Spectral Detection and Localization of Radio Events with Learned Convolutional Neural Features

Timothy J. O'Shea
Virginia Tech, ECE
Arlington, VA
oshea@vt.edu

Tamoghna Roy
Virginia Tech, ECE
Blacksburg, VA
tamoghna@vt.edu

Tugba Erpek
Virginia Tech, ECE
Arlington, VA
terpek@vt.edu

***Abstract*—We introduce a method for detecting, localizing and identifying radio transmissions within wide-band time-frequency power spectrograms using feature learning using convolutional neural networks on their 2D image representation. By doing so we build a foundation for higher level contextual radio spectrum event understanding, labeling, and reasoning in complex shared spectrum and many-user environments by developing tools which can rapidly understand and label sequences of events based on experience and labeled data rather than signal-specific detection algorithms such as matched filters.**

I. Introduction

Understanding what is going on in the radio spectrum is a key enabler of making efficient use of it. The ability to detect, identify and predict the access strategies of others in the band such as the presence of unexpected interference or spurious emissions are key to being able to react intelligently to such phenomenon and to adapt waveform parameters, channel access parameters, or other strategies driven by end-user performance requirements.

Meanwhile a strong analogue for this technical task exists in computer vision, which has matured rapidly over the past several years. Object identification and localization within imagery has been a key enabler for numerous autonomous systems and autonomous control systems. Methods for object detection, classification, and localization have in recent years shifted from more traditional image-feature extraction and higher level classification and localization logic [1], into end-to-end learned features and activation maps [2], [3]. This approach is quite exciting for other domains such as radio, as it does not rely on any imagery-specific feature or logic engineering, but instead learns features, class mapping, and localization generally for a 2D image-like input without significant over-specialization. This allows for a critical building block in contextual understanding of objects in a scene which can be in our case a spectrogram in time-frequency rather than a traditional 2D image.

II. Background

As we are leveraging a computer vision approach here for the radio spectrum sensing domain in which it has not been widely applied before, we introduce some background for each task separately before describing our approach. We have previously explored the areas of isolated single-carrier radio signal classification using convolutional networks on raw RF sample data [4], as well as of learning new radio communications systems using a similar approach [5], but this represents our first efforts into the area of complex multi-emitter scene understanding from an ML-centric approach.

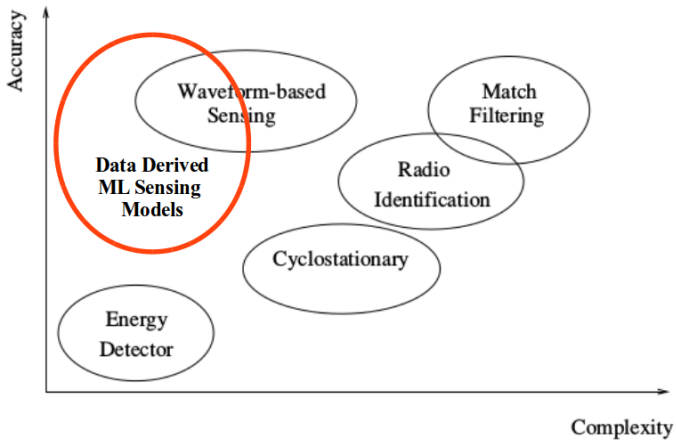

Figure 1. Objective method comparison to prior approaches from [6].

A. Radio Spectrum Sensing

Spectrum sensing has long been a core piece of cognitive radio [6] and a core enabler of how to decide and act within the spectrum to optimize for some end task. Much prior work and deployment of solutions has focused on so called *Waveform-based Sensing* methods, because they generally provide the best sensitivity when compared with simpler *Energy Detector* methods. In our approach, we focus on a variation of the Energy Detector, which performs pattern recognition on emissions within the wide-band spectrum in order to identify RF emissions not only based on the presence of energy, but also on its shape matching some expected pattern. By doing so we hope to provide significant accuracy improvement over simplistic energy thresholding methods, while also maintaining low model complexity due to the end-to-end learning nature of the approach and lack of need for any waveform-based algorithm specialization or tuning.

B. Visual Object Detection & Localization

Within the past 5-10 years, virtually all state of the art computer vision benchmark entries have transitioned to deep convolutional neural network (CNN) [7] based models [8], [9]. This replaces many years of domain-specific low-level

ISBN 978-0-9928626-7-1 © EURASIP 2017

feature engineering which was previously pervasive in the leading approaches. This can be thought of as an analogue to *Waveform-based Sensing* in which heavily signal-specific features are engineered with expert knowledge of the waveform contents i.e., reference tones, preambles, symbol rates, etc.

Detection and localization of objects in a scene generally takes on two classes of approach. First there are those which associate a simple label with the entirety of the picture/scene, i.e. 'this picture contains a cat', and secondly there are those which associate specific object classes with bounding box labels within the scene to provide more rich and accurate label information, i.e 'there is a cat contained within the bounding box given by (72,38,92,56)'.

For now we focus on the first of these two cases, as datasets are a critical limiting factor for building models for either approach. We build a dataset conforming to the first labeled data model (without bounding boxes) within this work, but we seek to address both learning models in future work.

The typical neural network model here is described by $\hat{Y} = f(X;\theta)$ where θ comprises the set of weights of a sequence of neural network layers. The typical variable shapes of these inputs and outputs are given in I.

Table I. TABLE INPUT/OUTPUT SHAPES

Variable	Shape
X	$[n_{channels}, n_{rows}, n_{cols}]$
Y & $\hat{Y}$	$[n_{classes}]$

A solver such as Adam [10] is used to perform gradient descent and iteratively solve for an optimal θ value to fit the dataset by minimizing a loss function. In this scenario a typical loss function would be categorical cross entropy given in 1 where labels and output values approximate the probability each class is present.

$$\mathscr{L}_{ce}(u,v) = -\Sigma_{i=0\ldots n_{classes}1-1}\,(u_i log(v_i)) \qquad (1)$$

Once such a network is learned it can easily be used for classification by prediction the most probable class via $\hat{y}_{ML} = argmax_i(\hat{Y})$. However in this case, the learned features contributing to the probability of each class probability are the items of most interest here.

There are a variety of techniques for object localization with this loosely labeled data i.e. label for the entire image is available but not the individual bounding boxes. In general they focus on building a neural network to form a classifier of objects in the scene, and then further using properties of network to perform localization.

One of the simplest methods to observe how the class probability surface relates to regions of input is by masking the input image (occluding all but a small patch) and observing how the probability of the prevailing class changes spatially. This has been shown to work [2] and provides an extremely simple method for using such a classifier for localization.

Class saliency maps [11] uses a technique similar to the deconvolution [12] method used for visualizing a CNN. In another technique, the fully connected layers are replaced with global average pooling layers [13] to obtain the class activation maps (CAM). Both of these methods assume that the fully connected layers are preceded by convolutional layers.

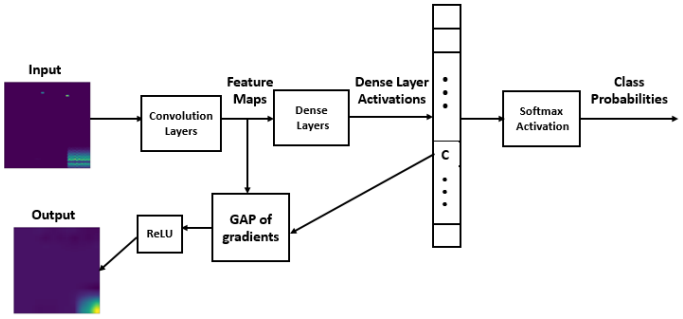

Figure 2. Block Diagram of Grad-CAM [14]

Gradient-weighted class activation maps (Grad-CAM) [14] are a more general approach which can be extended to different CNN based network architectures. A variant of the method called guided Grad-CAM produces high resolution class activation maps. In this work we will be using Grad-CAM to do the spectral event localization.

III. APPROACH

For computer vision tasks such as image recognition, publicly available datasets like Imagenet [15] are available which contains millions of labelled images. These datasets have been leveraged to train very deep networks [16], [17]. Moreover, availability of pre-trained network has facilitated transfer learning [18] between related vision tasks. Since neither is available for our problem, we start from designing a network which is commensurate with the available dataset.

A VGG [16] type of architecture is adopted i.e. convolution layers followed by dense layers leading upto the activation layer. Number of convolution layers, number of dense layers and number of kernels or feature maps in individual layers are all hyper-paramters. For this work, these parameters were chosen in an ad-hoc manner. Table II shows the network configuration. Note that dropout units were used in the convolution layer unlike [16]. This is done to improve the generalizability of the network. Removal of the dropout units from the convolution layers affected performance (verified by simulation). Each of the convolution layer is followed by a max-pooling layer of size 2x2 with an input stride of 2x2. The drop-out rate if present was set to 0.5. Apart from the last layer which has a softmax activation, the activation units for all the other layers were chosen as Rectified Linear Units (ReLu). The input spectrograms should be of the dimension 128x128.

Table II. TABLE INPUT/OUTPUT SHAPES

Layer Number	Layer Type	Kernel Size	Number of Feature Maps
1,2	Convolution	(3,3)	64
3,4	Convolution	(3,3)	128
5	Dense	n/a	128
6	Dense	n/a	$n_{classes}$

The network described in II is at first trained. Figure 2 shows the block-diagram of the Grad-CAM [14] which is used for spectral event localization. Given the input label C gradient of activation score y^C (not the class probability) are calculated with respect to all the feature maps of a given convolution

ISBN 978-0-9928626-7-1

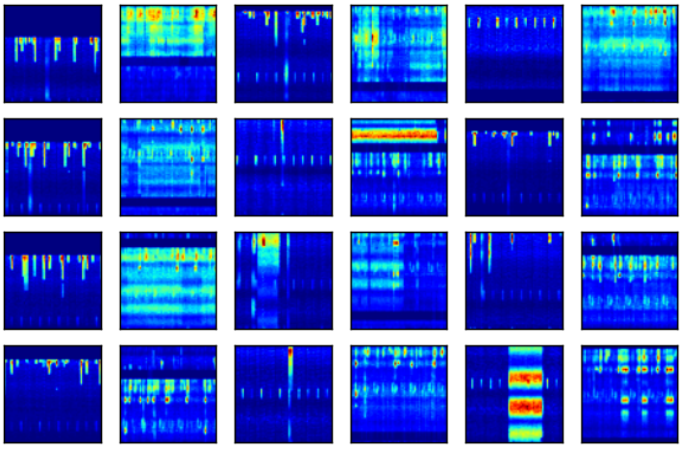

Figure 3. Example spectrograms of randomly sampled RF band samples: Signal time-frequency structures readily recognizable by a domain expert including 802.11B/G/N, Bluetooth, FD-LTE, QAM, etc

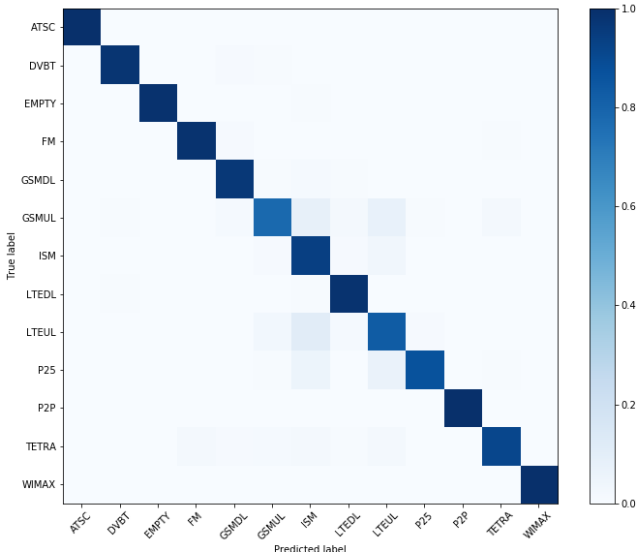

Figure 4. Confusion Matrix for RF Band Classification

layer. The global average pooling [13] of the gradients give the corresponding weight associated with the feature map. Finally the weighted sum of the feature maps is passed through an elemnt-wise ReLu unit to get the Grad-CAM. Detailed description of the method and its application to different CNN based architectures can be found in [14].

Note that the dimension of the CAM is equal to the output size of the convolution layer. For example, in our case if we choose the last convolution layer in our network (before the max-pooling layer), the size of CAM will be 12x12. This 12x12 map is extrapolated to the input image size which 128x128. Naturally because of this extrapolation, the obtained map is coarse. One of our future research directions will be to implement methods such as Guided Grad-CAM [14] which has a better resolution.

IV. Dataset

We construct a dataset using real radio spectrum data which has been manually labeled rather than attempt simulation. We focus on using a typical low-cost integrated RF transceiver, the AD9361 [19] on a Universal Software Radio Peripheral (USRP) B205-mini board [20] to capture RF data, and attempt to cover a wide range of the VHF, UHF, and SHF signals it can observe including all of the spurious noise, interference and distortion present within the environment and receiver system when doing so.

To add to the variation within the dataset, we manually build a labeled list of active bands in an environment (for several dense urban environments), and then we sample it randomly using a random distribution in center frequency and gain around the ideal values in order to create variations in the dataset. (signals are not always at the same relative frequency offsets, gain-related interference, distortion and sensitivity effects are sampled randomly rather than trained for one specific [contrived] set of values.)

The dataset contains 8512 spectrograms from 13 different bandssuch as GSM, LTE, ISM, FM etc. These are obtained from 8 different locations across 5 distinct cities to provide for emitter, layout, and loading variation. Each spectrogram is assigned one unique label. Note that no other information regarding the spectrograms are included in the dataset (center frequency, gain, sample rate, etc are discarded and not used in the model). Spectrograms are computed from 131,072 unique complex samples taken from the radio at a sampling rate between 10MSps to 30MSps. Spectrograms are computed with an fft size of 1024 and an overlap of 900 samples and a hanning window is used. The resulting spectrograms are then stored along with their class labels in smaller re-scaled images of 64x64, 128x128 or 256x256 pixels. Each 'look' at a band constitutes around 4-10ms of observation time, a relatively short time-window which, in the case of bursty protocols such as Wifi, LTE Uplink, or GSM Uplink is in some cases completely unoccupied during some observations.

A. Python Numpy-UHD

We introduce a new software tool called numpy-UHD (npuhd) [1], which provides a rapid means for sampling RF spectrum data using the USRP platform and interfacing with python, numpy and a variety of python-based machine learning tools extremely readily. A similar approach can easily be accomplished with GNU Radio [21], but is slightly more verbose and intended for streaming application than for rapid asynchronous sampling of the spectrum.

V. Results

A. Band Classification

The dataset is split into training and validation portions containing 6384 and 2128 samples respectively. The split is done randomly. Since the network is designed to accept inputs of dimensions 128x128, the training samples are rescaled to that dimension.

Availability of a large training set is one of the prerequisites of training the CNNs. To artificially increase the training set, each sample is flipped about the vertical axis. This

[1] available at https://github.com/radioML/npuhd upon publication

ISBN 978-0-9928626-7-1 © EURASIP 2017

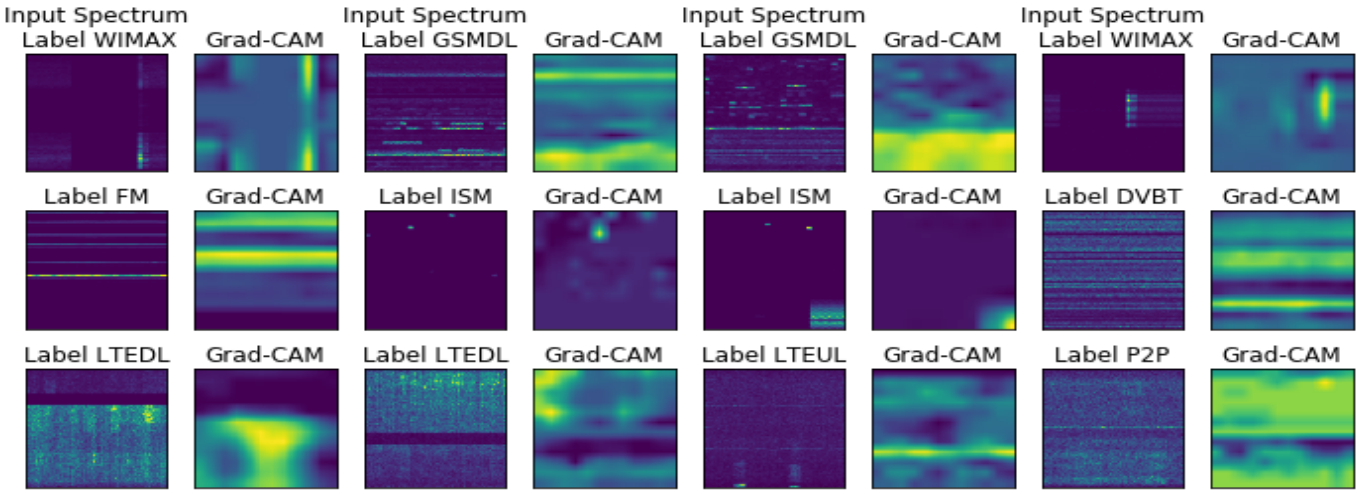

Figure 5. GradCAM based activation maps and corresponding input spectrograms for 12 test examples from the dataset.

is a common data augmentation technique used for different vision problems such as image recognition, object localization and classification. Thus, we have a total of 12,768 training samples. The network is trained using Keras [22] package on a single desktop machine with Intel i7-7700 processor and NVIDIA GTX 1080 GPU. To prevent over-fitting early stopping was used with maximum number of epochs set to 200. On an average, training takes around 10 minutes. The process of randomly splitting the dataset was repeated 10 times. Each time the network was trained from scratch. Average classification accuracy of 0.944 with a peak accuracy of 0.953 was obtained. Figure 4 shows the confusion matrix for this task.

Apart from the two uplink bands GSMUL (accuracy = 0.778) and LTEUL (accuracy = 0.828) and P25 (accuracy = 0.87) the individual classification accuracy for all the other bands is greater than 0.91. The two uplink bands are often misclassified as ISM bands, misclassification rate being 0.078 and 0.11 for GSMUL and LTEUL respectively. This is possibly due to the fact that the ISM band contains bursty signals similar to the uplink bands.

The network takes approximately 0.3 ms to classify each sample when processed in a batch. If samples are processed one at a time detection time increases to around 1.5 ms. At this rate real-time operation of such an approach could likely be achieved quite readily. We have not yet even begun to explore optimizations which may improve this such as reduced precision data-types (all work was conducted in float 32) or network distillation which would provide additional reductions to computational requirements and detection time.

B. Spectral Event Localization

We show the results from our GradCAM implementation in figure 5. Each pairwise example shows the input RF spectrogram of the example, followed by the class activation map for the appropriate target class. Here we can see in general we achieve the expected behavior, in the first example for instance, on top of the WiMAX burst in the spectrogram we have a hot region of activation in the WiMAX-class activation map. Likewise with each of the relevant classes shown. However, since the trained feature objective was to classify the band, not to necessarily activate all instances of a certain emission type (it would need labels to know what that meant!) we can see that we have not completely accomplished this goal here. For instance the DVBT activation map highlights only strong parts of the signal, certain parts of the LTE downlink signal seem to be favored for identification (possibly reference tones synchronization signals), and so forth with each different example type.

VI. Analysis & Conclusions

This method for detection, classifying and localizing communications signal emissions within a spectrogram appears to work relatively well in our experiments, providing quite high initial classification rates for most signal types and providing relatively correct activation maps in most cases. The most difficult classes appear to be those employing time-sharing channel access strategies (CSMA in the ISM Band, TDM in the GSM uplink band, and SC-FDMA in the LTE uplink band), we suspect this is partly due to low burst density and examples in which no traffic is present. In this case longer dwells would likely improve performance. Our dataset labeling technique also involved 'loose' labeling, where we simply specified the center frequency and the associated class. We did not label individual bursts or emissions within the data, and so in some cases we have spectrograms which in-actuality belong to 2 or more classes. For instance GSM and LTE signals are often adjacent bands and so while an effort was made to keep the sampler distinctly on one or the other, some examples do contain a mixture of the two signal types on the edges. Likewise, in the ISM band we see a mixture of Wifi, Bluetooth, and other unlicensed emissions which are all lumped into a single 'ISM' class. Therefore the performance of this classifier is quite limited based on the quality of the dataset.

We focused in this work on the generation of class activation maps for each emission type, but we did not yet look at the task of estimating time-frequency bounding boxes on

ISBN 978-0-9928626-7-1 © EURASIP 2017

these activation maps in order to associate a time and frequency with each detection. This process of converting activation maps to quantitative estimates about the bursts present and their spectral location is a critical next step, and one that will allow us to perform most rigorous quantitative comparison to baseline methods such as traditional band estimation accuracy using energy detection.

In future work we hope to improve our dataset size and quality to include more examples, more variation among examples, and more emitters. We also hope to explore more richly labeled datasets, such as manually labeled bounding boxes on different emission types rather than only general band-labels. We believe this method will drastically improve performance, but also will require a significant amount of labor in generating and curating datasets.

REFERENCES

[1] A. Wallack and D. Manocha, "Robust algorithms for object localization," *International Journal of Computer Vision*, vol. 27, no. 3, pp. 243–262, 1998.

[2] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in Neural Information Processing Systems*, 2013, pp. 2553–2561.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[4] T. J. OShea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International Conference on Engineering Applications of Neural Networks*, Springer, 2016, pp. 213–226.

[5] T. J. O'Shea and J. Hoydis, "An introduction to machine learning communications systems," *ArXiv preprint arXiv:1702.00832*, 2017.

[6] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE communications surveys & tutorials*, vol. 11, no. 1, pp. 116–130, 2009.

[7] Y. LeCun, Y. Bengio, *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[9] X. Zeng, W. Ouyang, J. Yan, H. Li, T. Xiao, K. Wang, Y. Liu, Y. Zhou, B. Yang, Z. Wang, *et al.*, "Crafting gbd-net for object detection," *ArXiv preprint arXiv:1610.02579*, 2016.

[10] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ArXiv preprint arXiv:1412.6980*, 2014.

[11] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *ArXiv preprint arXiv:1312.6034*, 2013.

[12] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, Springer, 2014, pp. 818–833.

[13] M. Lin, Q. Chen, and S. Yan, "Network in network," *CoRR*, vol. abs/1312.4400, 2013. [Online]. Available: http://arxiv.org/abs/1312.4400.

[14] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that?" *ArXiv preprint arXiv:1611.07450*, 2016.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: a large-scale hierarchical image database," in *CVPR09*, 2009.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385.

[18] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *CoRR*, vol. abs/1411.1792, 2014. [Online]. Available: http://arxiv.org/abs/1411.1792.

[19] A. D.-R. A. T. AD9361, "Url: Http://www.analog. com/static/imported-files/dataΓ"$026e30f$_sheets/ad9361. pdf (visited on 09/14/08)," *Cited on*, p. 103,

[20] M. Ettus, *Universal software radio peripheral*, 2009.

[21] E. Blossom, "Gnu radio: Tools for exploring the radio frequency spectrum," *Linux journal*, vol. 2004, no. 122, p. 4, 2004.

[22] F. Chollet, *Keras*, https://github.com/fchollet/keras, 2015.

ISBN 978-0-9928626-7-1 © EURASIP 2017