FUSING SHALLOW AND DEEP LEARNING FOR BIOACOUSTIC BIRD SPECIES CLASSIFICATION (PAPER 5)

Primary focus: Distinguishing between a large number of target species, i.e. large-vocabulary classification.

Method: We contrast the unsupervised dictionary learning approach with a deep convolutional neural network architecture which, in combination with data augmentation techniques, has been shown to outperform the dictionary learning approach for environmental (not bioacoustic) sound c1assification

Novelty: This is the first application of a deep convolutional neural network to flight call c1assification, and one of its first applications to bioacoustie c1assification in general. It is also the first application, as far as we know, of audio data augmentation (beyond simple time shifts) to bioacoustie c1assification.

Aim: To examine whether the dictionary learning method (which we consider a "shallow learning" technique because it learns a single representational layer from the input data) and the deep learning architecture are complementary to each other and whether combining their output in a late-fusion fashion can yield improved c1assification accuracy.

Methods in detail:

1. Unsupervised dictionary Learning :

Our "shallow learning" approach is based on *dictionary learning*.

- *spherical k-means algorithm* to learn a dictionary of representative code words, and then encode our data against the learned dictionary.
  - In this variant ofthe k-means c1ustering algorithm the centroids are constrained to have unit L2 norm (preventing them from becoming arbitrarily large or small)
  - its *resulting set of centroids* can be used as bases (a dictionary) for *mapping new data* into a *feature space* which reflects the discovered regularities
- *learn the dictionary* from time-frequency patches (TF-patches) extracted from the *log-scaled mel-spectrogram* representation of each audio clip
  - mel-spectrogram consists of *40 bands between 2000-11025 Hz*, and is computed using the *Essentia library* with a *Hann analysis window of 11.6 ms* and a *hop size of 1.45 ms*.
  - Each *TF-patch* spans the entire frequency range and has a *duration of 46 ms.*
- Procedure for *feature extraction* (learning dictionary + encoding data):
  - Prior to learning the dictionary the data is *PCA-whitened* (this improves the discriminative power of the learned features) keeping components to explain 99% of the variance in the data
  - We then learn a dictionary with 256 codewords (centroids)
  - Encode data against it by taking the matrix product of a datum's input representation (mel-spectogram) with the dictionary matrix

    - summarize the result over the time-axis using three summary statistics (mean, standard deviation and maximum)
- Training:
    - Resulting features are used to train (and test) a *Support Vector Machine* (SVM) classifier with a *radial basis function kernel* implemented in Python using *Scikit-learn*

2. Deep convolutional neural networks:

- The model is comprised of *3 convolutional layers* interleaved with *2 pooling* operations, followed by *2 fully connected (dense)* layers.
- Input
    - *log-scaled mel-spectrograms* with 128 *bands*
    - *trim* each clip to the *middle 150 ms* since clips in our evaluation dataset are of varying duration
    - To make the *network invariant* to *small time-shifts*, each clip is *shifted* in time by *up to ± 10 ms* to generate a total of 13 clips .
- Network structure
    - l1: *24 filters* with a *receptive field* of (5,5), followed by (4,2) *strided (non-overlapping) max-pooling* over the time and frequency dimensions respectively, and a rectified linear unit (*ReLU*) activation function h(x) = max(x, 0).
    - l2: *48 filters* with a *receptive field* of (5,5), followed by (4,2) *strided max-pooling* and a *ReLU* activation function.
    - l3 : *48 filters* with a *receptive field* of (5,5), followed by a *ReLU* activation function (no pooling)
    - l4: *64 hidden units*, followed by a *ReLU* activation function.
    - L5: *43 output units*, followed by a *softmax* activation function.
- Optimization & loss function :
    - optimizes *cross-entropy loss* via *minibatch stochastic gradient descent*
- Parameters:
    - *Constant learning rate of 0.01* and apply *dropout* to the input of the *last two layers with probability 0.5*.
    - *L2-regularization* is applied to the weights of the *last two layers* with a *penalty factor* of 0.001.
    - 100 *epochs* and is *checkpointed after each epoch*.
    - Validation: validation set is used to *identify the parameter setting* (epoch) achieving the highest classification accuracy
- CNN is implemented in Python using *Lasagne*
- *data stream multiplexing* (for training) is implemented using *Pescador*
- Improvements: increasing the size of the training set using *data augmentation*
    - adding *background noise*
    - *dynamic range compression*
    - *pitch shifting*
    - *time stretching*
    - augmentations are applied using the *MUDA library*

- *Point to Note :* Of the augmentations applied in this study, only *pitch shifting* and *time stretching* have the potential to invalidate a recording's label. To avoid this, the shifting/stretching *parameter ranges* were chosen to be *within the naturally occurring ranges for the majority of the species* in the evaluation dataset

Baseline model:

- Method :
  - *extracts Mel-Frequency Cepstral Coefficients* (MFCC) using *40 mel bands* and keeps the *first 25 coefficients*, which are summarized over time using 11 summary statistics
  - resulting *feature vectors* are used to train (and test) an *SVM classifier* with a *radial basis function kernel*.

Results:

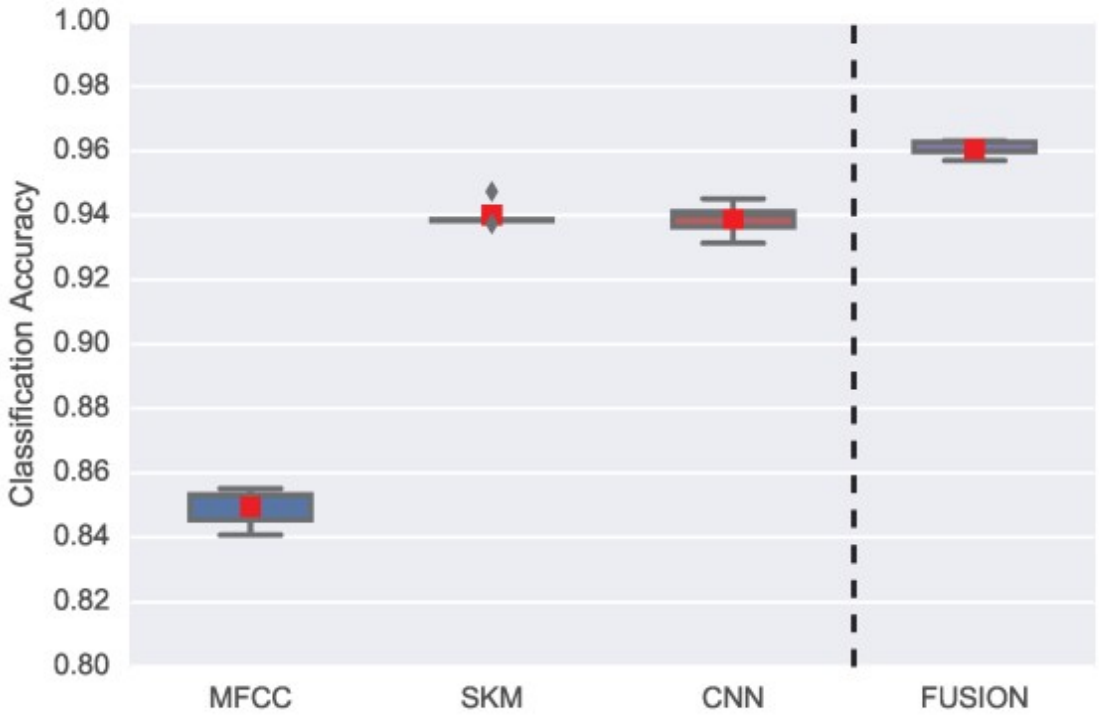

Fig. 1. Classification accuracy: baseline (MFCC), dictionary (shallow) learning (SKM), deep convolutional neural network (CNN), and late fusion of the SKM and CNN models (FUSION).

HANDCRAFTED FEATURES AND LATE FUSION WITH DEEP LEARNING FOR BIRD SOUND CLASSIFICATION (PAPER 8)

Overview:

- investigate acoustic features, visual features, and deep learning for bird sound classification.
- For the deep learning approach, the *CNN* layers are used for *learning generalized features and dimension reduction*, while a *conventional fully connected layer* is used for *classification*.
- Last fusion of acoustic features approach, visual & deeplearning approach (95.95%) > Deep learning method (94.36%) > acoustic features 88.89%) (> visual features (88.87%)

Method:

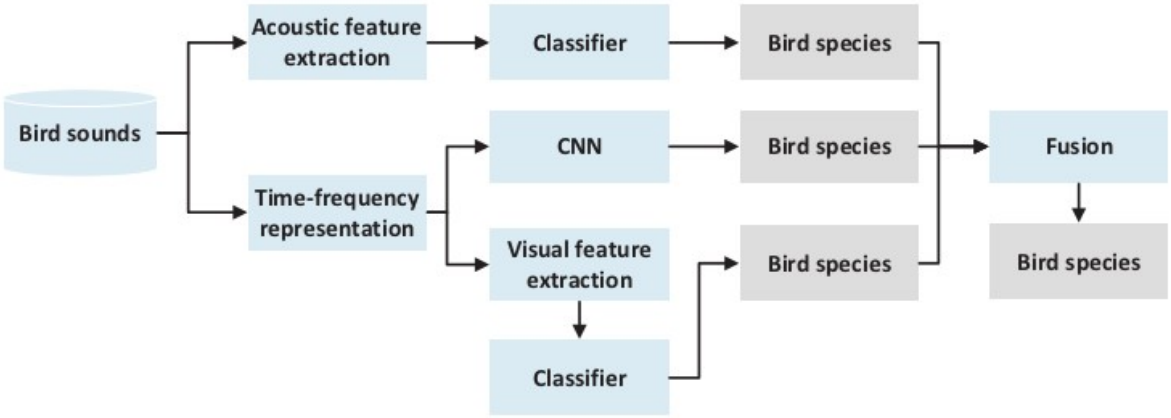

Fig. 1. The flow diagram of our proposed approach.

Audio Images:

- We convert the audio signals to a log scale time-frequency representation using the Constant-Q Transform (CQT) (1D to 2D)
- CQT transform of time-domain signal :

$$X[k,n] = \sum_{q=n-\lfloor N_k/2 \rfloor}^{n+\lfloor N_k/2 \rfloor} x(q) a_k^*(q - N + N_k/2)$$

Acoustic Features:

- *Spectral centroid* is the center point of spectrum distribution. With the magnitudes as the weight, it is calculated as the *weighted mean of frequencies*
- *Spectral bandwidth* can be used to represent the difference between the upper and lower *cut-off* frequencies.
- *Spectral contrast* is defined as the *decibel difference* between *peaks and valleys* in the spectrum
  - Then, all spectrum is sorted in a descending order, which can be represented

- as {x k, 1 ′ , x k, 2 ′ …, x k, N ′ }, where x k, 1 ′ > x k, 2 ′ > x k, N ′ . Next, the strength of spectral peaks and spectral valleys are estimated as:

$$Peak_k = log \frac{1}{\alpha N} \sum_{i=1}^{\alpha N} x'_{k,i}$$

$$Valley_k = log \frac{1}{\alpha N} \sum_{i=1}^{\alpha N} x'_{k,N-i+1}$$

Finally, spectral contrast is defined as

$$SC_k = Peak_k - Valley_k$$

where N is the total number in k-th sub-band.

- *Spectral flatness* provides a way to quantify the *tonality* of a sound. A higher spectral flatness indicates a similar *amount of power of the spectrum in all spectral bands*. Spectral flatness is measured by the ratio between the geometric mean and the arithmetic mean of the power spectrum and defined as

$$Sf = \frac{\sqrt{\frac{1}{N}\sum_{k=0}^{N-1} InX(k)}}{\frac{1}{N}\sum_{k=0}^{N-1} X(k)}$$

- *Spectral roll-off* is often used to measure the *spectral shape*, and defined as the *frequency* H. Here H is the value below which θ of the magnitude distribution is concentrated.

$$\sum_{k}^{H} X(k) = \theta \sum_{k=1}^{N-1} X(k)$$

where θ is set to 0.8.

- Zero-crossing rate denotes the rate of signal change along a signal. When adjacent signals have different signs, a zero-crossing occurs. The mathematical expression of zcr is shown as follows.

$$zcr = \frac{1}{2} \sum_{n=0}^{L-1} [sgn(x(n)) - sgn(x(n+1))]$$

where $x(n)$ is the framed signal, L is the length of the frame.

- Energy of a signal corresponds to the total magnitude of thesignal, which roughly corresponds to how loud the signal is. root-mean-square energy (RMSE) in a signal is defined as

$$RMSE = \sqrt{\frac{1}{N} \sum_{n} |x(n)|^2}$$

- *Mel-frequency Cepstral coefficients (MFCCs),* which are obtained by applying discrete cosine transform to a sub-band Mel-frequency spectrum within a short time
  - Step 1: *Band-pass filtering*: The amplitude spectrum is filtered using a set of triangular band-pass filters.

$$E_j = \sum_{k=0}^{N/2-1} \phi_j(k) A_k,\ 0 \le j \le J-1 \tag{13}$$

where J is the number of filters, ϕ_j is the j^{th} filter, and A_k is the amplitude of $X(k)$.

$$A_k = |X(k)|^2,\ 0 \le k \le N/2 \tag{14}$$

  - Step 2 : *Discrete cosine transform*: MFCCs for the i th frame are computed by performing DCT on the logarithm of E j .

$$C_m^j = \sum_{j=0}^{J-1} \cos\left(m\frac{\pi}{J}(j+0.5)\right) log_{10}(E_j),\ 0 \le m \le L-1 \tag{15}$$

where L is the number of MFCCs. The filter bank consists of 40 triangular filters, that is $J = 40$. The length of MFCCs of each frame is 12 (L = 12).

Visual Features :

- *Local binary pattern :* For LBP, each pixel of an image is calculated by comparing each central pixel, g c , with its neighboring pixels, g p . Here the distance between central pixel and P neighboring pixel is denoted by R. The calculation of LBP is shown as

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p$$

where

$$s(x) = \begin{cases} 1 & \text{if } x \ge 0 \\ 0 & \text{if } x < 0 \end{cases}$$

- *Histogram of oriented gradients*: HOG is used for emphasizing the fast spectral transitions between adjacent frames within an acoustic scene.
  - compute the gradient of the CQT representation;
  - compute angles of all pixel gradients;
  - split images into non-overlapping cells;

- count the occurrence of gradient orientations in a given cell;
- eventually normalize each cell histogram according to the histogram norm of neighboring cells.

- *filtering* and *pooling* are used for optimizing the HOG.
- pooling *size* is chosen among [(1,64); (64,1)]

KNN:

METHOD 1:

- For the k-NN classifier, an object is classified to the *majority class of its k nearest neighbors*
- For testing, The k closest vectors are selected for voting, then the classification for the input feature vector f i, c is assigned with the majority class.

- METHOD 2:
  - second classification combination method is to calculate the average distance between an input bird feature vector and k clos est vectors.
  - For example, the Euclidean distance between an input feature vector f (i, c) and one stored feature vector f (j, c) is calculated as

$$d(i,j) = \sqrt{\sum_{c=1}^{n} (f_{i,c} - f_{j,c})^2}$$

  - where i and j are indices of the feature vector, n means the dimension of the feature vector.
  - Next, k nearest neighbors of feature vector i are selected based on the Euclidean distance for voting.

$$\frac{1}{k_1} \sum_{j \in s_1} d(i, j(s_1)) < \frac{1}{k_2} \sum_{j \in s_2} d(i, j(s_2))$$

- where k = k 1 + k 2 , k 1 is the number of bird species s 1 , k 2 is the number of bird species s 2 . Here, the input feature vector i will be classified as bird species s 2 (if the above equation is satisfied)

METHOD 3:

- The third classification combination method is to calculate the sum of similarity of k closest feature vectors. For a binary classification task with two classes: k1 and k2 .
- Then the input feature vector i will be classified as belonging to class s2 if the below equation is satisfifed :

$$\sum_{j \in s_1} d(i, j(s_1)) < \sum_{j \in s_2} d(i, j(s_2))$$

- the distance function used for K-NN is the Euclidean function, and k is chosen among [1, 10, 20, 30, 40, 50].

Random Forest

- Random forest (RF) is a tree-based algorithm, which builds a specified number of classification trees without pruning
- The nodes are split on a random drawing of m features from the entire feature set M.
- A bootstrapped random sample from the training set is used to build each tree.
- advantage of RF is its ability to generate a metric to rank predictors based on their relative contribution to the model's predictive accuracy

$$Pred = \frac{1}{K}\sum_{n=1}^{K} T_i \tag{21}$$

where T_i is the n-th tree response of the RF. In this work, the number of trees K is chosen among [100, 1000, 2000, 3000, 5000]. As for the predictor variables m, it is set at $\sqrt{N}$, where N is the feature dimension.

Deep Learning for bird species classification

- The CNN architecture consists of *three* convolutional layers.
- *Receptive field of 5 × 5* followed by a *max pooling* operation for every convolutional layer.
- Rectified linear unit (ReLU) is used as an activation function.
- Dropout is employed in convolutional layers with rate 0.3 to address over-fitting.
- The CNN is optimized by employing back-propagation algorithm.

Table 2
CNN architecture. The data shape indicates time × frequency × number of filters.

Layer(type)	Filter/Stride	Output Shape	Param #
Conv1	5 × 5/2 × 2	64 × 250 × 128	3328
MaxPool1	5 × 5	32 × 125 × 128	
Conv2	5 × 5/2 × 2	16 × 63 × 128	400 K
MaxPool2	5 × 5	8 × 31 × 128	
Conv3	5 × 5/2 × 2	4 × 16 × 128	400 K
MaxPool3	5 × 5	2 × 8 × 128	
FC1		256	524 K
FC2		14	3598
Total			1351 K

Late Fusion:

- *class-based* late fusion method.
- It finds the final posterior probability (for all classifiers) for every class and then declares the class with the highest posterior prob as the final prediction.

Evaluation Rule:

- Accuracy and weighted F1-score

Results

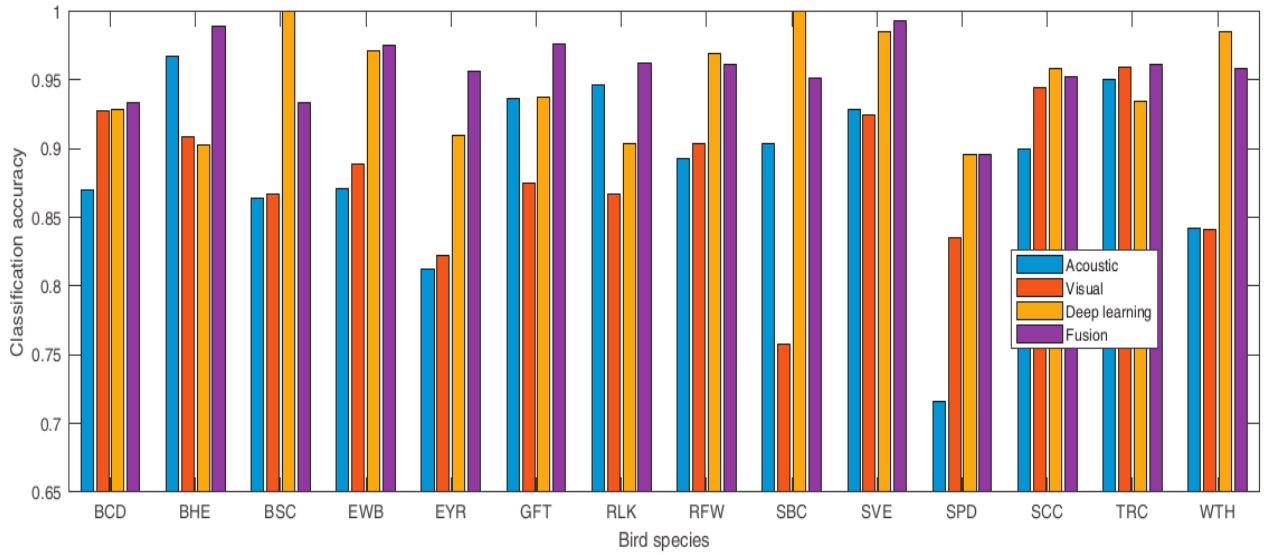

Fig. 5. Class-wise accuracy comparison of 14 bird species with four different classification methods. Here, the fusion is realized by acoustic feature approach, visual feature approach, and deep learning.