement cross-validation metrics between the splits of the same overlapping sounds. In other words, for each Dataset and Overlapping sound sampleset, we trained one model for each split(3 for each overlap).

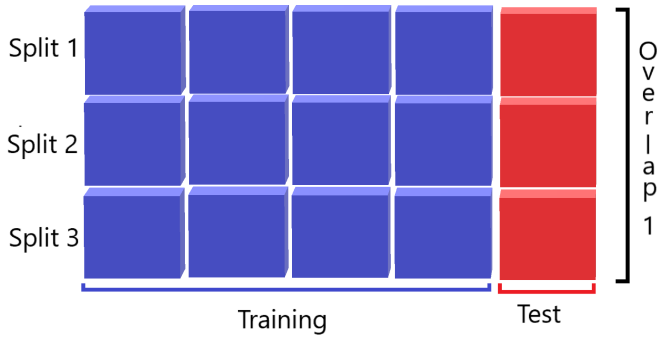


Figure 6: Test/training sets of each split

We then selected the best model between the three testing it with not only the test-set of his split, but using all the three test-sets.
Implementing so we have a better and mean value of the score in the three splits, that sometimes differ in classes and features.

7 Results

In this section we will report our results in comparison with the ones that were obtained in the original paper.
In particular we will firstly discuss about the model important metrics that we introduced in the previous chapter pointing out that we were in most of the cases not far from their scores.
We then will introduce the plots that we implemented in order to reproduce the ones presented in the original report. Those graphics will be able to explain visually how well our model performed in the different output predictions in comparison with the ground truth.

7.1 Metrics

In this table we can compare the results of the paper and our experiment.

		ANSYN			RESYN		
	Overlap	1	2	3	1	2	3
Original Paper	Error rate	**0.04**	0.16	**0.19**	**0.10**	0.29	**0.32**
	F-Score	**97.7**	89.9	85.6	**92.5**	79.6	76.5
	DOA error	**3.4**	**13.8**	**17.3**	**9.2**	**20.2**	**26.0**
	Frame recall	**99.4**	**85.6**	**70.2**	**95.8**	**74.9**	**56.4**
Our experiment	Error rate	0.08	**0.15**	0.20	0.12	**0.28**	0.35
	F-Score	94.8	**90.4**	**88.2**	92.4	**83.3**	**78.6**
	DOA error	13.8	27.3	33.0	23.9	37.0	45.1
	Frame recall	98.2	83.0	69.4	95.6	74.5	56.3

The main metrics we will measure are F-score and Error Rate for SED and Error and Frame Recall for DOA. More about those metrics can be found in the metrics paragraph.
Starting with ANSIM Dataset, we can immediately appreciate how our results were close in terms of metrics with the ones obtained in the original paper.
While the tests made on the first overlap show that the results obtained were really close but always lower in all the metrics, we were also able to obtain better results in some of the voices of Overlap 2 and 3.
For what concerns RESIM dataset tests, we can make the same observations. Overlap 1 had similar results while overlap 2 and 3 where similar and somehow better in some of the voices.
Another piece of data that we were able to confirm is the one that states how the metrics lack of accuracy in presence of echoic ambient. RESIM results

are always less accurate and this was predictable from the beginning. Echo in fact reduces accuracy since it introduces noise in the spectrogram.
One result to point out present in all the metrics is the presence of a bigger DOA error (always circa 10 points) in contrast with the recall that is really close to the values of the original paper.
Without considering those results, that had in most of the cases a real small improvement, we can consider ourself satisfied since we obtained a close score in all the voices of the table.

	ANSIM		
Overlap	1	2	3
SED error (95%)	0.032 ± 0.008	0.20 ± 0.01	0.35 ± 0.02
SED mean accuracy	0.97	0.80	0.64
SED crossval standard deviation	0.008	0.03	0.03
DOA mean variance	0.11	0.26	0.34
DOA crossval standard deviation	0.02	0.04	0.016

	RESIM		
Overlap	1	2	3
SED error (95%)	0.063 ± 0.007	0.31 ± 0.015	0.56 ± 0.016
SED mean accuracy	0.93	0.68	0.43
SED crossval standard deviation	0.021	0.040	0.029
DOA mean variance	0.18	0.42	0.59
DOA crossval standard deviation	0.05	0.04	0.027

In order to better know the performances of our model we decided to implement some other metrics and statistics with respect to the ones presented in the original paper. In particular we implemented the metrics that are represented in the table above.
SED error (95%) is the measure of the SED error, calculated as $E = \frac{bad\ predictions}{number\ of\ predictions}$, with a confidence interval of 95%. This represents an interval in which the error could be with 95% probability. The confidence interval is computed as $\pm const \cdot \sqrt{(error * (1 - error)/n)}$, where n is the total number of samples in the test set of the dataset, and *const* is an empirical value indicating the percentage of the confidence interval (for 95% $const = 1.96$).
SED mean accuracy is the measure of the average SED classification accuracy between the splits of the dataset. This metrics is represented by this formula: $SED\ accuracy = \frac{good\ predictions}{number\ of\ predictions}$. SED crossvalidation standard deviation is the standard deviation of the accuracy between the different values calculated as mean accuracy from the crossvalidation in the

previous metric.
For DOA we implemented two other metrics: DOA mean variance, and DOA crossvalidation standard deviation.
DOA mean variance represents how much every DOA prediction differs from the ground truth value, while DOA crossvalidation standard deviation is the metric indicating the standard deviation of the variation between the different values obtained by the crossvalidation.

Looking at the results we had, we can say that they confirm the good approximation presented by the metrics used in the paper.
The SED accuracy values are good and change in a predicted way, decreasing with the number of overlap.
The same can be said for the DOA variance, that increases with the number of overlapping sources. The error and confidence intervals follow the same obvious pattern.
Concerning the two deviations between the crossvalidation, we can say that the values doesn't change much in the different splits. But we could say that in advance since, as we saw in the subsection regarding dataset, the three splits are pretty similar and balanced.

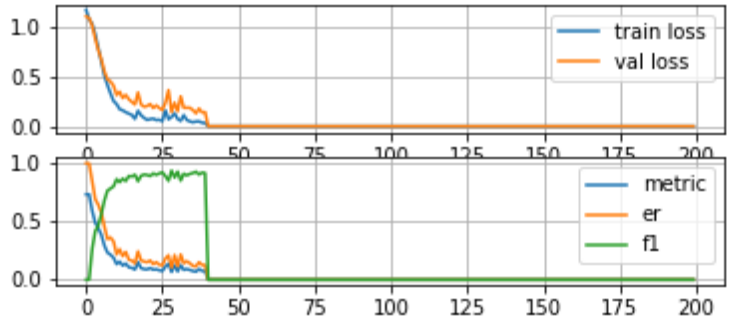
7.2 Plots

In this section we will present our results in a more reader-friendly way using plots: they will be about the training phase and predictions our neural network does.

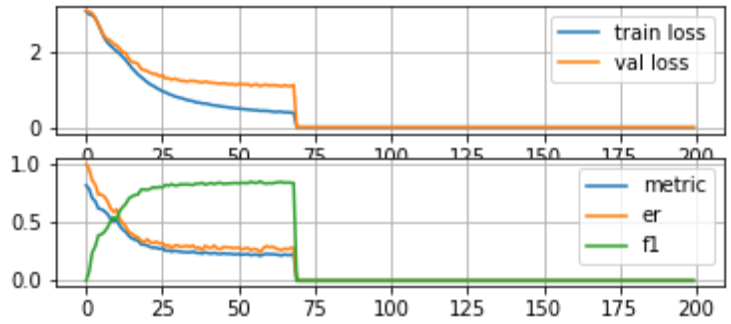
7.2.1 Training Plots

Training plots are thought to monitor the training phase when it is running, in fact SELDnet updates these plots at the end of every epoch.
When the training phase is over, this kind of plots are even more useful to draw conclusions about the training and adjust it in case of strange behaviours or as quality certificates in case of good results.
In the next graphics we will report about two meaningful plots: ANSIM overlap 1 and overlap 3. These two plots had a longer training phase hence the reader can better appreciate the trend of these plots.

Those two plots represent the trend of metrics and loss curves for the training of ANSIM ov1 and ov3.
In general we can observe how the two graphs, although similar in trends, present different thresholds. It is due to the fact that overall metrics reach a way lower accuracy when more sounds overlap.
For what concerns the plots, we are able to divide them in three main categories. The first one represents the trend of the loss curves during the training: from it we can observe that when the model starts to overfit



(a) Trends of the training phase of the model trained on ANSIM_ov1



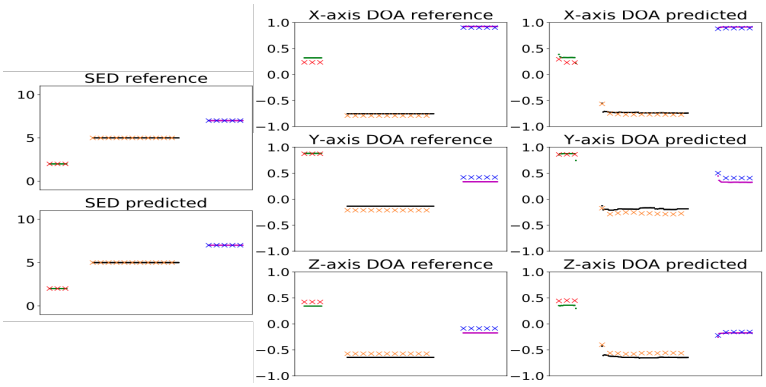
(b) Trends of the training phase of the model trained on ANSIM_ov3

Figure 7: Two samples of training plots

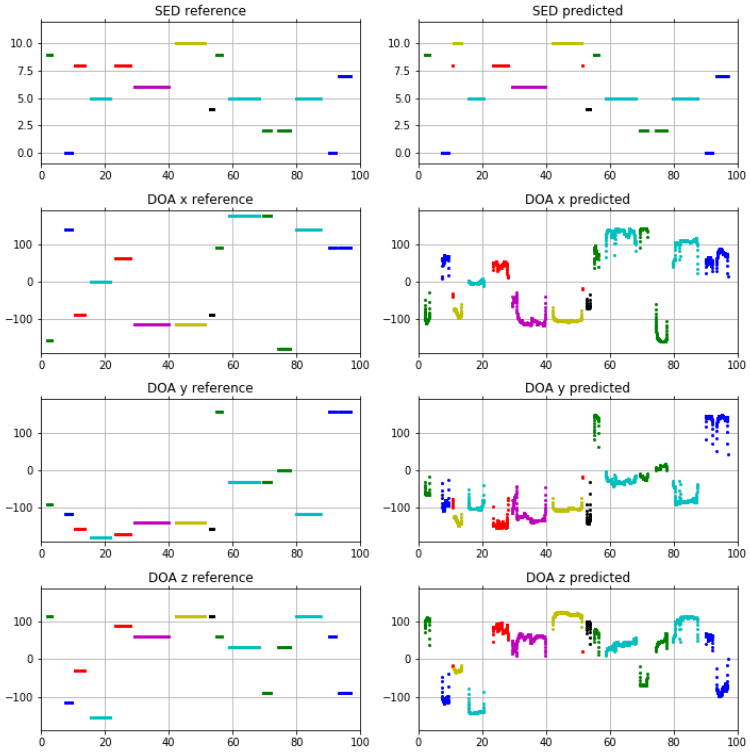
the training ends. this can be observed looking at how the training loss decreases and the validation don't in the last epochs of training and is due to the implementation of early stopping and patience.
The second plot represents the functions generated from the score assigned to the SED task: F-score and SED error. The F-score and the error, as expected, go under the biggest variations within the first epochs, while then they stabilize around their final value.

7.2.2 Prediction Plots

To better visualize the results of our experiment we made some plots starting from our models' predictions.
Those graphics represent the difference between the neural network outputs and ground truth data. The graphics labelled with "SED reference" and "SED predicted" report the comparison between ground truth and the neural network predicted multiclass classification.
The other graphics deal with DOA prediction, that is composed by three plots that represent the three coordinates of the direction of arrival x, y, z on a [-180, 180] interval. In general we can observe how the accuracy becomes lower with the augmentation of the overlaps.

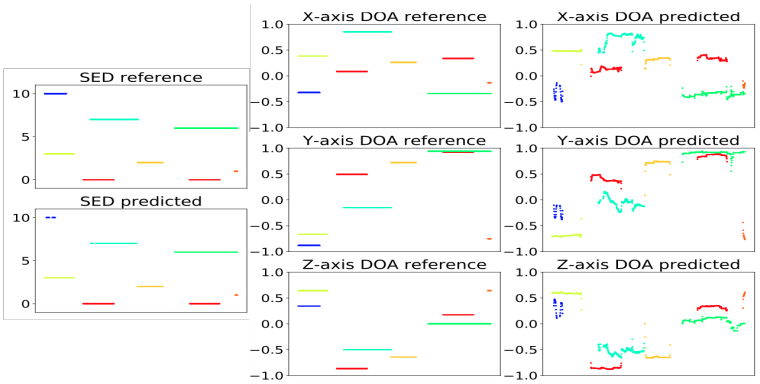


(a) Paper's comparison between ground truth and predictions ANSIM overlap 1

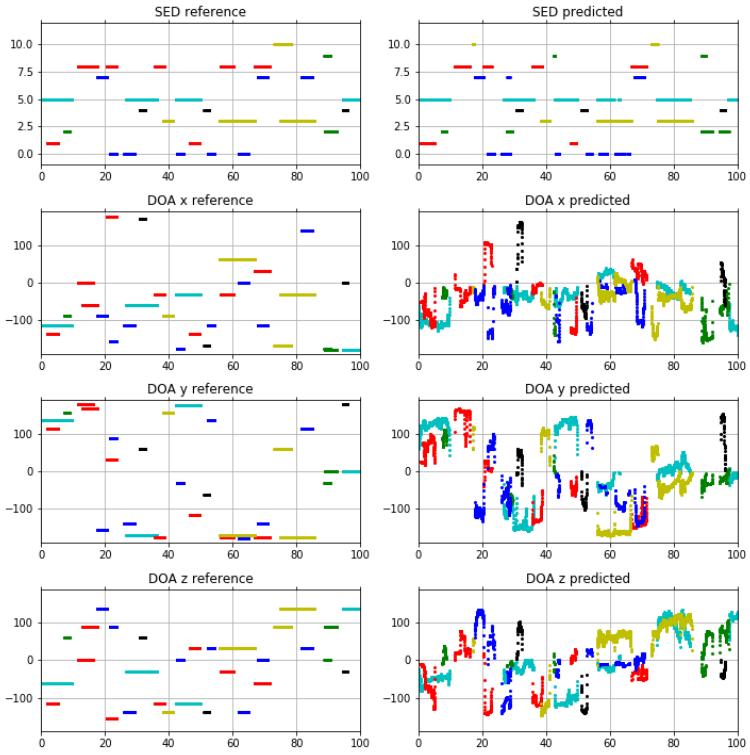


(b) Our comparison between ground truth and predictions ANSIM overlap 1

Here is the first comparison between our plots and the ones from the paper over the model trained on no overlapping sounds in ANSIM dataset. We couldn't know on what sound paper plots were built on, so we decided to pick random files from the dataset in order to give a comparison that can still produce observation on the performances of our model.
From this graph we can observe how close guesses of our model are comparing them with the one made on the original model. The model is very precise processing sounds without overlapping sources in an anechoic ambient.



(a) Paper's comparison between ground truth and predictions ANSIM overlap 2



(b) Our comparison between ground truth and predictions ANSIM overlap 2

In this figure we have the same comparison as before, but carried out on ANSIM overlap 2, so on a dataset with two overlapping sounds.
Here, while the SED predictions, even introducing some false positives and false negatives, remain very close to the ground truth, DOA recall becomes smaller, and this can be seen in three DOA prediction plots: in fact although the overall trend continues to be acceptable the accuracy of the directions on every axis begins to be not as much reliable as before.
With two overlaps these plots suggest that the neural network yields an

interval of values in which there is the right DOA instead of returning a single precise value. Fortunately these intervals are small enough so DOA returned by the neural network can still be considered reliable, with a small grade of error.
Also the ones from the paper seem to suffer from the same issue, this was expected, because these plots reflect the metrics we gave in the previous section.

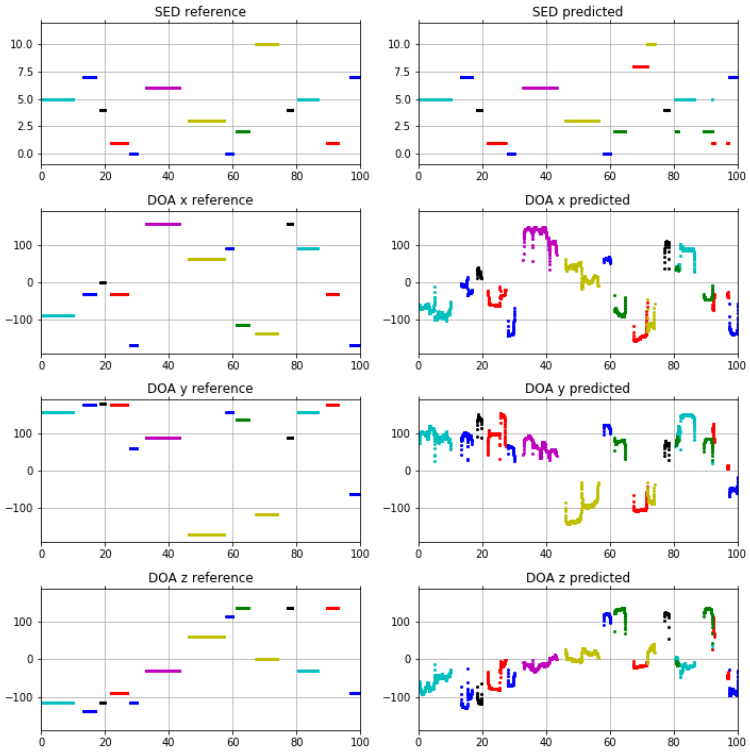


Figure 10: Comparison between ground truth and predictions RESIM ov1

These plots graphs the output of a model trained on RESIM with one overlap.
This was not in the original paper, but it is interesting to compare these plots with the ones about ANSIM overlap 1, because the two datasets in question are quite similar: the only difference is that RESIM is registered in an echoic ambient, while ANSIM is not.
Taking this into account we expected this model to be a bit less precise than the one trained on ANSIM, and this as we can see from the plots is true, but in this case the RESIM model behaved quite well and the prediction are close to the ground truth.

7.3 Conclusions

In this report we tried to reproduce the Sound Event Localization and Detection of Overlapping Sources Neural Network by *Sharath Adavanne, Archontis Politis, Joonas Nikunen and Tuomas Virtanen* on Google Colab.
This experiment gave us the opportunity to have an hands-on experience on a neural network project that is dealing with the state of the art for what concerns sound event localization and detection.
Moreover it taught us to use different machine learning tools such as plots, graphics and standard or custom metrics.
In the experiment we trained a neural network model on two different datasets: ANSIM and RESIM. Both had different sections with one, two or three overlapping sounds, so we have created different models trained on these different portions of these datasets.
We then evaluated those models using the custom metrics of the paper and some metrics used for general evaluation and machine learning statistics.
Another important feedback was given by the graphics. We implemented them using the *matplotlib* library and the results showed a good prediction accuracy of the results in a more quantifiable way.
In conclusion we can say that our experiment had good results since we were able to obtain results that are really close to the ones showed in the paper. There are some differences, mainly in the value of DOA error that is always higher than their values by 10 percentage points.
We had also some positive surprises regarding the values of some metrics results that were slightly higher in our experiment then the ones that were reported in the paper. We also decided to use some statistical metrics in order to evaluate the model, that confirmed that the trainings obtained the good results in term of accuracy and confidence intervals.
For what is concerning the plots that we created, they can be visually compared with the one of the paper and offer a good display of the prediction process, showing some false positives and little errors.
We can be overall satisfied of the results are satisfied of the results and the experience we obtained from this experiment.