FUNDAMENTAL FREQUENCY CONTOUR CLASSIFICATION: A COMPARISON BETWEEN HAND-CRAFTED AND CNN-BASED FEATURES

Jakob Abeßer[1] *Meinard Müller*[2]

[1] Semantic Music Technologies Group, Fraunhofer IDMT, Ilmenau, Germany
[2] International Audio Laboratories Erlangen, Germany

ABSTRACT

In this paper, we evaluate hand-crafted features as well as features learned from data using a convolutional neural network (CNN) for different fundamental frequency classification tasks. We compare classification based on full (variable-length) contours and classification based on fixed-sized subcontours in combination with a fusion strategy. Our results show that hand-crafted and learned features lead to comparable results for both classification scenarios. Aggregating contour-level to file-level classification results generally improves the results. In comparison to the hand-crafted features, our examination indicates that the CNN-based features show a higher degree of redundancy across feature dimensions, where multiple filters (convolution kernels) specialize on similar contour shapes.

Index Terms— fundamental frequency contours, feature learning, convolutional neural networks, activation maximization

1. INTRODUCTION

In the last years, data-driven algorithms for feature learning based on deep neural networks often outperformed traditional analysis methods that exploit domain expert knowledge. Compared to hand-crafted feature design, data-driven approaches often show superior performance within analysis and classification scenarios. However, as a main disadvantage, learned feature representations often lack a clear interpretation and give only little insight into the problem at hand.

In the field of Music Information Retrieval (MIR), fundamental frequency (f_0) contours, i. e., variable-length time-series representations of the pitch curve of musical notes, are a rich mid-level representation as they provide cues for both music performance analysis and music content analysis [1]. For example, frequency contours have been successfully applied for MIR tasks such as playing and singing style analysis, as well as genre and music instrument classification. In general, a reliable extraction of f_0 contours from polyphonic audio mixtures remains challenging to this day. One open issue is how to best map variable-length f_0 contours to fixed-size feature representations for music classification applications.

As the main contribution of this paper, we systematically evaluate different contour feature representations for a wide range of MIR classification tasks. In particular, we compare hand-crafted features (knowledge-driven approach) with features learned from data (data-driven approach). To capture dependencies over time, various sequence modeling techniques such as recurrent neural networks (RNN) or auto-regressive models exist. In this paper, we will focus on CNN-based methods for two reasons: first, shift-invariance with respect to time is a useful property in our context and, second, convolution kernels allow for a better interpretability (in terms of filters).

As a further contribution, we discuss a fusion approach based on fixed-size segments (subcontours), which lead to better classification results than approaches based on variable-length contours. Python code and data to reproduce the classification results are published on an accompanying website[1].

2. RELATED WORK

One prominent application scenario for frequency contour analysis in MIR is to classify instrument playing techniques as part of automatic transcription algorithms. For example, Barbancho et al. [7], Abeßer et al. [8], and Kehling et al. [3] showed for isolated violin, bass guitar, and electric guitar recordings, respectively, that typical frequency modulation techniques such as vibrato, bending, and slides can be classified with high accuracy above 90 % on a note-level. As for ensemble recordings, the classification problem becomes much harder. For example, Abeßer et al. reported in [9] accuracy values between 48% (fall-off) to 92 % (vibrato) for common modulation techniques in trumpet and saxophone jazz solos. The authors proposed a set of contour features that measures modulation, fluctuation, and the average gradient of f_0 contours (see PYMUS feature set, Section 4.1). In a follow-up publication, these features were used to investigate how the pitch modulation range and the intonation depend on the musical context (within a solo) and on the artist [10].

In [11], Salamon et al. used the Melodia melody detection algorithm [12] to extract f_0 contours from polyphonic music recordings. Based on these f_0 contours, the authors described a set of low-dimensional features including contour duration, pitch range, as well as vibrato rate, extent, and coverage. These contour features outperformed low-level timbre features for genre classification. Pantelli and Bittner proposed a set of contour features (see BITTELI feature set, Section 4.1) for singing style analysis [13]. First, f_0 contours were classified according to vocal/non-vocal categories. Then, a dictionary-learning approach based on spherical k-means clustering was used to derive fixed-size activation histogram for vocal contours. Finally, these histograms were used as features to analyze different singing styles. Using the same feature set, Bittner et al. reported in [1] accuracy values around 0.72 for related tasks like vocal/non-vocal, bass/non-bass, melody/non-melody, and singer's gender (male, female) classification.

3. DATASETS & CLASSIFICATION SCENARIOS

In this paper, we use four datasets, which cover various music analysis tasks and different levels of timbre complexities. Table 1 provides a general overview over all datasets. We apply a post-processing (resampling) to have the same time resolution of 5.8

[1] https://github.com/dfg-isad/icassp_2019_f0_contours

Table 1: Dataset overview. Bold prints in column "Classes" indicate corresponding class abbreviations used in Figure 1. Final three columns indicate number of classes, contours (C), subcontours (SC), and files in each dataset.

Label	Task	Classes	Dataset	Complexity	Contour Estimator	Number of Classes	Number of C (SC)	Number of Files
GENRE	Music Genre	**flamenco**, **inst**rumental **jazz**, **opera**, **pop**, **vocal jazz**,	In-house dataset [2]	multitimbral	Melodia	5	12531 (487386)	499
GUITAR	Playing Style (guitar)	bending (**BE**), normal (**NO**), slide (**SL**), vibrato (**VI**)	IDMT-SMT-GUITAR [3]	monotimbral	Score + pYin	4	2240 (67728)	191
INST	Instrument	**cla**rinet, **flu**te, **sax**ophone, **s**inging **v**oice (**f**emale), **s**inging **v**oice (**m**ale), **tru**mpet, **vio**lin	IDMT-MONOTIMBRAL [4]	multitimbral	Melody Transcription + Peak Tracking	8	10214 (179743)	180
WJD	Playing Style (saxophone, trumpet etc.)	(pitch) **ben**d, **fal**l, **sli**de, **vib**rato	Weimar Jazz Database (WJD) [5]	monotimbral after source sep.	Score-Informed Separation + pYin [6]	4	4964 (126808)	360

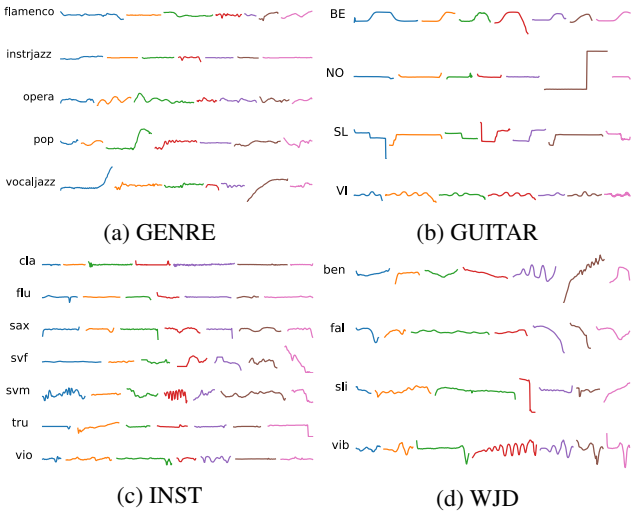
ms for all contours across all four datasets. As will be detailed in Section 4.2, we only consider contours from 197.2 ms to 1995.2 ms duration. The GENRE database was used in [11] and contains 12531 contours from 500 30-second excerpts equally distributed along the five music genres opera, pop, flamenco, vocal jazz, and instrumental jazz. The contours were extracted using the Melodia algorithm [12] covering a frequency range of five octaves between 55 Hz and 1760 Hz. The GUITAR dataset includes 2240 tones extracted from monotimbral electric guitar recordings in the IDMT-SMT-GUITAR dataset [3]. The notes are annotated with five the playing style classes bending, normal (stable pitch), slide, and vibrato. Again, the pYin pitch tracker was used for note-wise contour extraction. The INST dataset includes 10214 contours extracted from the IDMT-MONOTIMBRAL dataset [4]. Here, only the monophonic instrument classes violin, flute, trumpet, saxophone, clarinet, as well as female and male singing voice were considered. Contours were extracted by first running the automatic melody transcription algorithm by Dressler [14] followed a partial tracking based on linear interpolation as part of the solo/accompaniment source separation [15]. The WJD dataset includes a subset of 4964 tones taken from the Weimar Jazz Database (WJD) [5], which are annotated with one of the four playing style classes drop-off, slide, pitch-bend, and vibrato. The WJD includes jazz ensemble recordings with predominant solo instruments such as trumpet, tenor, alto, soprano saxophone, and trombone. Using the same procedure as described in [5], we first applied score-informed solo/accompaniment source separation [15] to extract the solo instrument, and then applied the pYin pitch tracking algorithm [6] to extract frequency contours for all notes.

Figure 1 shows seven randomly chosen example contours for each dataset and each class. For instance, characteristic contour shapes such as the periodic frequency modulation of vibrato tones can be recognized for the playing style classification tasks (GUITAR and WJD). For high-level classification tasks such as genre classification (GENRE) and instrument classification (INST), the classes tend to be less homogeneous in the sense that there are often several different contour shapes associated to a single class. Furthermore, some contours cover multiple playing techniques such as an initial pitch slide followed by a vibrato (see WJD examples).

4. FEATURE REPRESENTATIONS

4.1. Hand-Crafted Audio Features

In our experiments, we use two hand-crafted features sets. The first one is called BITTELI[2] and was introduced in [13]. From this feature



(a) GENRE (b) GUITAR (c) INST (d) WJD

Fig. 1: Randomly selected contours from each class of the four datasets introduced in Table 1.

set, we use 18 features including 6 features that capture the shape and coverage of vibrato, 8 features derived from a polynomial approximation of frequency contours, as well as 4 features derived from global statistics.

The second hand-crafted feature set is the PYMUS[3] set, which consists of 17 features including 3 features describing vibrato characteristics, 10 features measuring different contour shape properties related to fluctuation and gradient, as well as 4 features derived from a temporal contour segmentation. The two features sets contain different types of features and overlap only with regard to vibrato rate descriptors.

4.2. Feature Learning

Next, we introduce some feature sets that are automatically derived using a data-driven approach based on CNNs. In particular, as shown in Figure 3, we compare two neural network architectures with one (CNN-1) or two processing blocks (CNN-2) followed by two fully-connected (FC) layers. Each processing block includes a one-dimensional convolutional layer (CONV) followed by batch-normalization (BN) [16] and a rectified linear unit (ReLU), and a dropout (DO) [17] layer. In each convolution layer, we empirically selected 30 filters, each filter having a size of 10 (corresponding to

[2] https://github.com/rabitt/motif

[3] https://github.com/jazzomat/python/tree/master/pymus

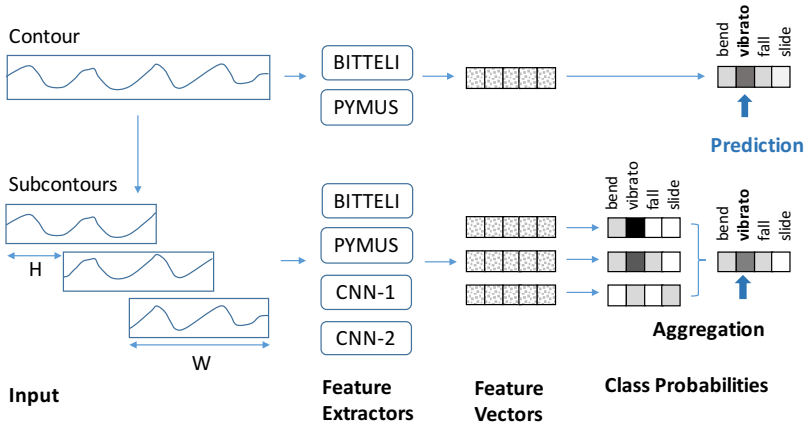


Fig. 2: Summary of the contour classification strategies. Variable-length contours can be processed solely by hand-crafted features (BITTELI, PYMUS). Fixed-size subcontours can be processed by all feature extractors.

Table 2: Features included in the hand-crafted feature sets BITTELI and PYMUS sorted by three feature categories (italic print). Salience-based features are discarded.

BITTELI	PYMUS
Vibrato features	
Rate (1) Extent (1) Coverage features (4)	Modulation frequency (1) Modulation dominance (1) Number of modulation periods (1)
Contour shape features	
Polynomial-fit on frequency contour (8) ~~Polynomial-fit on salience contour (7)~~	Measures for intonation & fluctuation (6) Frequency gradient descriptors (4) Contour segmentation features (4)
Global statistics	
Duration (1) Pitch (3) ~~Salience (3)~~	
18 (total)	**17 (total)**

a duration of 58 ms). We used the Adam optimizer, a learning rate of 10^{-3}, and a batch-size of 256. Optimizing the hyperparameters is not within the scope of this paper.

Each contour is assigned to one class (compare Table 1). These classes are used as target for the final softmax layer to train the models in a supervised fashion. After training, the activations of the penultimate fully-connected layer are used as features. For training the networks, we extract fixed-size subcontours from the variable length contours as input data for the CNN model using a window size W and a hopsize of 50 %. The class label of each contour is transferred to its subcontours. Since a common range for the vibrato rate is between 5 and 12 Hz [5], we use $W = 34$ (corresponding to 197.2 ms) as window size to capture at least one full vibrato period. In our classification experiment described in Section 5, we include all original contours that have a minimum length of 34 frames (197.2 ms). As shown in Figure 2, the predicted class probabilities on a subcontour level are aggregated by averaging to obtain contour-level predictions.

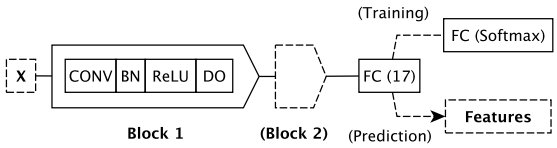


Fig. 3: Neural network architecture used for automatic contour feature learning (see Section 4.2).

5. EVALUATION

The four feature sets have comparable numbers of feature dimensions: PYMUS (17), BITTELI (18), CNN-1 (17), and CNN-2 (17). We perform a three-fold cross validation. In each fold, we repeat the following procedure for each of the classification tasks: The current dataset is randomly split into training and test set (80 % : 20 %) based on unique file assignments to avoid that f_0 contours from the same recording end up in both sets. Using the subcontours extracted from the training set as input and their class labels as targets, we train the CNN-1 and CNN-2 models. Afterwards, we extract feature vectors from the training set using all four feature extractors, normalize them to zero mean and unit variance, and train four independent random forest classifier models [18] with 50 trees each. Finally, we evaluate the classification performance on the test set by applying feature scaling using mean and standard variance values derived from the training set and computing the F1 score of the model predictions. The random forest classifier was chosen as it easily allows us to further analyze the importance of different feature dimensions for the trained models (compare Section 6.2). As illustrated in Figure 2, we compare both subcontour-level and contour-level classification for the two hand-crafted feature sets BITTELI and PYMUS. In addition, for the GENRE and INST dataset, we investigate a file-level aggregation strategy by averaging over all contour-level class probabilities.

Table 3 shows the mean F1 scores obtained from the cross-validation folds for all combinations of classification-level, dataset, and aggregation strategy. As general findings, we observe very similar scores for both hand-crafted features and learned features

Table 3: Mean F1 scores from 3-fold cross-validation. Results are shown for different feature extractors, datasets, classification levels (C = contour, SC = subcontour), and result aggregation level (contour-level, file-level). Best results for each dataset are highlighted in bold print.

Extractor	Aggr. Dataset	Contour-Level GENRE	GUITAR	INST	WJD	File-Level GENRE	INST
BITTELI	C	0.51	0.97	0.38	**0.87**	0.73	0.56
	SC	0.54	0.96	**0.43**	0.82	0.76	0.54
PYMUS	C	0.53	**0.98**	0.35	**0.87**	0.79	0.52
	SC	0.55	0.97	0.31	0.83	0.85	0.45
CNN-1	SC	0.54	0.95	0.34	0.83	0.85	0.49
CNN-2	SC	**0.63**	0.96	**0.43**	0.84	**0.94**	**0.67**

as well as for subcontour and contour classification. Throughout all feature extractors, aggregating the contour-level classification results in file-level results clearly boosts the F1 scores by up to 0.24.

The highest scores are achieved for the two playing-style classification datasets GUITAR and WJD. While all four models perform comparably well on the easier-to-classify GUITAR dataset, the hand-crafted features perform better on the WJD dataset. For the more difficult classification tasks based on the INST and GENRE datasets, the two-level CNN model (CNN-2) clearly outperforms its simpler counterpart (CNN-1) and the two hand-crafted features. Presumably, the CNN-2 model can learn to recognize more complex contour shapes.

6. CNN MODEL INSPECTION

6.1. Prototype Contours

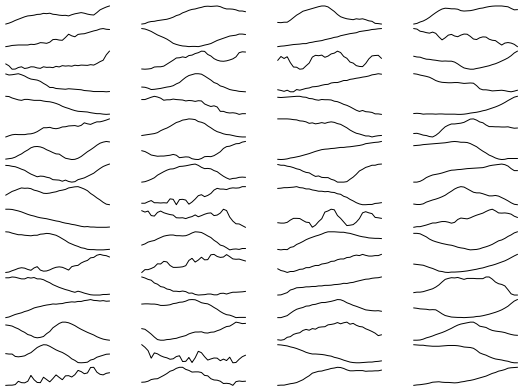
In the following, we aim to get a better insight into which contour shapes the CNN-based features capture. As an example, we investigate the two-layer CNN-2 model. We use the activation maximization algorithm [19][4] to generate frequency contours that maximize the activations of each of the 17 neurons in the penultimate dense layer, i. e., the individual feature dimension values. Figure 4 shows such contours for each of the four datasets. Despite the different underlying classification tasks, the networks learn to recognize similar contour shapes—increasing and decreasing frequency contours (pitch slides), alternating sequences of increasing and decreasing contour parts (pitch bends) as well as modulating (vibrato-like) shapes.

6.2. Feature Set Redundancy

The contours shown in Figure 4 indicate a certain redundancy as similar shapes can be found across neurons. In order to measure the information redundancy within different feature sets, we first compute all pair-wise correlation coefficients between features of the same set. Here we only focus on features extracted from subcontours. We observe significantly higher mean correlation values of 0.451 for the learned feature sets than for the hand-crafted feature sets (0.189).

Additionally, we analyze the feature importances, which measure their effect in the Random Forest models. Low information redundancy leads to only a few features having high importance values whereas high redundancy would result in an almost equal distribution. We compute the entropy H to measure the uniformity of the distribution over the feature importance values. We observe higher entropy values for the feature learning configurations of 0.960 than for the configurations using hand-crafted features (0.925). Both results indicate that discriminative information is more concentrated in the hand-crafted features, where only a subset of the features have a high effect in the classifier models. In contrast, the learned features show a higher information redundancy across feature dimensions.

[4]We used the keras-vis implementation of the activation maximization algorithm [20] with hand-tuned parameter values tv_weight $= 0.01$ and lp_norm_weight $= 0.01$.



(a) GENRE (b) GUITAR (c) INST (d) WJD

Fig. 4: Normalized frequency contours that maximize neuron activations in penultimate dense layer, which are used as features vectors.

7. CONCLUSION

This paper compares hand-crafted features and automatically learned features for different f_0 contour classification tasks. Our findings show that embedding features from a simple non-optimized neural network architecture with two convolutional layers can outperform hand-crafted features based on expert knowledge. Multiple convolutional layers allow for learning more complex contour shapes, which is beneficial especially for higher-level tasks such as genre and instrument recognition. The evaluation results show that using fixed-length subcontours in combination with an aggregation strategy leads to comparable classification accuracies as compared to an approach using global contour. As an advantage, classifying subcontours allows for a more complex (time-dependent) description of playing techniques on a note-level (e. g., initial pitch bend followed by a vibrato). Future work should address the use of convolutional recurrent neural network architectures that allow to classify input sequences of arbitrary length. A close investigation of the CNN-based features revealed that the learned feature sets have a higher redundancy across feature dimensions than the hand-crafted features. Reducing the number of filters and the amount of redundancy while maintaining the classification performance could be a guideline for model size optimization.

8. ACKNOWLEDGEMENTS

This work has been supported by the German Research Foundation (AB 675/2-1, MU 2686/11-1). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institut für Integrierte Schaltungen IIS. The authors would like to thank Justin Salamon, Rachel Bittner, and Maria Pantelli for valuable discussions and for sharing their contour datasets.

9. REFERENCES

[1] Rachel M. Bittner, Justin Salamon, Juan J. Bosch, and Juan P. Bello, "Pitch contours as a mid-level representation for music informatics," in *AES International Conference on Semantic Audio*, Erlangen, Germany, 22/06/2017 2017.

[2] Justin Salamon, Geoffroy Peeters, and Axel Röbel, "Statistical characterisation of melodic pitch contours and its application for melody extraction," in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, Porto, Portugal, 2012, pp. 187–192.

[3] Christian Kehling, Jakob Abeßer, Christian Dittmar, and Gerald Schuller, "Automatic Tablature Transcription of Electric Guitar Recordings by Estimation of Score- and Instrument-related Parameters," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Erlangen, Germany, September 2014.

[4] Juan Gómez, Jakob Abeßer, and Estefanía Cano, "Jazz solo instrument classification with convolutional neural networks, source separation, and transfer learning," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018.

[5] Martin Pfleiderer, Klaus Frieler, Jakob Abeßer, Wolf-Georg Zaddach, and Benjamin Burkhart, Eds., *Inside the Jazzomat - New Perspectives for Jazz Research*, Schott Campus, 2017.

[6] Matthias Mauch and Simon Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014, pp. 659–663.

[7] Isabel Barbancho, Christina de la Bandera, Ana M. Barbancho, and Lorenzo J. Tardon, "Transcription and expressiveness detection system for violin music," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 189–192.

[8] Jakob Abeßer, Hanna Lukashevich, and Gerald Schuller, "Feature-based Extraction of Plucking and Expression Styles of the Electric Bass Guitar," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, USA, 2010, pp. 2290–2293.

[9] Jakob Abeßer, Estefanía Cano, Klaus Frieler, Martin Pfleiderer, and Wolf-Georg Zaddach, "Score-informed analysis of intonation and pitch modulation in jazz solos," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, Málaga, Spain, 2015, pp. 823–829.

[10] Jakob Abeßer, Klaus Frieler, Estefanía Cano, Martin Pfleiderer, and Wolf-Georg Zaddach, "Score-informed analysis of tuning, intonation, pitch modulation, and dynamics in jazz solos," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 168–177, Jan 2017.

[11] Justin Salamon, Bruno Rocha, and Emilia Gómez, "Musical genre classification using melody features extracted from polyphonic music signals," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, 2012.

[12] Justin Salamon and Emilia Gómez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1759–1770, 2012.

[13] Maria Panteli, Rachel M. Bittner, Juan Pablo Bello, and Simon Dixon, "Towards the characterization of singing styles in world music," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, 2017, pp. 636–640.

[14] Karin Dressler, *Automatic transcription of the melody from polyphonic music*, Ph.D. thesis, TU Ilmenau, Germany, Jul 2017.

[15] Estefanía Cano, Gerald Schuller, and Christian Dittmar, "Pitch-informed solo and accompaniment separation: towards its use in music education applications," *EURASIP Journal on Advances in Signal Processing*, pp. 1–19, 2014.

[16] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on International Conference on Machine Learning (ICML)*, 2015, pp. 448–456.

[17] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[18] Leo Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001.

[19] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent, "Visualizing higher-layer features of a deep network," Tech. Rep. 1341, University of Montreal, June 2009.

[20] Raghavendra Kotikalapudi and contributors, "keras-vis," `https://github.com/raghakot/keras-vis`, 2017.