Automated classification of syllable types in vocal sequences of pups of the greater sac-winged bat Saccopteryx bilineata

Simon Hiller

Andreas Fischer & Daniel Wegmann
University of Bern - Fribourg

02. March 2021

Presentation Overview

- Introduction
- Methods
- Results & discussion
- Conclusion & outlook
- Questions

Human language capacity

- Decades of research already revealed many processes involved in human speech
  - The full complexity of the human language acquisition is still to be understood
  - The same is true for the evolution of languag
- Both open topics are investigated in biolinguistics science

Biolinguistic

- It combines: neurogenetics, animal behaviour, linguistics, developmental psychology, bioinformatics and mathematics
- The complex system of language is composed of:
  - key factors: syntax, semantics, vocal imitation
  - cognitive abilities: social learning, theory of mind, recursive processing
- Non-human animal vocal communication can be complex: vocal repertoire size, syntactical rules and vocal versatility
- Non-human animals possess multiple cognitive skills such as associative learning, vocal imitation and joint attention

Saccopterix bilineata

- The neotropical grater sac-winged bat Saccopteryx bilineata is vocal production learner & possesses a large vocal repertoire
- Pups acquire a part of the adult vocal repertoire through vocal imitation
- The precursors of the territorial song syllables gradually converge towards the territorial song of tutor males
- This is irrespective of relatedness and pup sex
- To study the underlying mechanisms crucial for vocal imitation could serve to unravel shared mechanisms and key factors across mammalian vocal learners including humans.

Saccopterix bilineata

Figure: Greater sac-winged bats (*Saccopteryx bilineata*) roosting in their day roost, the young one on the right is vocalising. The smaller bats with the dark fur are juveniles and the others with lighter fur are their mothers. (© Michael Stifter)

Reason

- Time-consuming, yet crucial manually labelling of syllables sequences in audio files
- Manually classify them based on spectral similarity to the syllables from the adult vocal repertoire
- Interest in deep learning models

Related works - based on machine learning

- Many vocalisation studies exists for species classification, monitoring the health condition of the environment or the animal
- But, there seems to exists no work addressing automatic classification of syllable types for any bat species
- The closest related works are:
  - "Bat detective—Deep learning tools for bat acoustic signal detection" by Mac Aodha et al.
  - "Automatic standardized processing and identification of tropical bat calls using deep learning approache" by Chen et al.
- "Bird Voice Deep Learning" by Gilles Waeber, providing a framework with:
  - feature preprocessing similar to our requirements
  - training and testing of deep learning models

How

- Approach: supervised deep learning
- Model: deep learning model with pattern learning abilities.
  - Input: spectrogram or slit-style HOG
  - Output: corresponding syllable type

K-fold cross-validation

- Measuring the extent to which a model generalize by comparing the predictions from the unseen data with the true output
- Preferred validation process when the dataset has limited samples
- We used for all experiments $k = 8$ width one validation bin and two testing bins.

	Test	Validation	Training	
Experiment 1	1	2	3	4
Experiment 2	2	3	4	1
Experiment 3	3	4	1	2
Experiment 4	4	1	2	3

Figure: A visualisation of the distribution of bins in a k-fold CV with $k = 4$, using one bin each for validation and testing.

Pipeline

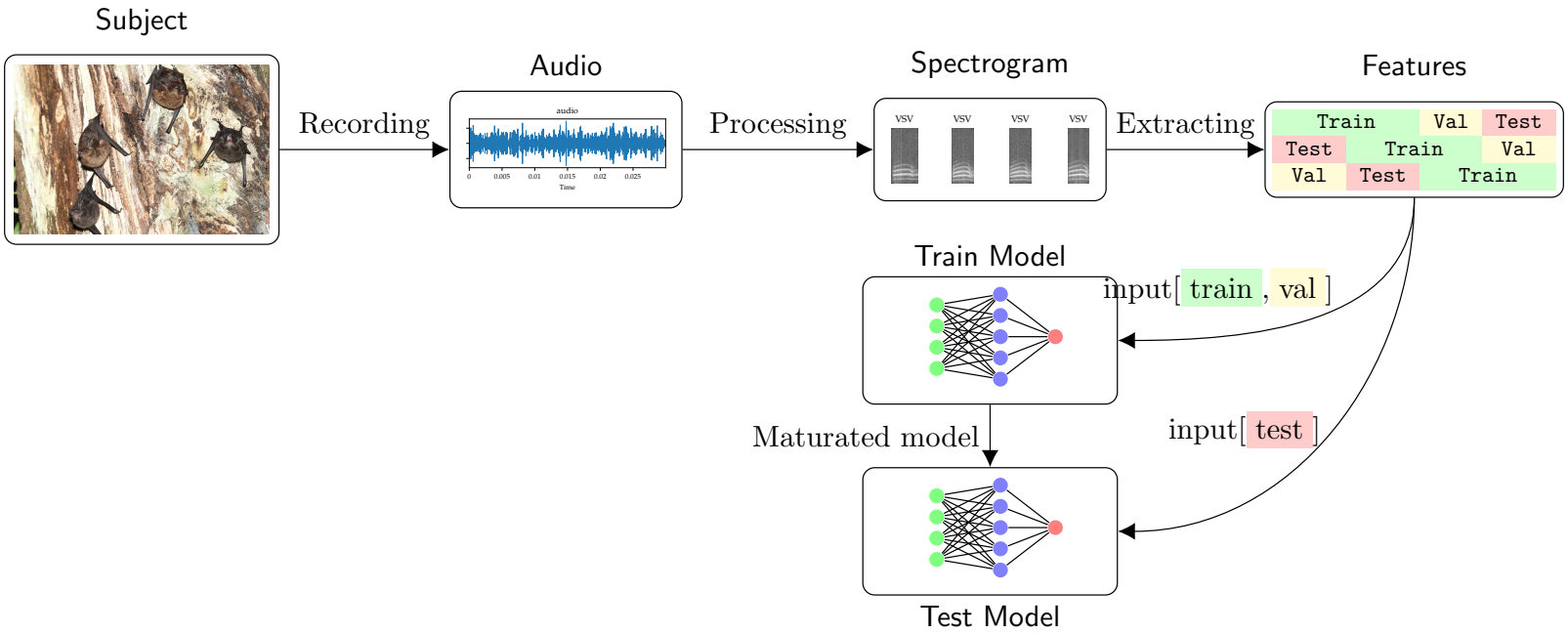


Figure: Flow diagram of the data.

Preprocessing overview

- Filter out too short syllables and syllable types with not enough samples
- Generate silent audio files per recording and create noise profiles with them
- Extract audio parts which belongs to a syllable by the defined start and end time point
- Left pad the extracted syllables to the same maximum length over all with the signal data from the background audio file
- Apply noise reduction filter to the audio files based on the generated noise profiles
- Finally generate spectrogram or HOG features

Preprocessing - silent segmentation

$$silent = \begin{cases} 1 & mse < mean(mse \text{ without label}) + 1, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

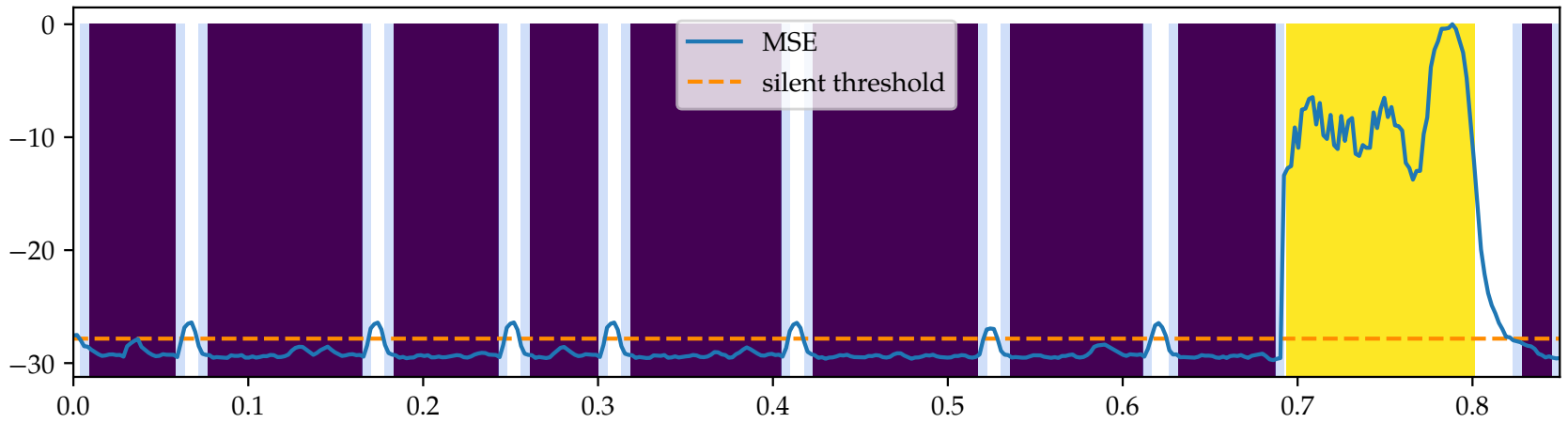


Figure: Plot of the result of the silence detection algorithm. The blackcurrant rectangles are the silent parts, on both sides the piece of 5 ms by which the silent parts were reduced is shown in transparent light blue. The yellow rectangle is a labelling part.

Multilayer perceptron

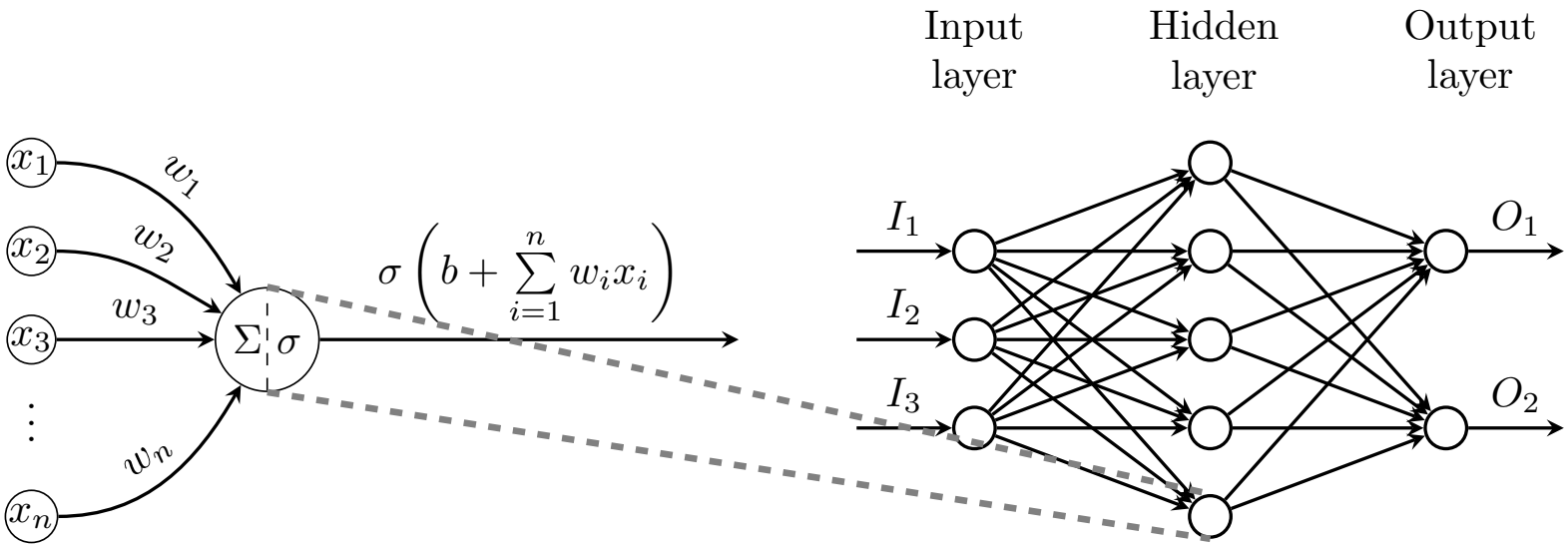
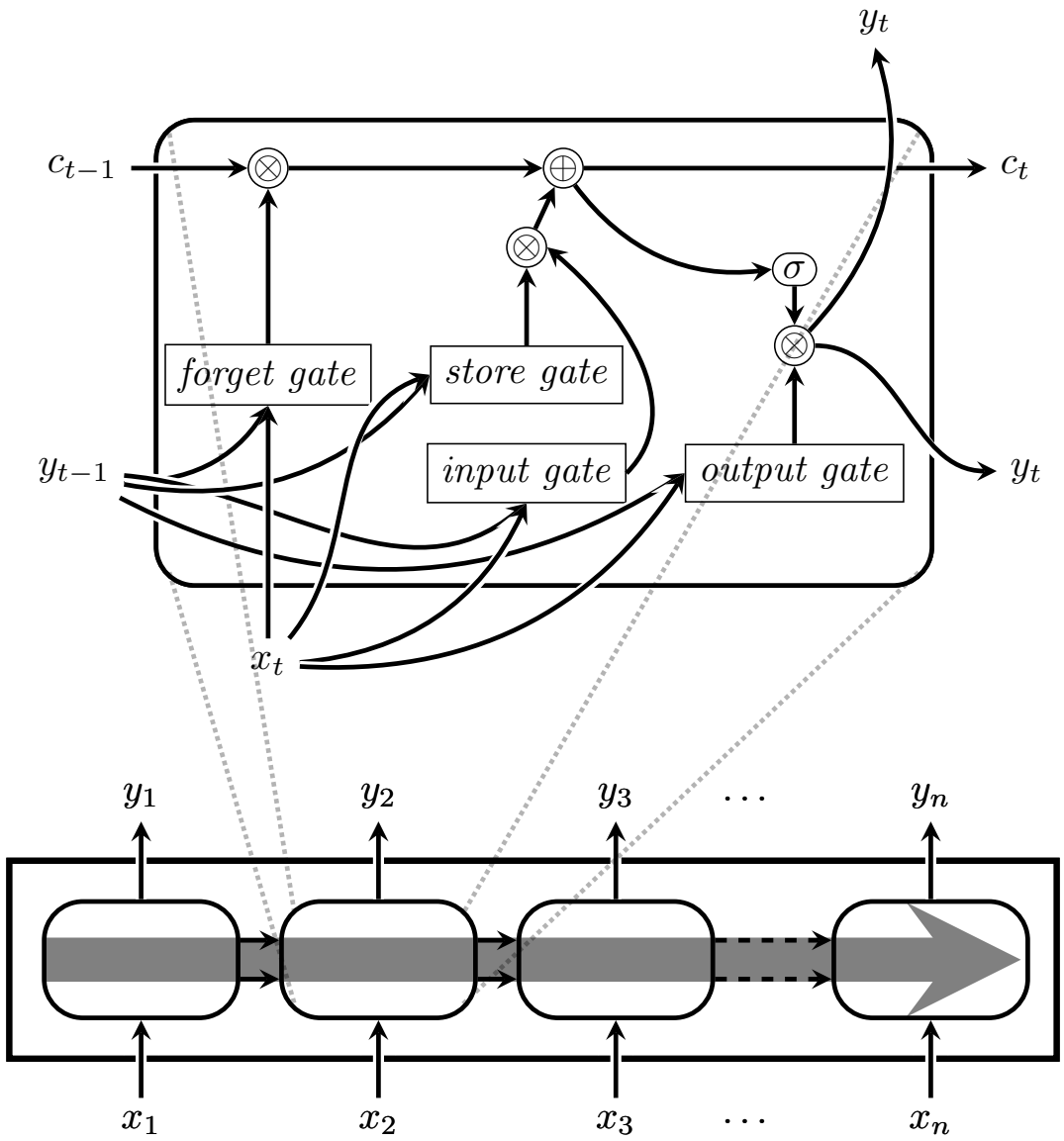


Figure: A diagram of a single perceptron, along with its position within a multilayer perceptron with fully connected layers.

Long short term memory (LSTM)



LSTM model

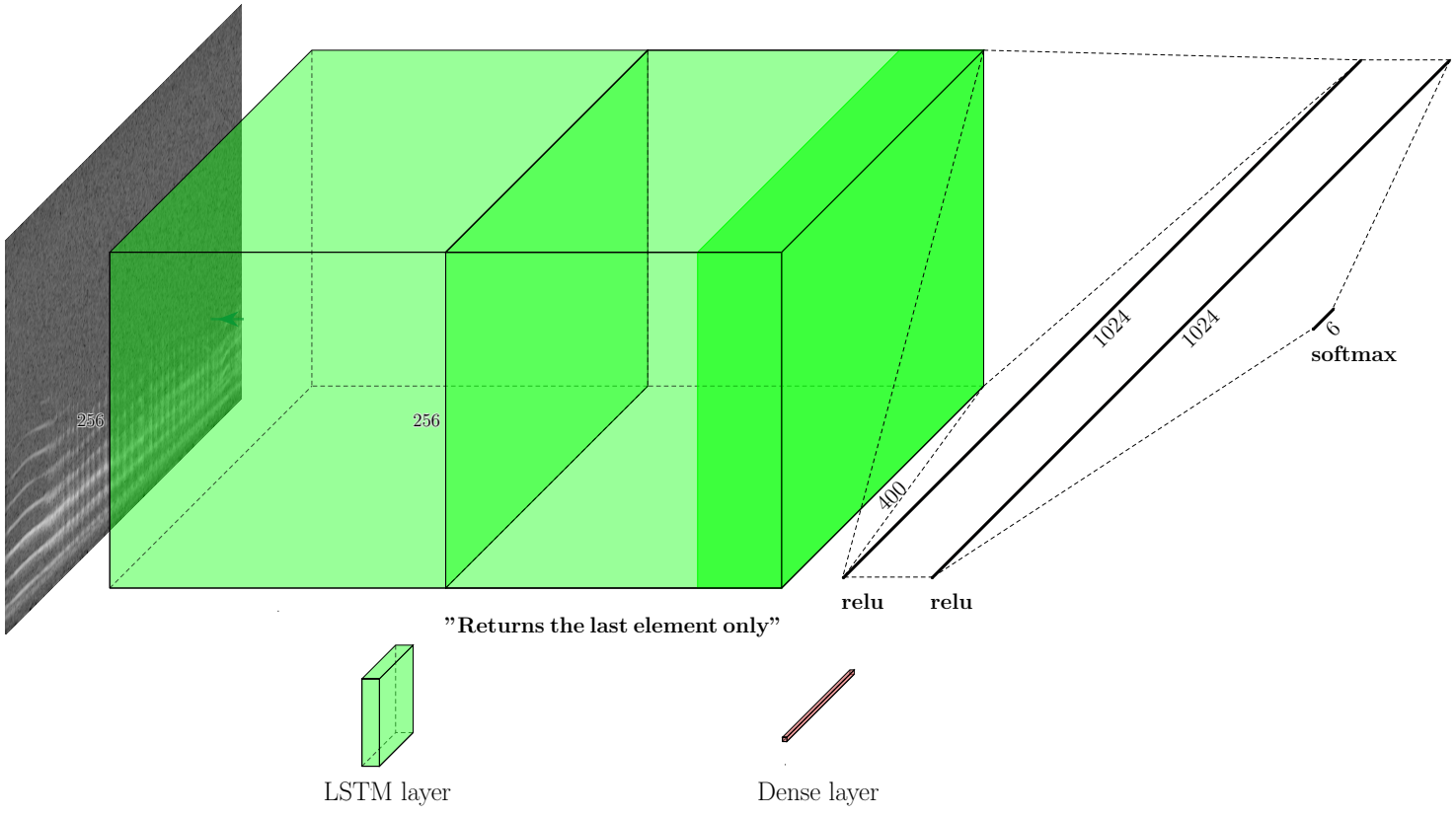


Figure: Visualisation of the LSTM architecture. The size of the elements does not correspond exactly to their real dimension numbers.

Convolution layer

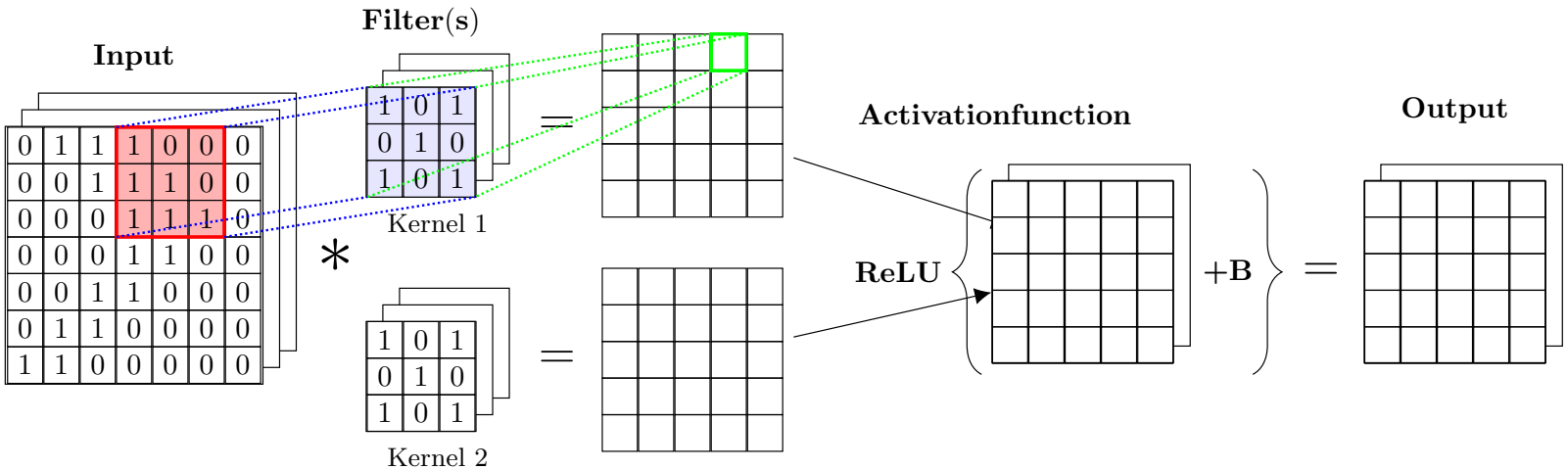


Figure: A diagram of a single convolution layer depicts the fate of the features through them.

Convolution neural network model

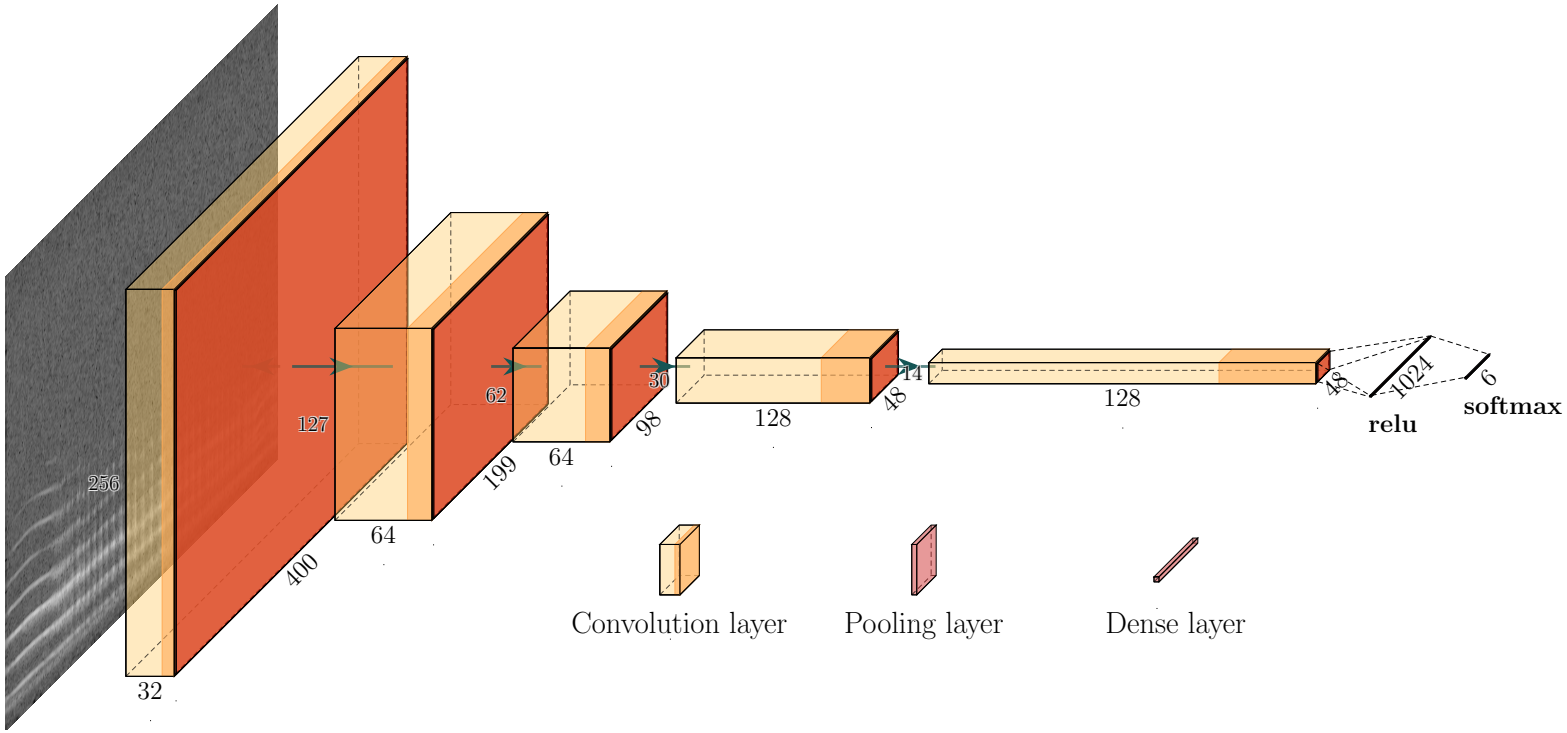


Figure: Visualisation of the CNN architecture. The size of the elements do not correspond exactly to their real dimension numbers.

DenseNet Model

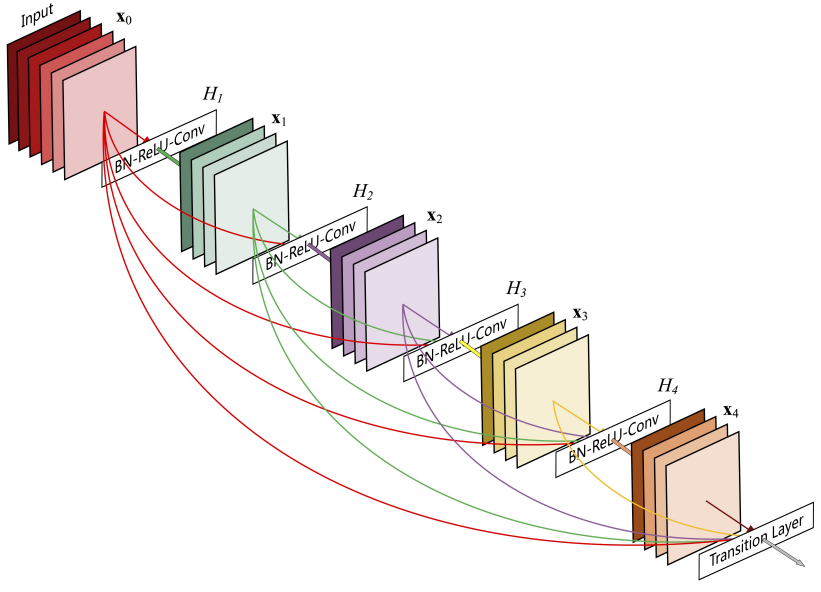


Figure: Densely connected convolutional network block with 4 weight layers and a growth rate of 4. (© Huang et al.)

Test experiments

- We conducted 3 test experiments:
  - compressed
  - variable length
  - left padded

Test experiments - compressed

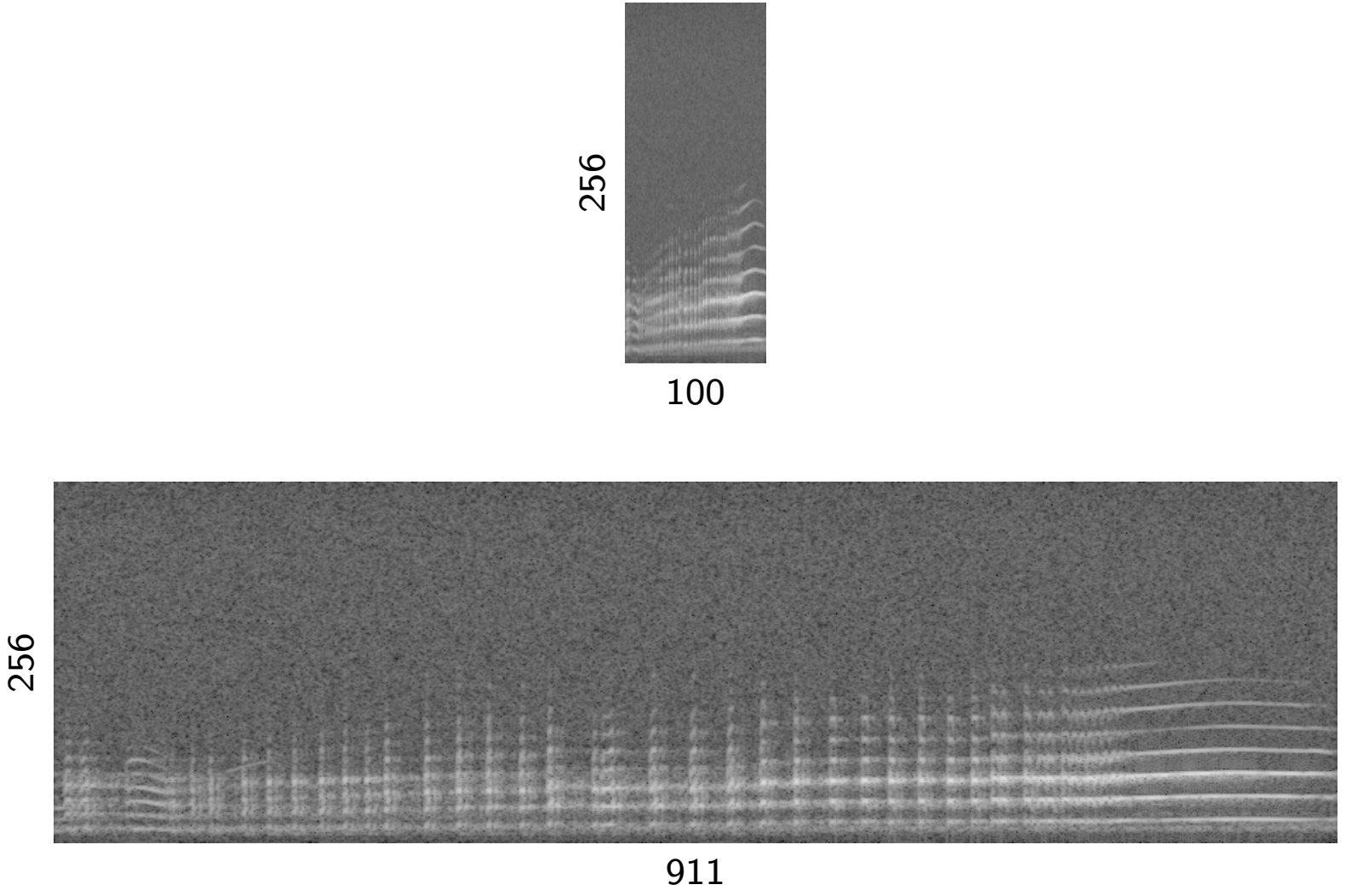


Figure: Visualisation of a compression ratio of 1:9 on the basis of the longest syllable sample.

Compressed set sample distribution

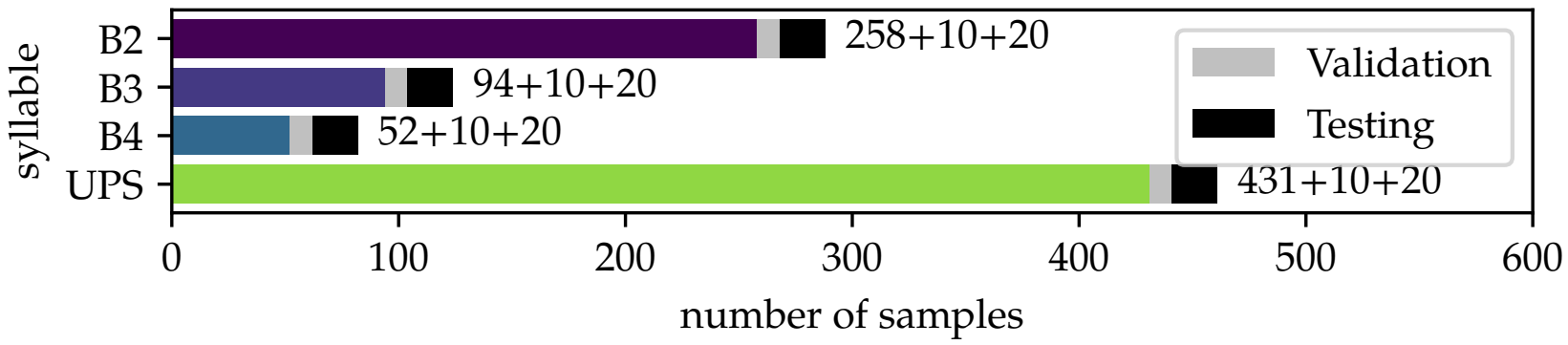
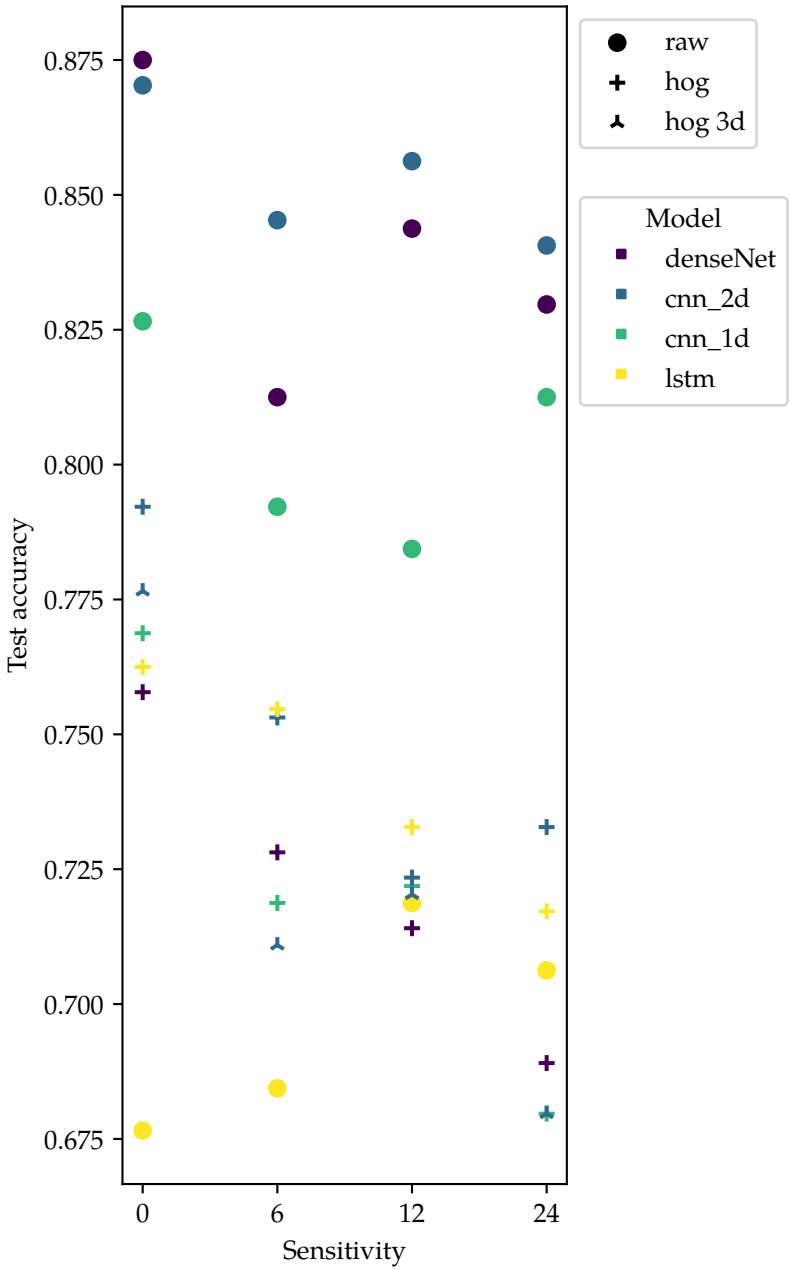


Figure: The distribution of the samples used for the compressed test experiment. For all syllables we used 10 samples for validation and 20 for testing. The dataset is not balanced with a ratio of about 1 : 8 in the number of training samples between the least and most represented syllable type. Only around 17.35% of the most represented syllable type samples are rotated in the k-fold process.

Test experiments - compressed



Model	Test Accuracy (%)	Loss
DenseNet, nrs: 0, raw	87.5 ± 2.8	0.65 ± 0.29
CNN 2D, nrs: 0, raw	87.0 ± 4.4	0.43 ± 0.18
CNN 2D, nrs: 12, raw	85.6 ± 6.0	0.41 ± 0.14
CNN 2D, nrs: 6, raw	84.5 ± 5.5	0.38 ± 0.14
DenseNet, nrs: 12, raw	84.4 ± 6.5	0.72 ± 0.42
CNN 2D, nrs: 24, raw	84.1 ± 5.7	0.46 ± 0.20
DenseNet, nrs: 24, raw	83.0 ± 4.5	0.71 ± 0.30
CNN 1D, nrs: 0, raw	82.7 ± 4.7	0.49 ± 0.08
DenseNet, nrs: 6, raw	81.2 ± 5.6	0.66 ± 0.31
CNN 1D, nrs: 24, raw	81.2 ± 6.1	0.63 ± 0.27
CNN 1D, nrs: 6, raw	79.2 ± 2.7	0.61 ± 0.11
CNN 2D, nrs: 0, HOG	79.2 ± 4.6	0.56 ± 0.14
CNN 1D, nrs: 12, raw	78.4 ± 4.3	0.68 ± 0.15
CNN 2D, nrs: 0, HOG 3D	77.7 ± 3.7	0.54 ± 0.05
CNN 1D, nrs: 0, HOG	76.9 ± 5.1	0.57 ± 0.06
LSTM, nrs: 0, HOG	76.2 ± 2.6	0.82 ± 0.34
DenseNet, nrs: 0, HOG	75.8 ± 4.4	1.20 ± 0.21
LSTM, nrs: 6, HOG	75.5 ± 4.9	0.84 ± 0.23
CNN 2D, nrs: 6, HOG	75.3 ± 6.0	0.81 ± 0.30
LSTM, nrs: 12, HOG	73.3 ± 6.0	0.94 ± 0.37
CNN 2D, nrs: 24, HOG	73.3 ± 9.7	0.76 ± 0.22
DenseNet, nrs: 6, HOG	72.8 ± 6.2	1.63 ± 0.62
CNN 2D, nrs: 12, HOG	72.3 ± 8.0	0.75 ± 0.18
CNN 1D, nrs: 12, HOG	72.2 ± 9.8	0.76 ± 0.24
CNN 2D, nrs: 12, HOG 3D	72.0 ± 6.9	0.71 ± 0.16
CNN 1D, nrs: 6, HOG	71.9 ± 7.1	0.77 ± 0.17
LSTM, nrs: 12, raw	71.9 ± 7.2	1.40 ± 0.55
LSTM, nrs: 24, HOG	71.7 ± 8.2	0.98 ± 0.27
DenseNet, nrs: 12, HOG	71.4 ± 4.5	1.90 ± 0.38
CNN 2D, nrs: 6, HOG 3D	71.1 ± 6.9	0.74 ± 0.18
LSTM, nrs: 24, raw	70.6 ± 5.6	1.22 ± 0.41
DenseNet, nrs: 24, HOG	68.9 ± 4.9	1.77 ± 0.28
LSTM, nrs: 6, raw	68.4 ± 6.9	1.26 ± 0.63
CNN 2D, nrs: 24, HOG 3D	68.0 ± 10.2	0.79 ± 0.22
CNN 1D, nrs: 24, HOG	68.0 ± 8.8	0.89 ± 0.24
LSTM, nrs: 0, raw	67.7 ± 2.3	1.23 ± 0.52

Dataset sample distribution

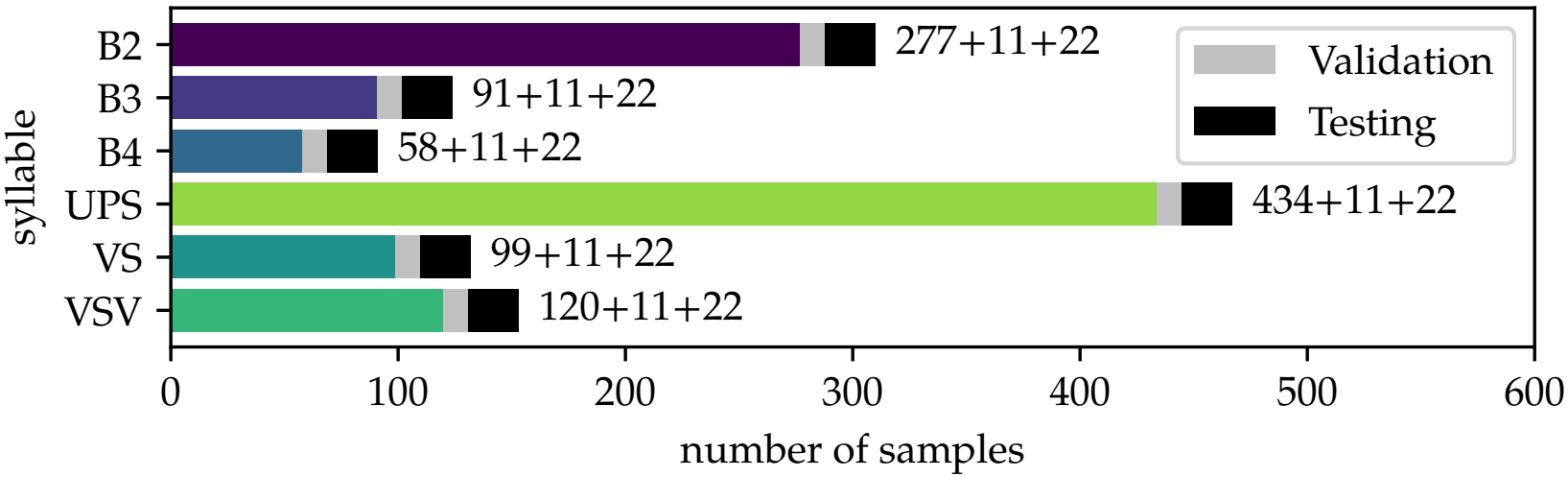


Figure: The distribution of the syllables from the simple call data set. Over all syllables we used 11 samples for validation and 22 for testing. The dataset is not balanced in the number of training samples, with a ratio of about 1 : 7 between the least and most represented syllable. Only around 18.84% of the most represented syllable type samples are rotated in the k-fold process.

Test experiments - variable length

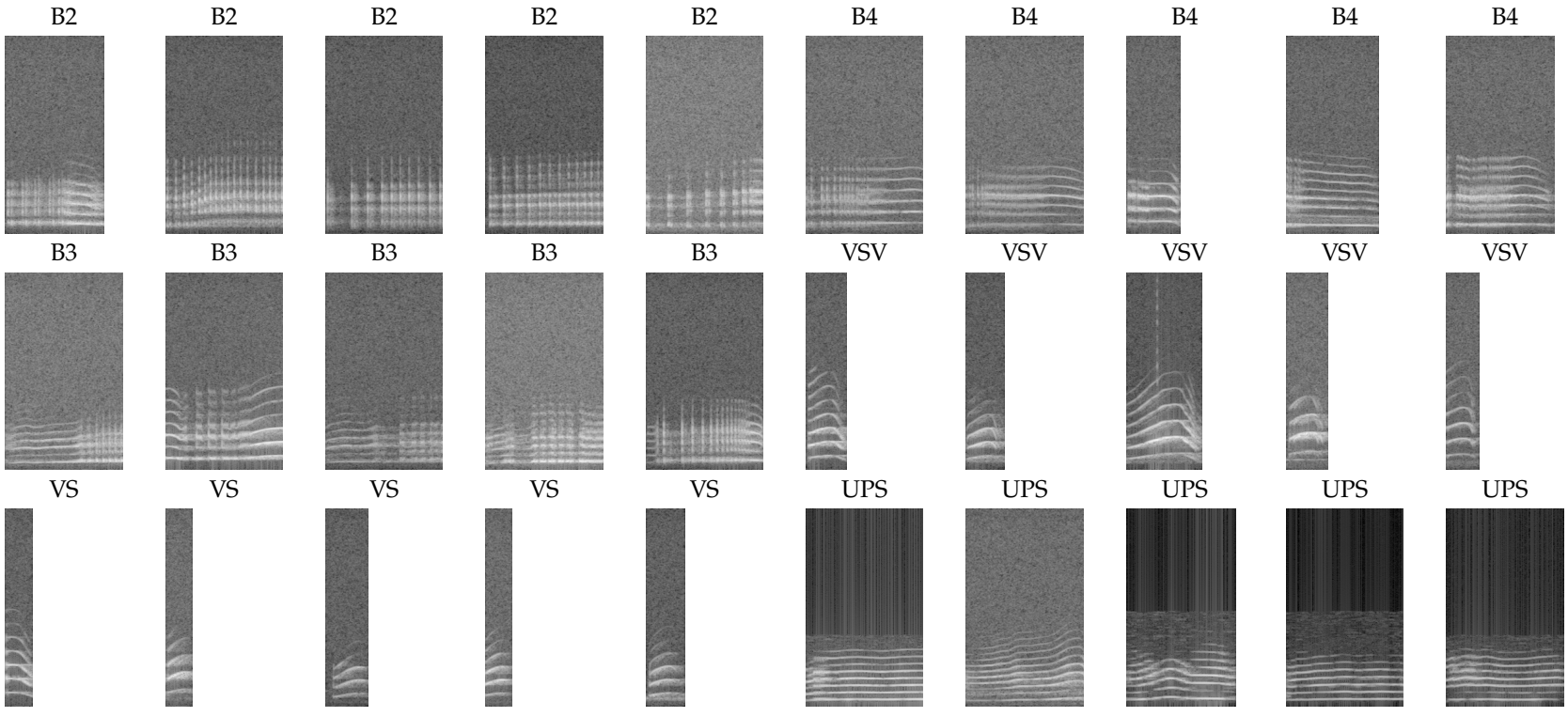
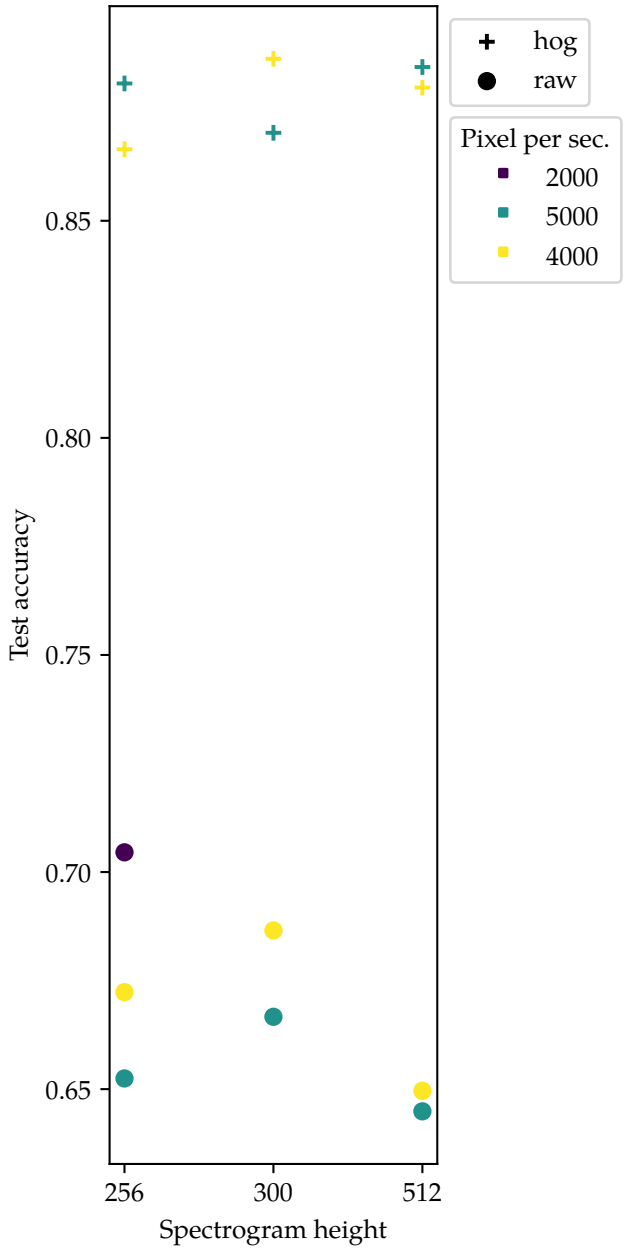


Figure: Feature preview, for every syllable type are 5 samples randomly selected.

Test experiments - variable length



Model	Test Accuracy (%)	Loss
LSTM, xpps: 4K, height: 300, HOG	88.7 ± 2.4	0.44 ± 0.17
LSTM, xpps: 5K, height: 512, HOG	88.5 ± 1.8	0.40 ± 0.11
LSTM, xpps: 5K, height: 256, HOG	88.2 ± 1.7	0.42 ± 0.11
LSTM, xpps: 4K, height: 512, HOG	88.1 ± 3.4	0.42 ± 0.13
LSTM, xpps: 5K, height: 300, HOG	87.0 ± 2.4	0.46 ± 0.12
LSTM, xpps: 4K, height: 256, HOG	86.6 ± 2.9	0.52 ± 0.18
LSTM, xpps: 2K, height: 256, raw	70.5 ± 3.8	1.50 ± 0.53
LSTM, xpps: 4K, height: 300, raw	68.7 ± 1.9	1.68 ± 0.37
LSTM, xpps: 4K, height: 256, raw	67.2 ± 2.7	2.05 ± 0.22
LSTM, xpps: 5K, height: 300, raw	66.7 ± 1.6	1.76 ± 0.44
LSTM, xpps: 5K, height: 256, raw	65.2 ± 2.9	1.62 ± 0.45
LSTM, xpps: 4K, height: 512, raw	65.0 ± 3.7	1.82 ± 0.45
LSTM, xpps: 5K, height: 512, raw	64.5 ± 3.0	1.82 ± 0.76

Test experiments - padded

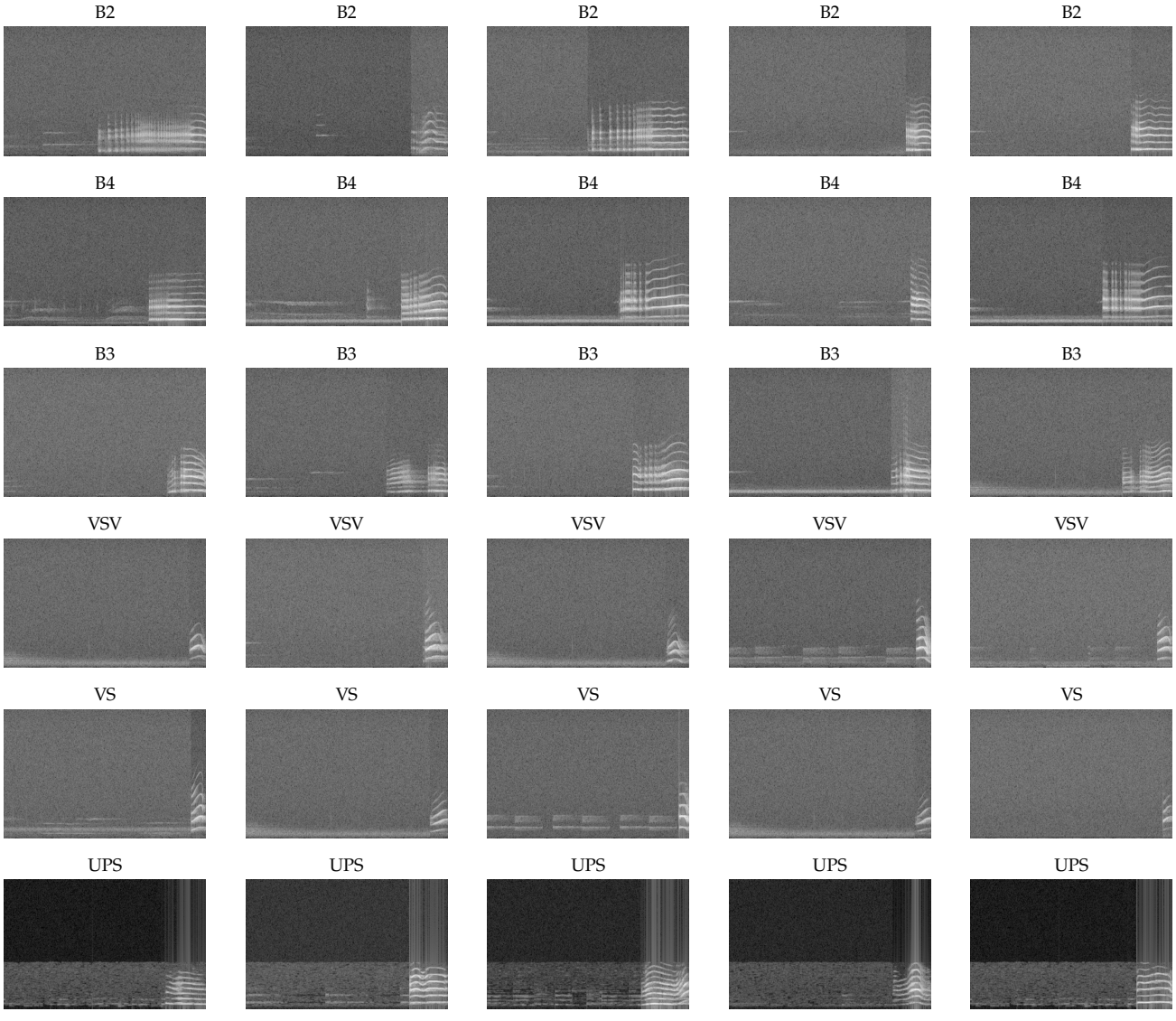
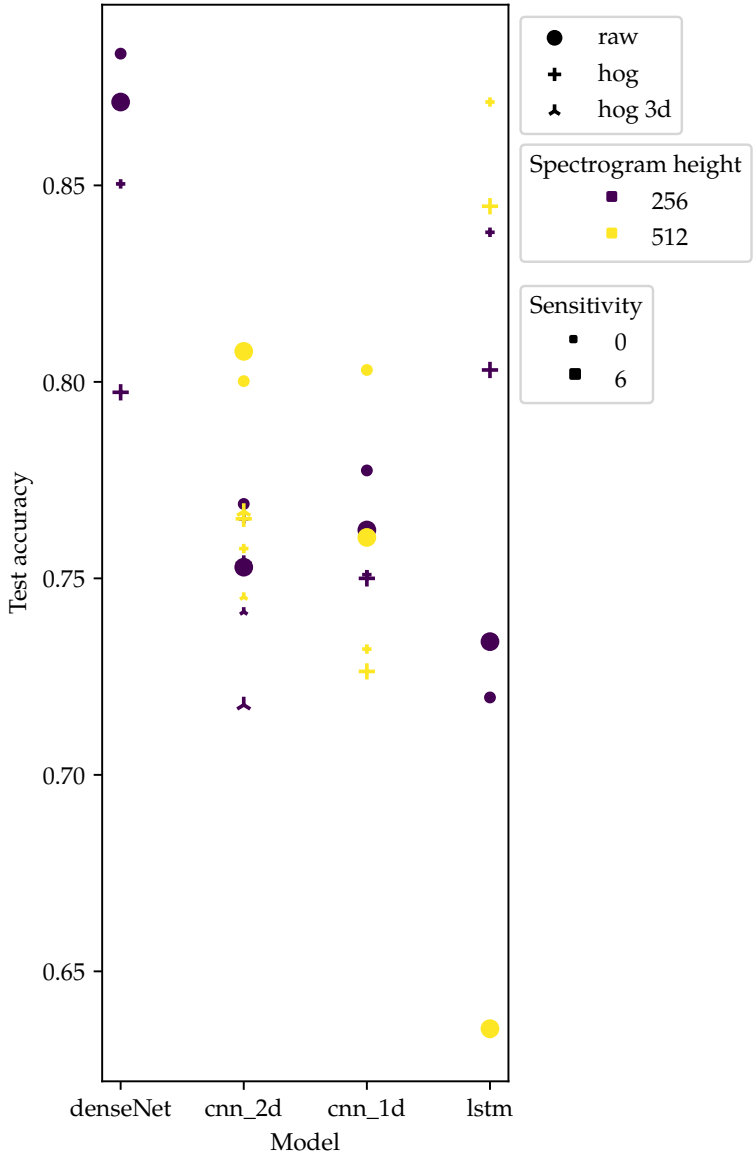


Figure: Feature preview, for every syllable type are 5 samples randomly selected.

Test experiments - padded



Model	Test Accuracy (%)	Loss
DenseNet, nrs: 0, xpps: 2K, height: 256, raw	88.4 ± 3.5	0.62 ± 0.31
LSTM, nrs: 0, xpps: 5K, height: 512, HOG	87.1 ± 2.1	0.45 ± 0.11
DenseNet, nrs: 6, xpps: 2K, height: 256, raw	87.1 ± 4.4	0.81 ± 0.43
DenseNet, nrs: 0, xpps: 2K, height: 256, HOG	85.0 ± 4.8	0.88 ± 0.41
LSTM, nrs: 6, xpps: 5K, height: 512, HOG	84.5 ± 4.6	0.59 ± 0.19
LSTM, nrs: 0, xpps: 2K, height: 256, HOG	83.8 ± 2.3	0.74 ± 0.19
CNN 2D, nrs: 6, xpps: 5K, height: 512, raw	80.8 ± 2.7	0.99 ± 0.44
LSTM, nrs: 6, xpps: 2K, height: 256, HOG	80.3 ± 3.9	0.72 ± 0.11
CNN 1D, nrs: 0, xpps: 5K, height: 512, raw	80.3 ± 3.7	0.73 ± 0.13
CNN 2D, nrs: 0, xpps: 5K, height: 512, raw	80.0 ± 3.2	1.06 ± 0.47
DenseNet, nrs: 6, xpps: 2K, height: 256, HOG	79.7 ± 3.4	0.99 ± 0.32
CNN 1D, nrs: 0, xpps: 2K, height: 256, raw	77.7 ± 5.4	0.63 ± 0.12
CNN 2D, nrs: 0, xpps: 2K, height: 256, raw	76.9 ± 5.4	0.89 ± 0.35
CNN 2D, nrs: 6, xpps: 5K, height: 512, HOG 3D	76.7 ± 4.2	1.01 ± 0.34
CNN 2D, nrs: 0, xpps: 2K, height: 256, HOG	76.5 ± 2.8	0.89 ± 0.31
CNN 2D, nrs: 6, xpps: 5K, height: 512, HOG	76.5 ± 3.3	0.95 ± 0.27
CNN 1D, nrs: 6, xpps: 2K, height: 256, raw	76.2 ± 3.8	0.71 ± 0.19
CNN 1D, nrs: 6, xpps: 5K, height: 512, raw	76.0 ± 3.2	0.80 ± 0.17
CNN 2D, nrs: 0, xpps: 5K, height: 512, HOG	75.8 ± 4.1	1.05 ± 0.37
CNN 2D, nrs: 6, xpps: 2K, height: 256, HOG	75.4 ± 5.3	0.74 ± 0.25
CNN 2D, nrs: 6, xpps: 2K, height: 256, raw	75.3 ± 2.3	0.81 ± 0.20
CNN 1D, nrs: 0, xpps: 2K, height: 256, HOG	75.1 ± 3.3	0.59 ± 0.09
CNN 1D, nrs: 6, xpps: 2K, height: 256, HOG	75.0 ± 7.6	0.68 ± 0.14
CNN 2D, nrs: 0, xpps: 5K, height: 512, HOG 3D	74.5 ± 3.8	0.79 ± 0.15
CNN 2D, nrs: 0, xpps: 2K, height: 256, HOG 3D	74.1 ± 2.2	0.82 ± 0.20
LSTM, nrs: 6, xpps: 2K, height: 256, raw	73.4 ± 3.5	1.58 ± 0.40
CNN 1D, nrs: 0, xpps: 5K, height: 512, HOG	73.2 ± 5.6	0.77 ± 0.13
CNN 1D, nrs: 6, xpps: 5K, height: 512, HOG	72.6 ± 6.9	0.92 ± 0.17
LSTM, nrs: 0, xpps: 2K, height: 256, raw	72.0 ± 3.4	1.40 ± 0.51
CNN 2D, nrs: 6, xpps: 2K, height: 256, HOG 3D	71.8 ± 4.2	0.87 ± 0.15
LSTM, nrs: 6, xpps: 5K, height: 512, raw	63.5 ± 1.5	2.11 ± 0.53
LSTM, nrs: 0, xpps: 5K, height: 512, raw	63.4 ± 3.9	1.73 ± 0.69

Test experiments - compressed - learning curves

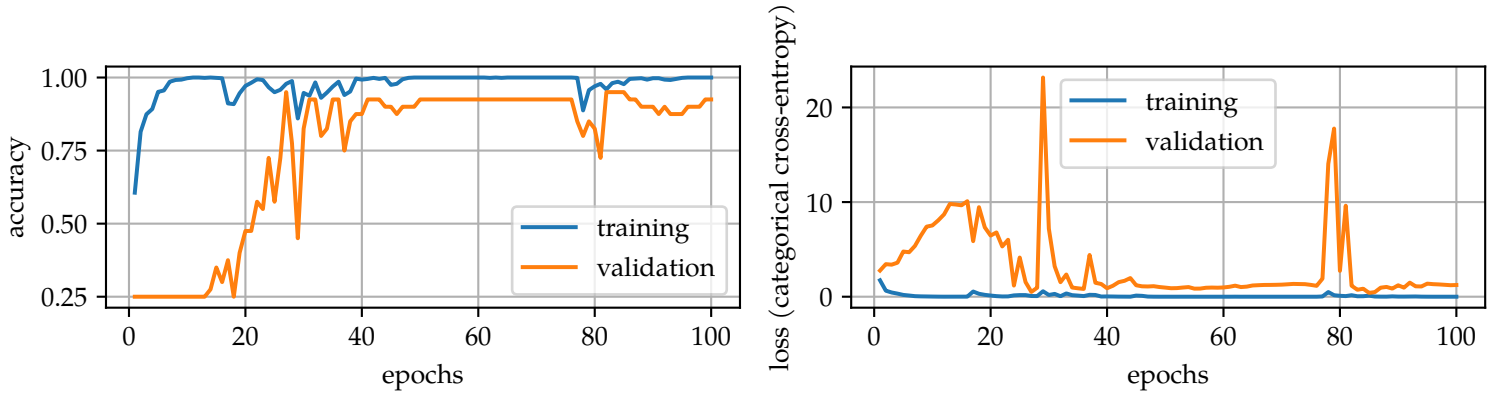


Figure: Training progression for the best DenseNet model on raw features.

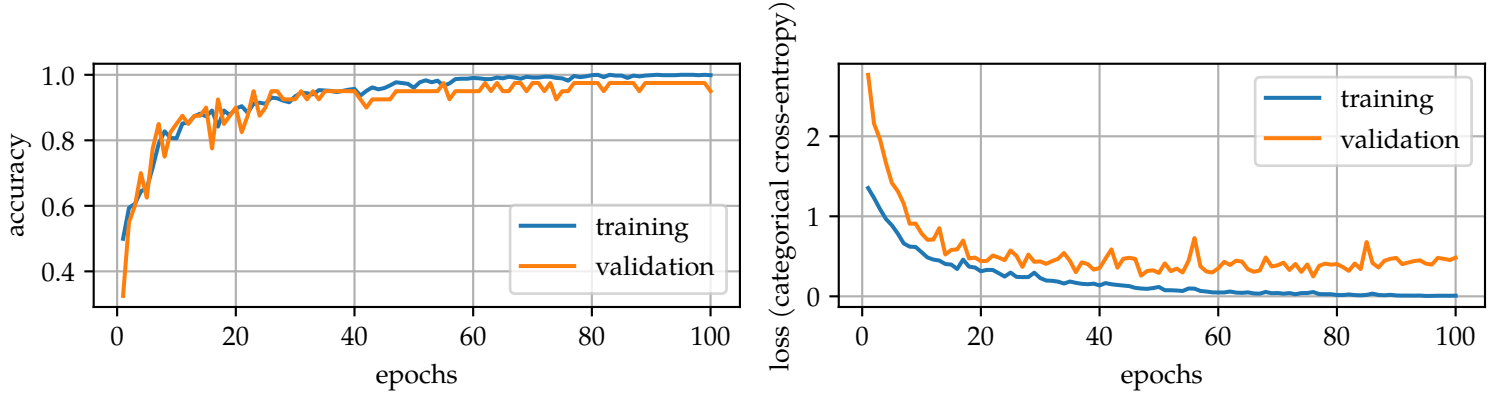


Figure: Training progression for the best CNN 2D model on raw features.

Test experiments - compressed - learning curves

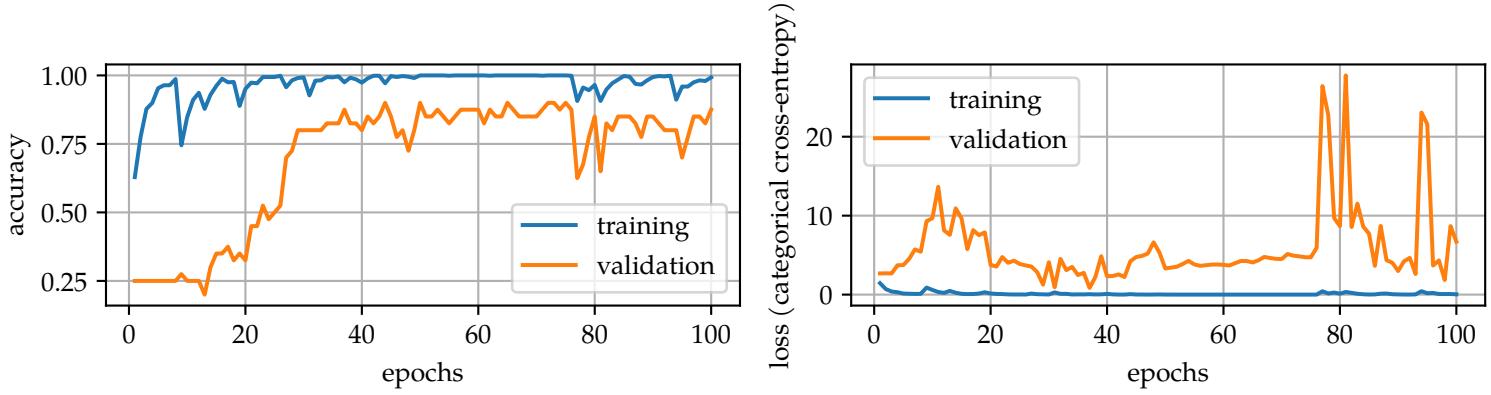


Figure: Training progression for DenseNet model on by 12% noise reduced raw features.

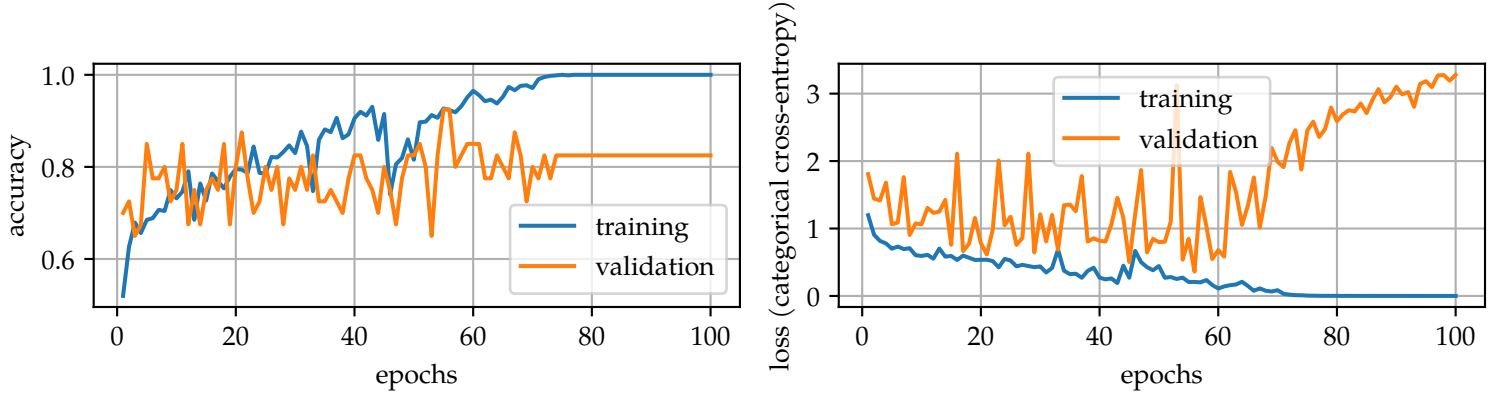


Figure: Training progression for the worst CNN 2D model on HOG features.

Test experiments - variable length - learning curves

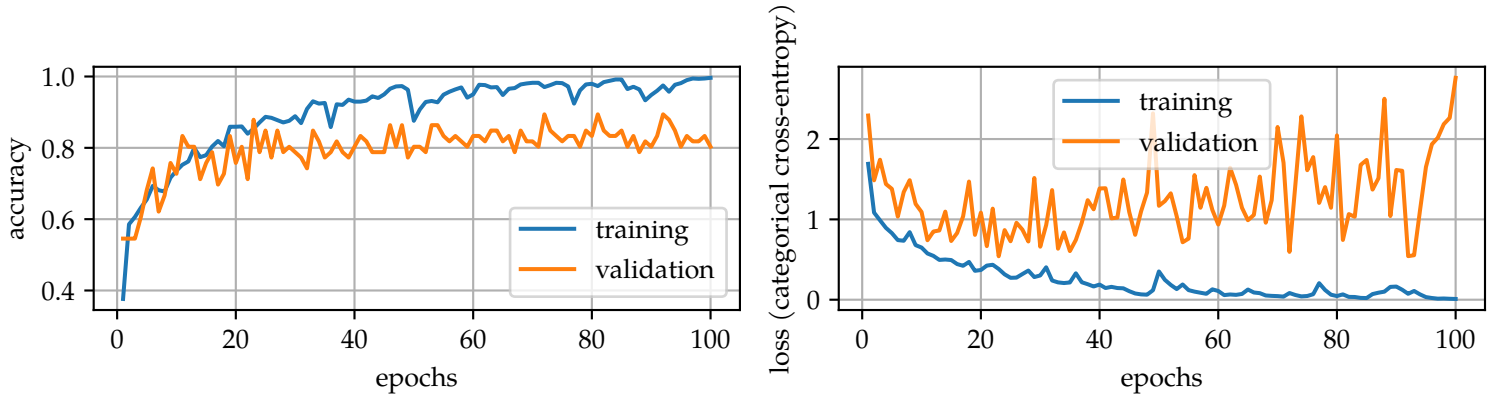


Figure: Training progression for the best LSTM model on HOG features.

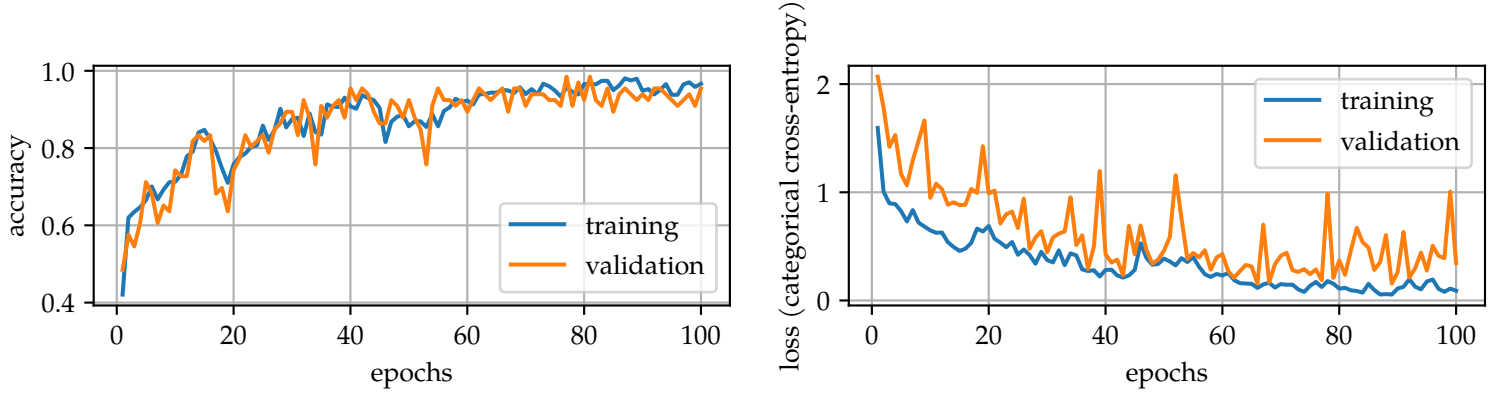
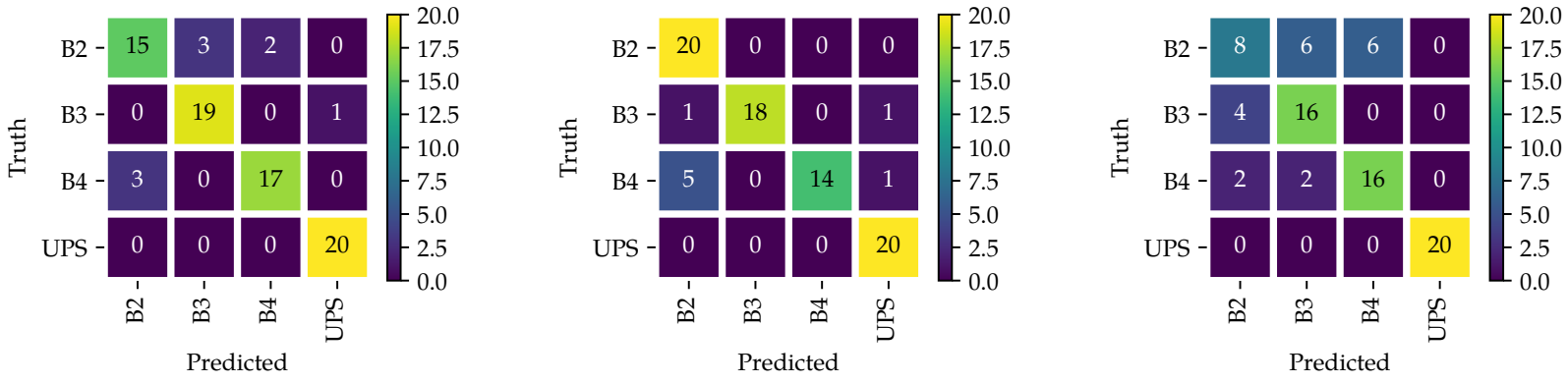


Figure: Training progression for LSTM on HOG features at high resolution.

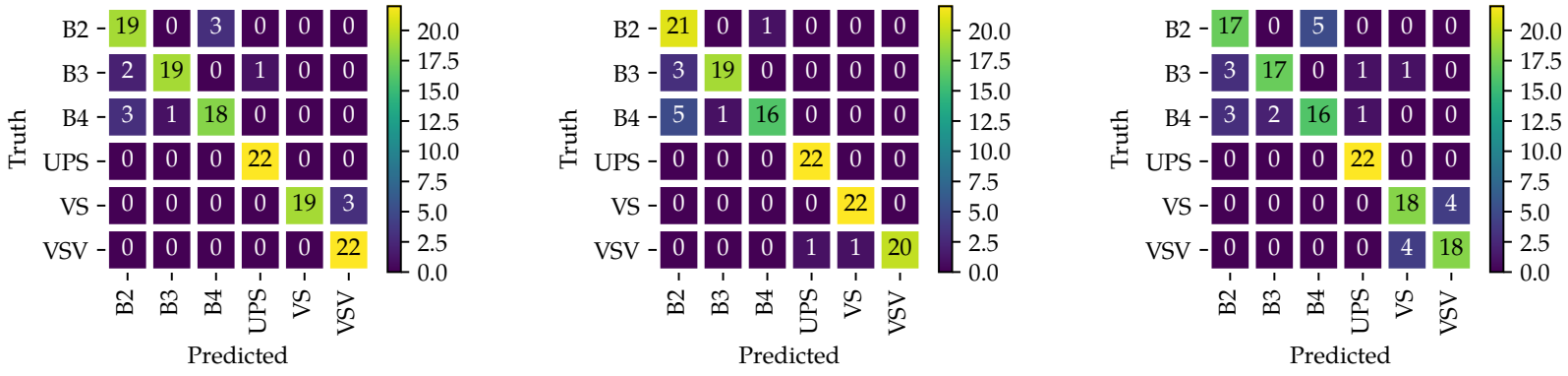
Test experiments - compressed - confusion matrices



A B C

Figure: Confusion matrices from some models of the compressed test experiment performing on the same test set: (A) DenseNet model, (B) CNN 2D model, (C) LSTM model.

Test experiments - padded - confusion matrices



A B C

Figure: Confusion matrices of some models of the padded test experiment performing on a test set: (A) the best DenseNet model, (B) the best LSTM model, (C) the best CNN 2D model.

Test experiments - resumé

- Compression of the features leads to a drop in the performance
- Variable length input combination with HOG features and a LSTM network yields promising results
- Padding on the wrong side could dramatically decrease the performance of the LSTM models
- Noise reduction tends to worsen the performance
- Higher resolution leads to better performance
- CNN's like raw features & LSTM's like HOG features more

Sequence experiments

- We evaluate a pipeline for a prototype of an automated syllable type classifier
- Adapted pipeline:
  - split audio recordings into slices in a windowing manner defined by window length and strides.
  - The audio slice is assigned to the corresponding syllable type as soon as a defined percentage or more of a slice is covered by a syllable annotation or the boundaries of the syllable annotation lie within the window.

Audio splitting algorithm

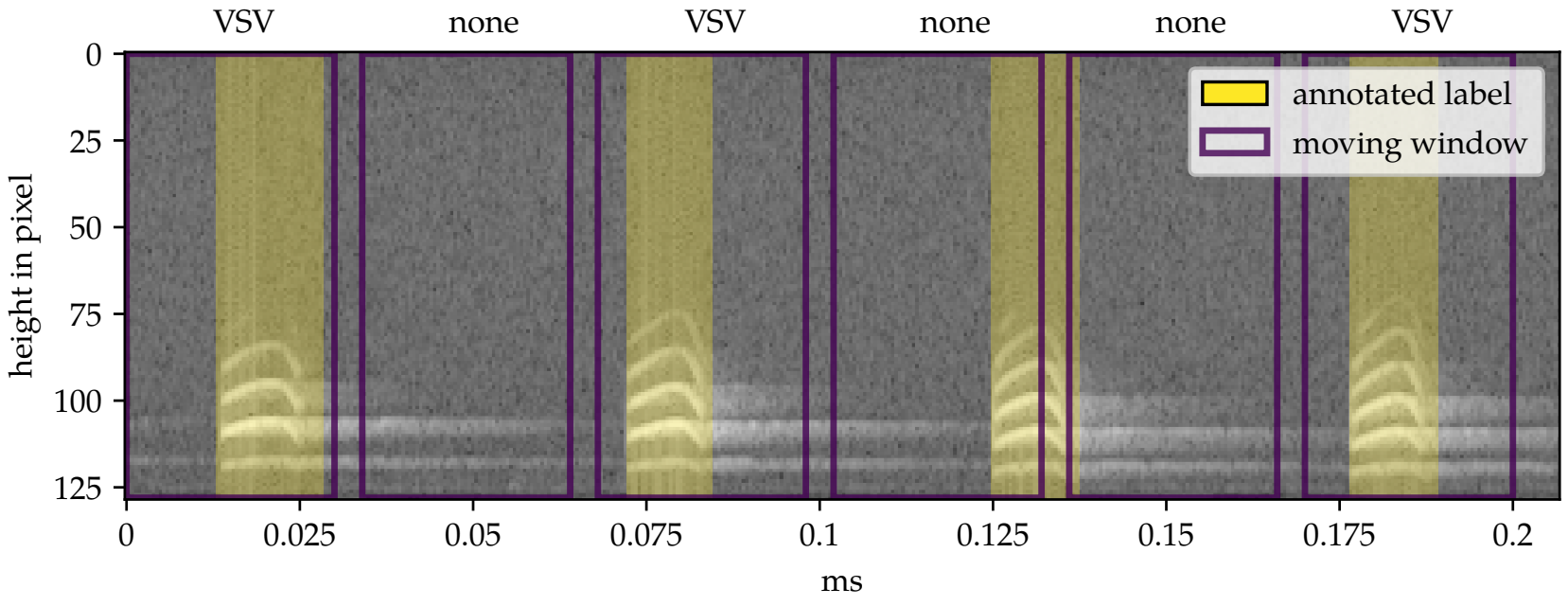


Figure: Visualisation of the audio splitting algorithm with a spectrogram as background from a labelled audio file. The rectangle with the blackcurrant borders represents the moving window with the assigned labels on the upper x-axes. The yellow rectangle covers the boundaries of the annotated syllable. For visualisation reasons, we used parameters (30 ms window length and 34 ms strides) that do not result in overlapping windows.

Sequence experiment - setup

- window length of 30 ms
- stride of 10 ms or 5 ms
- minimum coverage length (mcl) of 60% or 80% (minimum width of a slice that has to be covered by a syllable if the boundaries of the syllable are not within the slice.)

Sequence experiment - datasets

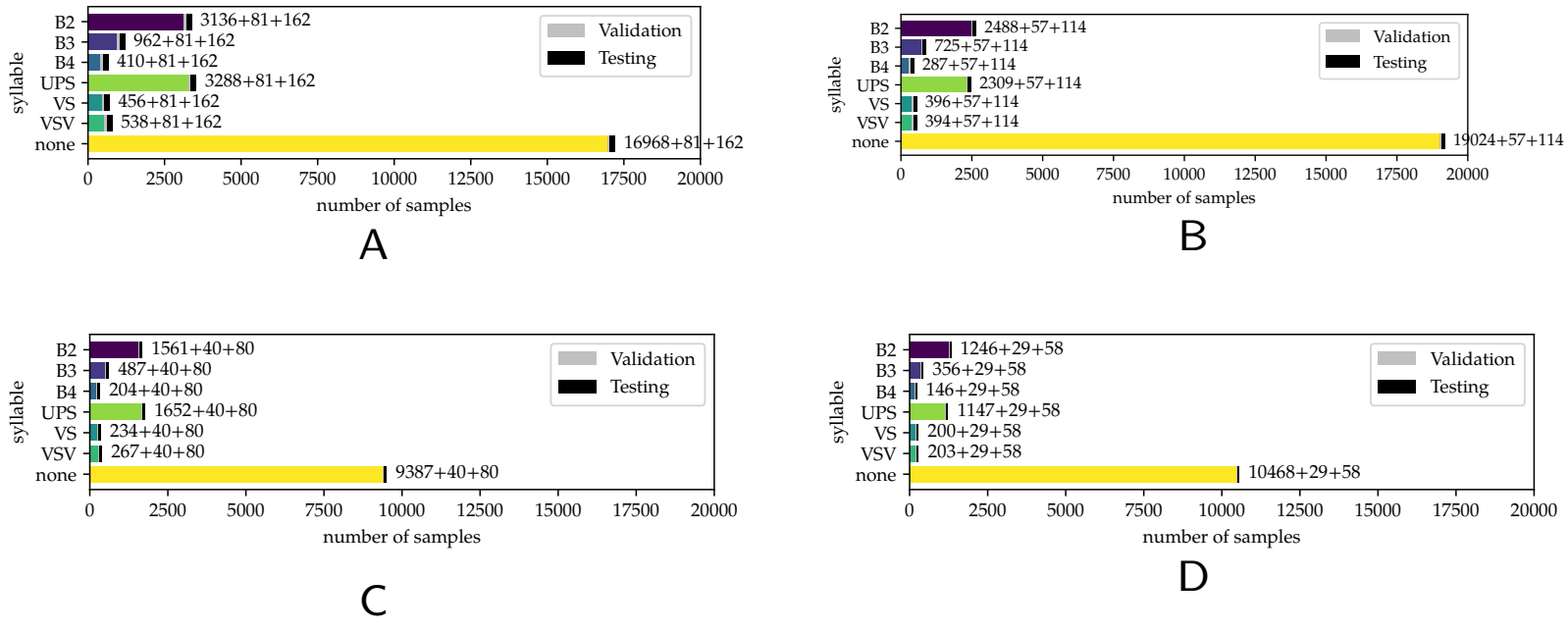
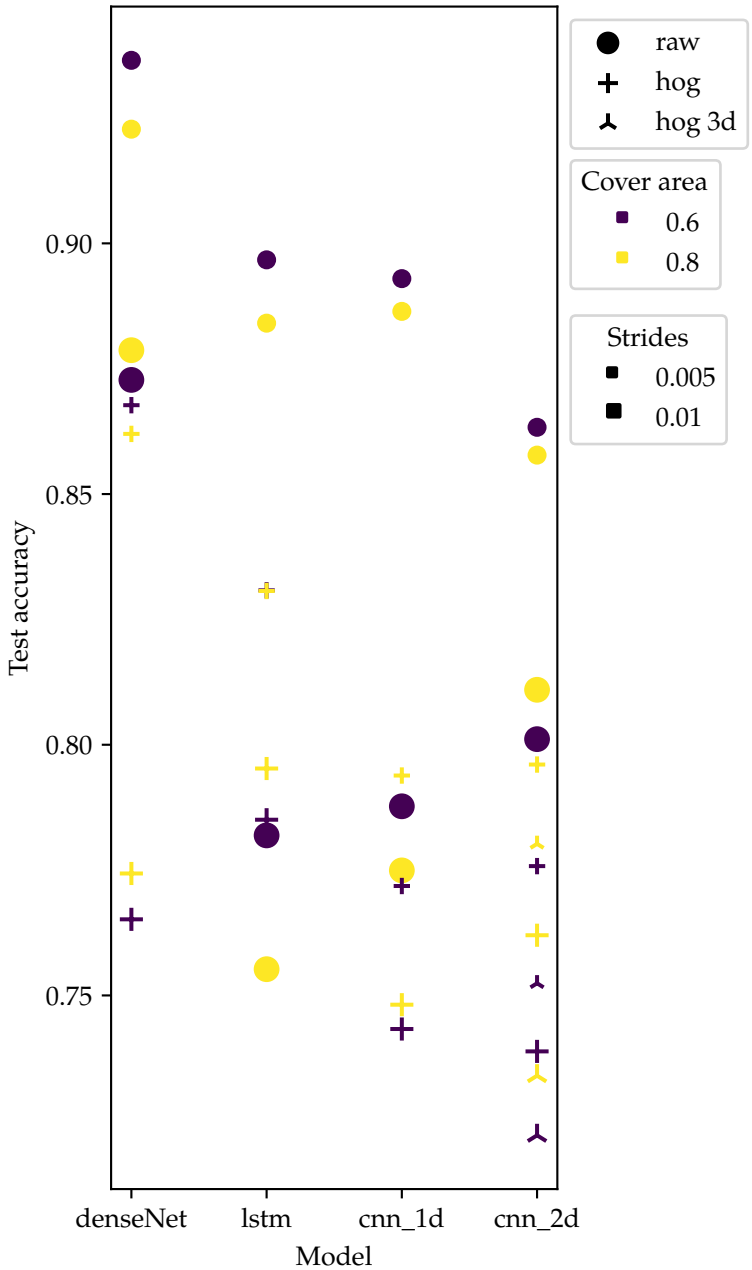


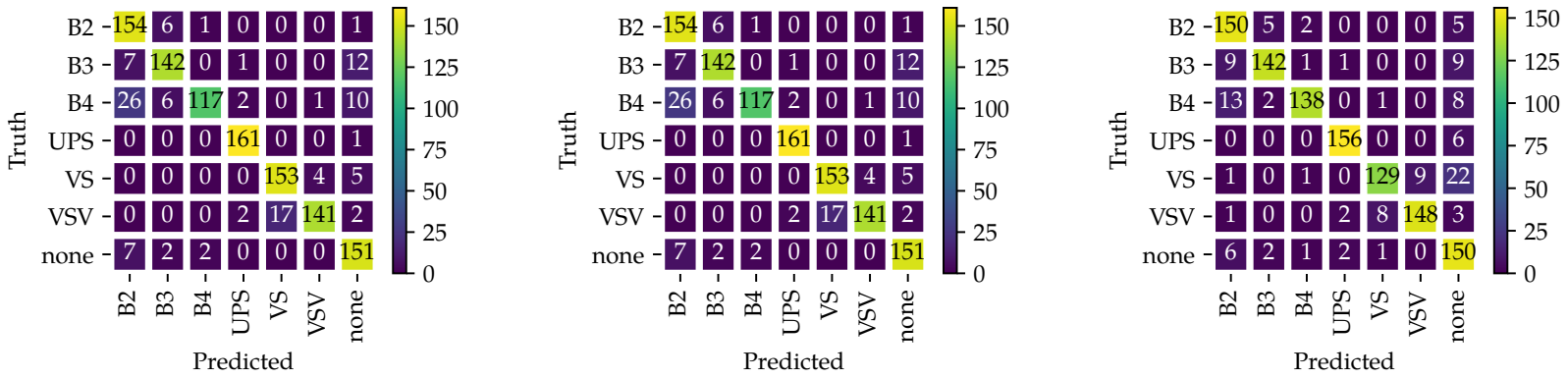
Figure: The datasets are not balanced with the ratio r expressing the number of training samples ratio between the least represented syllable and background samples. A) stride is 5 ms, mcl is 60% and r is around $1:41$. B) stride is 5 ms, mcl is 80% and r is around $1:66$. C) stride is 10 ms, mcl is 60% and r is around $1:46$. D) stride is 10 ms, mcl is 80% and r is around $1:72$.

Sequence experiment



Model	Test Accuracy (%)	Loss
DenseNet, stride: 0.005, mcl: 0.6, raw	93.7 ± 0.6	0.25 ± 0.04
DenseNet, stride: 0.005, mcl: 0.8, raw	92.3 ± 1.4	0.28 ± 0.09
LSTM, stride: 0.005, mcl: 0.6, raw	89.7 ± 1.1	0.43 ± 0.08
CNN 1D, stride: 0.005, mcl: 0.6, raw	89.3 ± 0.9	0.42 ± 0.06
CNN 1D, stride: 0.005, mcl: 0.8, raw	88.6 ± 1.1	0.39 ± 0.05
LSTM, stride: 0.005, mcl: 0.8, raw	88.4 ± 0.7	0.52 ± 0.04
DenseNet, stride: 0.01, mcl: 0.8, raw	87.9 ± 1.5	0.45 ± 0.08
DenseNet, stride: 0.01, mcl: 0.6, raw	87.3 ± 1.4	0.63 ± 0.17
CNN 2D, stride: 0.005, mcl: 0.6, raw	86.3 ± 1.5	0.55 ± 0.20
CNN 2D, stride: 0.005, mcl: 0.8, raw	85.8 ± 1.4	0.59 ± 0.14
CNN 2D, stride: 0.01, mcl: 0.8, raw	81.1 ± 2.0	0.57 ± 0.03
CNN 2D, stride: 0.01, mcl: 0.6, raw	80.1 ± 1.8	0.71 ± 0.18
CNN 1D, stride: 0.01, mcl: 0.6, raw	78.8 ± 2.6	0.89 ± 0.32
LSTM, stride: 0.01, mcl: 0.6, raw	78.2 ± 1.9	0.98 ± 0.15
CNN 1D, stride: 0.01, mcl: 0.8, raw	77.5 ± 2.4	0.81 ± 0.13
LSTM, stride: 0.01, mcl: 0.8, raw	75.5 ± 1.1	1.00 ± 0.17

Sequence experiment - confusion matrices



A B C

Figure: Confusion matrices of the following models performing on the test set: (A) the best DenseNet model, (B) the best LSTM model, (C) the best CNN 1D model.

Sequence experiment - resumé

- Short window length of 30ms and short step sizes of 5ms show the best result
- Syllables being confused with background and vice versa
- All models are overfitting.

Conclusion

- We were able to show that machine learning is suitable to classify the provided syllables with an acceptable accuracy based on the deep learning models applied
- We elicited which preprocessings give rise to or shorten the deep learning model performance
- There is a lot of space for future work
  - Models are overfitting
  - Improving the pipeline

Feature work

- Feature representation:
  - high level features
  - bit-representation
  - different HOG variants
- Pipeline modification:
  - Syllable segmentation based on static functions
  - Syllable segmentation based on trainable functions aké artificial neural networks
- Software optimization:
  - UX Design (GUI)
  - memory consumption
  - storage consumption

Acknowledgments

I would like to say a big thank you to my advisers:

Ahana Aurora Fernandez
Andreas Fischer
Daniel Wegmann

As well as my friends Anna, Bettina and Sophie who all proofread my thesis.

References

Hauser, M. D., Chomsky, N., & Fitch, W. T. (2002). Neuroscience: The faculty of language: What is it, who has it, and how did it evolve?

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 2261–2269.

Knörnschild, M., Nagy, M., Metz, M., Mayer, F., & von Helversen, O. (2010). Complex vocal imitation during ontogeny in a bat. *Biology Letters*, *6*(2), 156–159.

Vernes, S. C., & Wilkinson, G. S. (2020). Behaviour, biology and evolution of vocal learning in bats.

Waeber, G. (2019). Bird Voice Deep Learning.

Questions?

Dropout

- Randomly drops units with probability p from the network during training, which corresponds to an activation of zero
- This forces the model to create alternate decision paths
- To account for the scaled weights, the weights are multiplied by the probability $r = 1 - p$

Spectrogram

In our experiments we use a Fast Fourier transform which computes the discrete Fourier transform (DFT) with a sufficiently good time complexity. The DFT for a fixed N is the linear operator $\mathcal{F}: x^N \to X^N$ on $\mathbb{C}^N$ defined by:

$$X_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N} kn}, \quad k = 0, ..., N-1 \in \mathbb{Z} \tag{2}$$

Histogram of oriented gradients

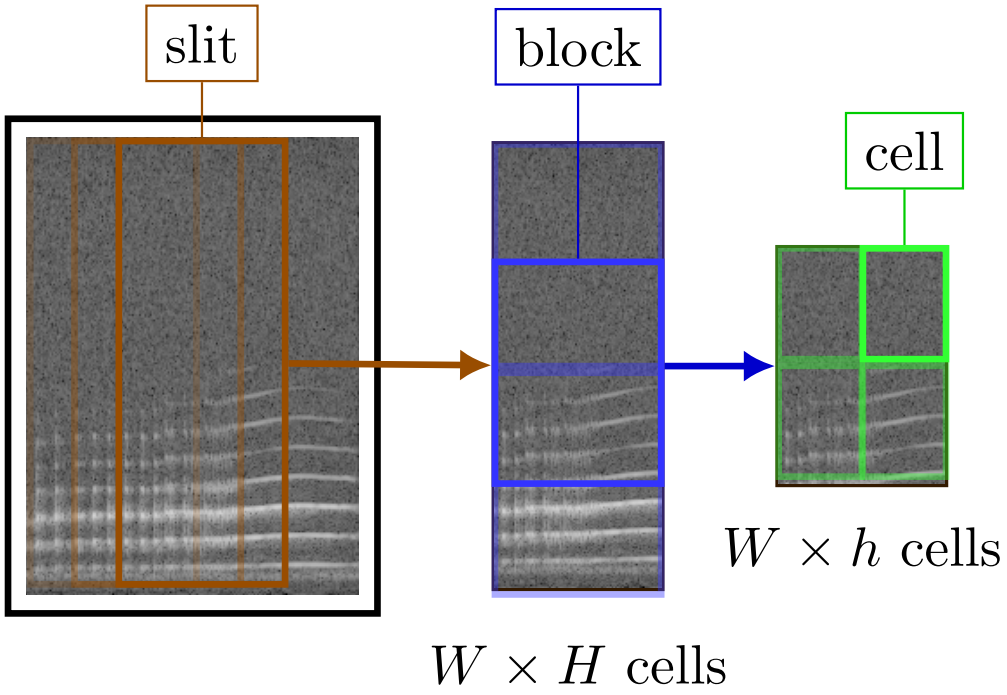


Figure: Visualisation of the different areas defined in the slit style HOG transformation and how they are encapsulated in each other. The slit contains 4×2 cells and a block is $h = 2$ cells high.

Cost function

$$C = -\frac{1}{n}\sum_{i}^{n}\left(y_i \ln \hat{y} + (1 - y_i)\ln\left(1 - \hat{y}_i\right)\right) \tag{3}$$

The convention is that $0 \ln 0 = 0$ but $\hat{y} \ln 0 = \infty$ for $\hat{y} \neq 0$ makes the loss function perfect for binary classification tasks.

Stochastic gradient descent

A basic stochastic optimizer used for deep learning is the SGD algorithm. This algorithm alters the model parameters by specific proportions in respect to the gradient value of the cost.

$$\Theta := \Theta - \eta \nabla J(\Theta)$$

Backpropagation

- feedforward network in a mathematically simplified way: $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$
- Each layer is described by a function $f^{(d)}$
- The first derivative of this function: $\frac{\partial f}{\partial x} = f^{(3)'}(f^{(2)}(f^{(1)}(x))f^{(2)'}(f^{(1)}(x))f^{(1)'}(x)$
- Memorise in forward pass $f^{(1)}(x)$.
- Backpropagation solves the partial derivative $\frac{\partial C}{\partial p}$ layer-wise

Test experiments - Layer wise relevance propagation

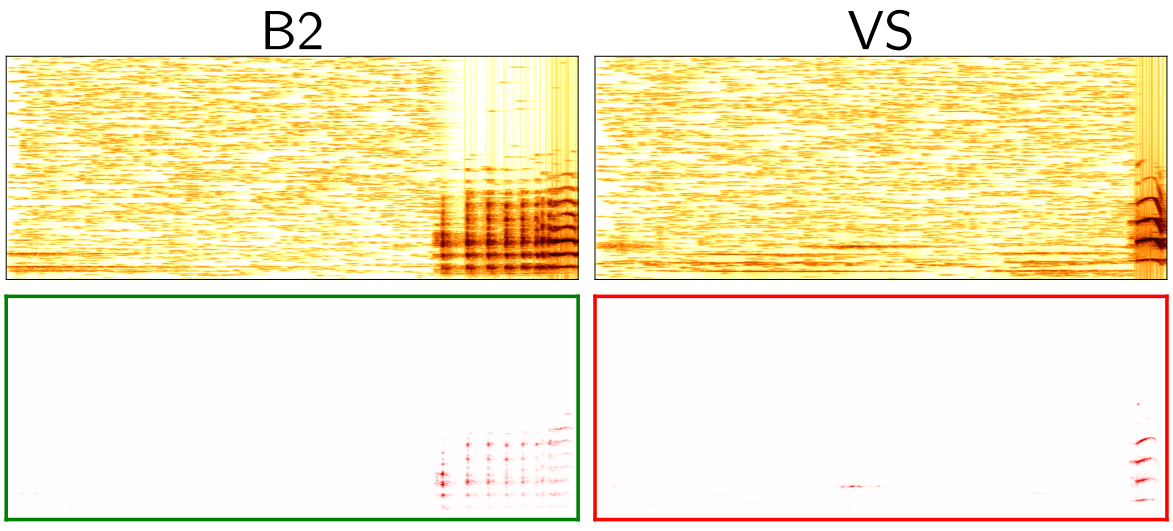


Figure: LRP of the best CNN 2D model from the padded test experiment on raw features (red = wrong prediction).