

FROM PDES TO WINGBEATS: A NOVEL CONVOLUTIONAL FOURIER LAYER-BASED RESNET MODEL

Anonymous authors

Paper under double-blind review

ABSTRACT

Recent advancements in Deep Learning apply Fourier Neural Operators (FNOs) for generating numerical solutions of Partial Differential Equations (PDEs). They are efficient due to their global spectral representations. However, their abilities in applied classification or regression tasks for time series have not been studied previously. We further investigate the motivation behind FNOs and provide a more detailed Discrete Fourier Transform-based definition. Furthermore, we introduce *CF-ResNet-1D*, a novel ResNet-inspired model built from Convolutional Fourier Layers being parallel units of FNO and 1D-Convolution. *CF-ResNet-1D* can perform time-series data analysis on raw time-domain signals while also taking advantage of the parallel spectral processing of the FNOs. This combined processing method outperforms spectrogram-based models for insect wingbeat sound classification, achieving state-of-the-art accuracy on benchmark datasets. The outcomes of our research offer promising insights about FNO application in real-world problems, such as mosquito management and the mitigation of insect-related diseases.

1 INTRODUCTION

A Convolutional Neural Network (CNN) can learn time series features well using one-dimensional local convolutional kernels (Goodfellow et al., 2016). On the other hand, time series can also be thought of as continuous functions. Therefore, they can be represented in Fourier space and global convolutions can be used to obtain global features.

The Fourier Neural Operator (FNO) (Li et al., 2021) is a modern technique that builds on top of the Discrete Fourier Transform. This neural operator performs learnable transformations in the Fourier domain as a layer-level unit of a deep neural network. These transformations are analogous to convolutions (Kabri et al., 2023) in nature, with the advantage of directly accessible frequency components, thus they are suitable for frequency-based machine learning processes. These operators are extensively used in modeling partial differential equations (PDEs) that govern mathematical and physical simulations providing state-of-the-art engineering solutions (Chaohao et al., 2022; Li et al., 2022; Zhang et al., 2022).

However, only a few studies focus on applying FNOs for classical data science-motivated tasks. Among these according to our best knowledge, only image classification capabilities have been explored (Johnny et al., 2022; Kabri et al., 2023). Both of these studies use FNOs as a global alternative to the convolution operator. While this is a good strategy for processing high-level features, locality is also important in such CNNs. To explore the time series processing-related capabilities of the FNO operator we apply them in the domain of insect wingbeat sound classification.

Every year, around 700 million people are infected and more than one million die from mosquito-borne diseases (Caraballo & King, 2014). Such diseases include malaria, dengue, Zika virus fever, yellow fever, West Nile fever, and encephalitis viruses (Palmer et al., 2017). For each disease, one or more mosquito species are responsible for the transmission. Monitoring is an essential part of control and successful intervention. This is why it is important to classify mosquitoes efficiently and quickly.

Two types of machine learning approaches are applied for this task: audio and image-based. For image-based solutions, classification is based on images of mosquitoes (Okayasu et al., 2019; Motta

et al., 2020; Kittichai et al., 2021). It has previously been shown that the wingbeats of different species have different audio characteristics (Offenhauser & Kahn, 1949), this also justifies the examination of audio or audio-like signals. For audio-based methods, either the raw wingbeat recording is used (Yin et al., 2021), or data obtained after applying some kind of transformation on the raw signals, such as the short-time Discrete Fourier Transform or the Discrete Fourier Transform (Arthur et al., 2014; Chen et al., 2014b; Ouyang et al., 2015; Vasconcelos et al., 2019; Luna-Gonzalez et al., 2020; Wei et al., 2022).

A system was presented to record the wingbeat of insects based on the large aperture optical sensors that turn the light fluctuations into sound (Potamitis & Rigakis, 2016; Rigakis et al., 2019). Here, the light fluctuations are caused by the partial occlusion of light from the wings. Without aiming to be exhaustive, the acquired datasets include namely *Wingbeats* and *Fruitflies*. The *Wingbeats* dataset was investigated using different state-of-the-art deep learning architectures (Wei et al., 2022; Fanioudakis et al., 2018). Fanioudakis et al. (2018) converted the audio signals into spectrograms, and a DenseNet-121 based model achieved 96% test accuracy. However, these results could not be reproduced by Wei et al. (2022). Chen et al. (2014a) have used a very similar technique to capture insect flying sounds, the collected dataset is called *Insects*. Mukundarajan et al. (2017) have shown that even low-cost mobile phones are capable of acquiring acoustic data on mosquito wingbeat sounds. The resulting *Abuzz* dataset was investigated with the same technique and with the same *WbNet* architecture (Wei et al., 2022) as the *Wingbeats* dataset, too.

In our contribution, we provide further details on the mathematical motivation of the FNO for discrete time series processing. One advantage of this task type is that the heuristic justification of the FNO definition is much clearer. We provide a refined definition of the Discrete Fourier Neural Operator that is directly applicable to Deep Learning. We propose a novel Convolutional Fourier Layer by extending FNO’s point-wise operation to a proper 1D-Convolution. This Convolutional Fourier (CF) Layer is then used to build ResNet-style models (CF-ResNet-9-1D) that are capable of sound classification. To the best of our knowledge, this is the first attempt that FNO has been used to investigate a real-world time series dataset. We test the proposed model on insect wingbeat sound classification benchmarks, due to the impact of their possible applications.

We run numerical experiments on the *Wingbeats*, *Fruitflies*, *Insects*, and the *Abuzz* datasets for evaluation using the above-mentioned state-of-the-art spectrum-based MobileNet, DenseNet-121, and *WbNet* architectures. We also report the performance of classical 1D ResNet architectures and vanilla FNO-based ResNet architectures trained on raw time-series data. We repeat experiments by Wei et al. (2022) and Fanioudakis et al. (2018) as their original implementation does not differentiate between ¹ validation and test data, thus it contains an inherent data leakage which results in misleading test performance.

Our proposed model achieves state-of-the-art results with the largest one overperforming all of the baselines on the majority of benchmark datasets. Furthermore, Convolutional Fourier Layers significantly improve the performance compared to the original FNO implementation by Li et al. (2021).

The paper is structured as follows. In Section 2, we provide some background materials concerning the definition of the Discrete Fourier Neural Operator and the investigated *Wingbeats*, *Fruitflies*, *Insects* and *Abuzz* datasets. Also, the proposed *CF-ResNet-9-1D* architecture is described in detail, then the corresponding classification results are presented in Section 3. Finally, we summarize our findings in Section 4.

2 METHODS

2.1 DISCRETE FOURIER NEURAL OPERATOR

In this subsection, we discuss the motivation behind defining a crucial component of our proposed architecture: the Fourier Neural Operator, which is based on the work by Li et al. (2021). In this motivation part, rigorous mathematical precision is ignored. Although, the precise definition of the operator is given at the end of this subsection.

¹Reevaluated experiment solving data leak in the original implementation.

Let the numbers $I, O \in \mathbb{N}$ and the functions $r \in L^1(\mathbb{R}; \mathbb{R}^{O \times I})$, $u \in L^1(\mathbb{R}; \mathbb{R}^I)$ be given. Here, r is called as the kernel function. The convolution of these functions $v = r * u \in L^1(\mathbb{R}; \mathbb{R}^O)$ is obtained by the next formula.

$$v(x) = (r * u)(x) = \int_{\mathbb{R}} r(x-y) u(y) dy, \quad x \in \mathbb{R}. \quad (1)$$

Applying the convolution theorem the following equation holds.

$$v(x) = \mathcal{F}^{-1}(\mathcal{F}(r) \cdot \mathcal{F}(u)) = F_r(u)(x), \quad x \in \mathbb{R}. \quad (2)$$

The operator F_r in equation (2) is called the Fourier Neural Operator. Assume that $D = (-\pi, \pi)$ and $r \in L^2(D; \mathbb{R}^{O \times I})$, $u \in L^2(D; \mathbb{R}^I)$ also hold true. Here, we assume the quadratic integrability holds per component. Let $u(x) = (u_1(x), u_2(x), \dots, u_I(x))^T$, or shortly $u(x) = (u_j(x))^T$ given by its Fourier series componentwise, i.e. $u_j(x) = \sum_{k \in \mathbb{Z}} U_{k,j} e^{ikx}$. Similarly, we suppose that $r(x) = (r_{l,j}(x)) \in \mathbb{R}^{O \times I}$, where $r_{l,j}(x) = \sum_{k \in \mathbb{Z}} R_{k,l,j} e^{ikx}$. The following equations are used.

$$\mathcal{F}(x \mapsto e^{ikx})(y) = \sqrt{2\pi} \delta(y-k) \quad \text{and} \quad \mathcal{F}^{-1}(y \mapsto \sqrt{2\pi} \delta(y-k))(x) = e^{ikx}, \quad (3)$$

where δ is the Dirac-delta distribution. Let us assume that the Fourier transformation can be carried out term by term in the Fourier series of the r and u functions. In this case, $\mathcal{F}(r)(y) \in \mathbb{C}^{O \times I}$ and $\mathcal{F}(u)(y) \in \mathbb{C}^I$. With this assumption, we investigate their product, namely $\mathcal{F}(r)(y) \cdot \mathcal{F}(u)(y) \in \mathbb{C}^O$. The l -th component of this vector is to be calculated as follows.

$$[\mathcal{F}(r)(y) \cdot \mathcal{F}(u)(y)]_l = \sum_{j=1}^I \mathcal{F}(r_{l,j})(y) \cdot \mathcal{F}(u_j)(y) \quad (4)$$

It is necessary to calculate $\mathcal{F}(r_{l,j})(y)$ and $\mathcal{F}(u_j)(y)$, which, taking the Fourier transform element-wise in the Fourier series satisfy the following equations.

$$\mathcal{F}(r_{l,j})(y) = \mathcal{F}\left(\sum_{k \in \mathbb{Z}} R_{k,l,j} \cdot e^{ikx}\right)(y) = \sqrt{2\pi} \sum_{k \in \mathbb{Z}} R_{k,l,j} \delta(k-y), \quad (5)$$

similarly

$$\mathcal{F}(u_j)(y) = \mathcal{F}\left(\sum_{k \in \mathbb{Z}} U_{k,j} \cdot e^{ikx}\right)(y) = \sqrt{2\pi} \sum_{k \in \mathbb{Z}} U_{k,j} \delta(k-y). \quad (6)$$

Using the identities in the formulas (5)-(6), equation (4) can be further modified as follows.

$$\begin{aligned} & [\mathcal{F}(r)(y) \cdot \mathcal{F}(u)(y)]_l \\ &= \sum_{j=1}^I \left(\sqrt{2\pi} \sum_{k \in \mathbb{Z}} R_{k,l,j} \delta(y-k) \right) \cdot \left(\sqrt{2\pi} \sum_{k \in \mathbb{Z}} U_{k,j} \delta(y-k) \right) \\ &= 2\pi \sum_{j=1}^I \sum_{k \in \mathbb{Z}} R_{k,l,j} \cdot U_{k,j} \delta(y-k) = 2\pi \sum_{k \in \mathbb{Z}} \sum_{j=1}^I R_{k,l,j} \cdot U_{k,j} \delta(y-k). \end{aligned} \quad (7)$$

At the last equation, it was assumed that the two sums can be exchanged. Let us note that this is the Fourier transform of the function $v(x) = (v_1(x), v_2(x), \dots, v_O(x))$, $v \in L^2(D; \mathbb{R}^O)$, given by the following Fourier series, assuming that we can take its Fourier transform term by term in its Fourier series.

$$v_l(x) = \sum_{k \in \mathbb{Z}} \sum_{j=1}^I R_{k,l,j} \cdot U_{k,j} \sqrt{2\pi} e^{ikx}. \quad (8)$$

Now, we take finite-dimensional parameterizations of both u and r by truncating their Fourier representation at the maximum number of modes $|k| \leq k_{\max}$. Thus, we write $\mathbf{R} = (R_{k,l,j}) \in \mathbb{C}^{k_{\max} \times O \times I}$ directly as a complex-valued tensor and similarly $\mathbf{U} = (U_{k,j}) \in \mathbb{C}^{k_{\max} \times I}$. We may assume this, because both the function u and the kernel function r are real-valued, and we want v to be real-valued as well. Therefore, we impose conjugate symmetry on the coefficients of the Fourier series of r and u , i.e. $R_{k,l,j} = \overline{R_{-k,l,j}}$ and $U_{k,j} = \overline{U_{-k,j}}$ where $k \in \mathbb{Z}$, $|k| \leq k_{\max}$ and $j = 1, \dots, I$, $l = 1, \dots, O$. After this heuristic approach, we introduce the following definition.

Definition 2.1 Assume that $k_{\max} \leq \lfloor \frac{n}{2} \rfloor + 1$ and let the Discrete Fourier Neural Operator $\hat{F}_{\mathbf{R}, k_{\max}} : \mathbb{R}^{n \times I} \rightarrow \mathbb{R}^{n \times O}$ defined by the equation

$$\hat{F}_{\mathbf{R}, k_{\max}}(\mathbf{U}) = \hat{\mathcal{F}}^{-1}(\mathbf{R} \cdot \hat{\mathcal{F}}(\mathbf{U})) \in \mathbb{R}^{n \times O},$$

where $\mathbf{R} \in \mathbb{C}^{k_{\max} \times O \times I}$, $\mathbf{U} \in \mathbb{R}^{n \times I}$, $\hat{\mathcal{F}}(\mathbf{U}) \in \mathbb{C}^{\lfloor \frac{n}{2} \rfloor + 1 \times I}$ and $\hat{\mathcal{F}}$ denotes the one-dimensional Fast Fourier Transform for real-valued functions (RFFT). We calculate the product $\mathbf{R} \cdot \hat{\mathcal{F}}(\mathbf{U})$ in the way we truncate the higher order modes in $\hat{\mathcal{F}}(\mathbf{U})$ and obtain $\hat{\mathcal{F}}(\mathbf{U}) \in \mathbb{C}^{k_{\max} \times I}$ then we can apply the straightforward modification of equation (8), i.e.

$$\left(\mathbf{R} \cdot \left(\hat{\mathcal{F}}(\mathbf{U})\right)\right)_{k,l} = \sum_{j=1}^I R_{k,l,j} \left(\hat{\mathcal{F}}(\mathbf{U})\right)_{k,j},$$

where $k = 1, \dots, k_{\max}$, $l = 1, \dots, O$. The product $\mathbf{R} \cdot \hat{\mathcal{F}}(\mathbf{U}) \in \mathbb{C}^{k_{\max} \times O}$ must be padded with zeros in order to bring it to the appropriate shape, i.e. $\mathbf{R} \cdot \hat{\mathcal{F}}(\mathbf{U}) \in \mathbb{C}^{\lfloor \frac{n}{2} \rfloor + 1 \times O}$ before applying the inverse of the RFFT operator.

Remark: we use the Fourier Neural Operator expression for the Discrete Fourier Neural Operator in the following.

2.2 CONVOLUTIONAL FOURIER LAYER

We use the Discrete Fourier Neural Operator in our proposed model in a very similar way to usual convolution layers. The layer corresponding to the operator has three parameters: the number of the input channels I , the number of the output channels O , and the size of the kernel truncation k_{\max} . In the network architecture, we refer to a Fourier layer as the parallel coupling of a layer corresponding to a Discrete Fourier Neural Operator and a conventional one-dimensional convolution layer with the corresponding padding size to get the same output size, see Figure 1. The motivation behind this coupling is the following. Filters in convolutional neural networks are usually local. They are good at capturing local patterns. The filters in the Fourier Neural Operator are global functions, therefore we expect they are good at capturing global patterns.

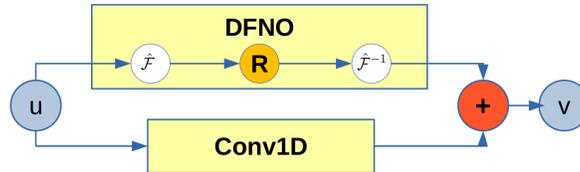


Figure 1: The Convolutional Fourier layer is a parallel coupling of a layer corresponding to the Discrete Fourier Neural Operator and a conventional one-dimensional convolution layer with the corresponding padding size.

2.3 PROPOSED *CF-ResNet-9-1D* MODEL

The model we propose is a simple *ResNet*-type model (He et al., 2016), we call it *CF-ResNet-9-1D*. The total number of trainable parameters is 2.6M-8M. Essentially, we have replaced only the one-dimensional convolution layers with Fourier layers in the simplest *ResNet-9* model, thus obtaining a smaller and larger network, see Figure 2. More technical information about the network is coming now with some explanations for the notations. We use Gaussian Error Linear Units (GELU) as activation functions (Hendrycks & Gimpel, 2016). The further abbreviations on Figure 2 are:

- $\text{FL}(I, O)$: Fourier Layer with I input channels and with O output channels,
- GELU: The GELU activation function
- $\text{BN}(I)$: Batch normalization in I channels,

- $FC(I, O)$: Fully connected layer between I and O neurons.
- `avgpool, 2`: Performs average pooling operation on the input by the kernel size 2, this is also the size of the stride here.
- `AdaptiveAvgPool1d(1)`: Performs adaptive average pooling operation, in this case, the size of the output is 1 by each channel.
- `num_classes`: number of classes at the output layer.

Other important parameters of the layers are:

- The size of the truncation in the Fourier Layers are the same with the parameter $k_{\max} = 16$.
- The size of the kernel is 11 and the size of the padding is 5 in the one-dimensional convolution layers.

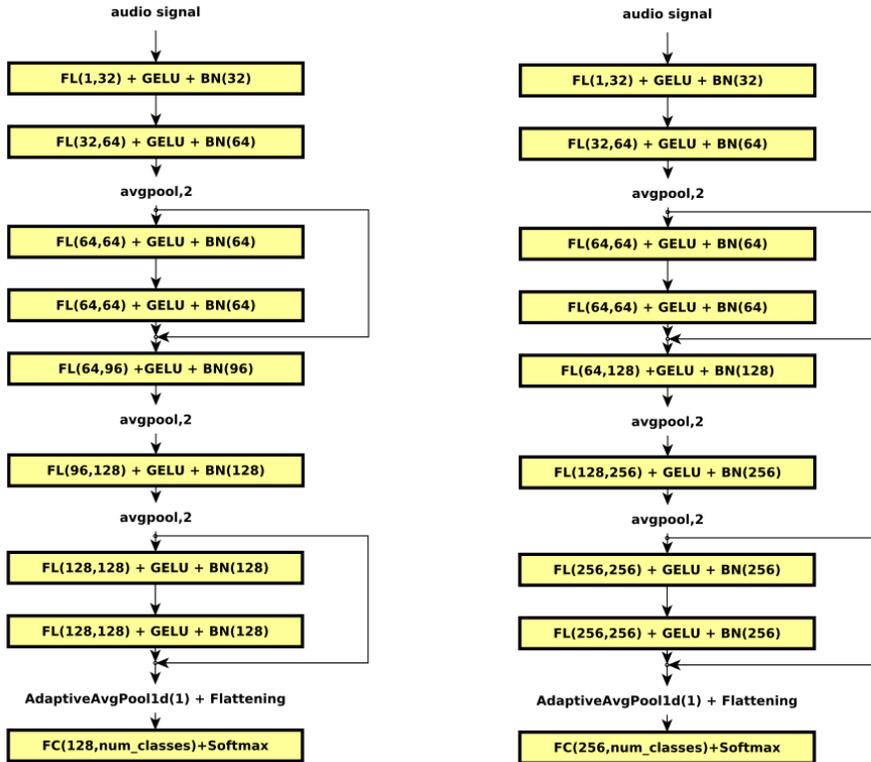


Figure 2: The architecture of the proposed small and large *CF-ResNet-9-ID* models.

Furthermore, the two sideways arrows in the figure represent the residual connections.

2.4 DATASETS

This study utilizes four publicly available dataset whose names are *Wingbeats* (Potamitis & Rigakis, 2016), *Fruitflies* (Rigakis et al., 2019), *Insects* (Chen et al., 2014b) and *Abuzz* (Mukundarajan et al., 2017).

The *Wingbeats* and *Abuzz* datasets contain raw audio signals of six mosquito species of three different genera, namely these are *Ae. aegypti*, *Ae. albopictus*, *An. arabiensis*, *An. gambiae*, *Cu. pipiens*, and *Cu. quinquefasciatus*. The data were collected individually at the premises of Biogents (Regensburg, Germany) and recorded by large aperture optoelectronic devices in the *Wingbeats* dataset. Each recorded audio sound is 0.65s long or has a length of 5000 samples at 8KHz sampling rate.

The total number of records is 279556. The dataset is imbalanced, the details can be seen in Table 5.

The *Fruitflies* dataset was collected from Gouves and Chersonisos areas in Crete using the same collection method as the *Wingbeats* dataset. It comprises audio signals from three distinct species of fruit flies: *Drosophila melanogaster*, *Drosophila suzukii*, and *Zaprionus*. The signals were sampled at 8kHz. The dataset contains 34,518 recordings, each lasting 0.65 seconds. The class sizes are shown in Appendix, Table 6.

The *Insects* dataset was generated by the UCR Computational Entomology Group (Chen et al., 2014b) collected in a similar manner using pseudo-acoustic optical sensors. It contains 10 classes, namely *Aedes aegypti (female)*, *Aedes aegypti (male)*, *Drosophila simulans*, *Musca domestica*, *Cx. quinquefasciatus (female)*, *Cx. quinquefasciatus (male)*, *Cx. stigmatosoma (female)*, *Cx. stigmatosoma (male)*, *Cx. tarsalis (female)* and *Cx. tarsalis (male)*. The dataset contains a total of 50,000 recordings, each 0.1s long sampled at 6kHz, and each class consists of 5,000 elements.

In the *Abuzz* dataset the data were collected by mobile phones. Originally, the length of the recordings varies up to 5 min, with sample rates of 8000Hz and 44,100Hz. Here, we use the pre-processed dataset from the article (Wei et al., 2022). This means that we have 10s long signals. The total amount of recording is 915, and the element numbers of each class can be seen in Appendix, Table 5. During the numerical experiments, we converted each signal at 8000Hz sampling rate.

2.5 DATA PREPROCESSING

In all scenarios, we consistently utilize raw audio signals. We aim to compare our results fairly with those presented by Wei et al. (2022). Following the methodology Fanioudakis et al. (2018), we partition the dataset exactly the same way into training and testing sets.

The learning set contains 80% of the data and it is further divided into training and validation sets in a stratified fashion for cross-validation. This means that we consider a 60/20/20% split for the training/validation/test sets, the exact sizes of the sets can be seen in Table 1. Then, we normalize the data as simply as possible, which means that the entire dataset is considered as the measured data of one variable. Therefore, we perform simple standardization by calculating the two required scalars, the mean and the standard deviation of the training set.

Table 1: Sizes of the training, validation, and testing sets.

Dataset	Training	Validation	Testing
<i>Wingbeats</i>	167,739	55,914	55,913
<i>Abuzz</i>	549	184	182
<i>Fruitflies</i>	20,710	6,904	6,904
<i>Insects</i>	30,000	10,000	10,000

2.6 TRAINING

A stochastic Gradient optimizer is used with Nesterov momentum during the training. We also use the One-cycle learning rate scheduler (Smith & Topin, 2017). The setting and choosing of the learning rate scheduler was also a crucial part of our work.

The further parameter settings for the training can be seen in Table 2. These parameters were chosen by hand among evaluating the model with different settings on the validation set, to avoid overfitting and underfitting. This is also true for the selection of the optimizer. As the evaluation metric on the test set, we use accuracy. More precisely, this means that the model with the best validation accuracies during training was saved and then evaluated on the test set. All models were implemented on *NVIDIA GeForce RTX 3090 GPU* of 24GB of memory, using *Python3.8* with supported libraries of *PyTorch*, *Librosa*, *Pandas*, and *NumPy*.

Table 2: Parameter settings in the training processes.

Dataset	Batch size	Epochs	Learning rate	Weight decay
<i>Wingbeats</i>	32	25	0.0005	0.005
<i>Abuzz</i>	4	150	0.0005	0.005
<i>Fruitflies</i>	32	25	0.0002	0.005
<i>Insects</i>	32	25	0.0005	0.005

3 RESULTS

We found that our proposed *CF-ResNet-9-ID* models have outperformed other models on the *Wingbeats*, the *Fruitflies* and the *Insects* datasets, i.e. the *WbNet* (Wei et al., 2022) and even *DenseNet-121*, *MobileNet* models (Fanioudakis et al., 2018). We summarize these results in Table 3, the two largest values are written in bold in each column.

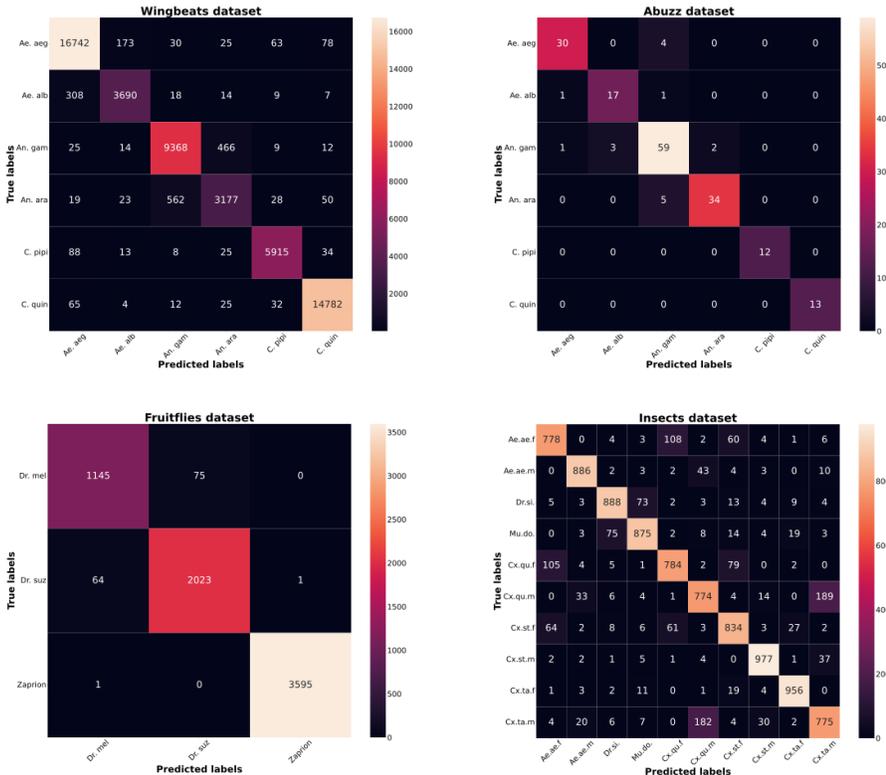


Figure 3: Confusion matrices on the testing datasets for the large *CF-ResNet-9-ID* model which has the best validation accuracy among 5 different runs.

The proposed model was also evaluated without the Fourier layer in the numerical experiments as well, so that to substantiate the utility of the Fourier layer. To be more specific, the numerical experiments were performed using both small (with 670K trainable parameters) and large (with 8.0M parameters) *ResNet-9* models, which include one-dimensional convolutional layers. The specifics of these *ResNet-9* models can be found in the figure 5 included in the Appendix.

The small and large *CF-ResNet-9-ID* models achieved 95.87% and 96.01% average test accuracy on the *Wingbeats* dataset among five different runs. The average test accuracies achieved by the

Table 3: Accuracies for different architectures. The abbreviations stand for the following: *TP* - the number of the trainable parameters, *BVA* - the best validation accuracy over five different runs, *TA* - the corresponding test accuracy, *ATA* - average test accuracy, *RS* - raw samples, *SP* - spectrogram.

WINGBEATS					
Architecture	TP	Data	BVA %	TA %	ATA %
small <i>CF-ResNet-9-ID</i> (ours)	2.6M	RS	96.08	95.92	95.87
large <i>CF-ResNet-9-ID</i> (ours)	7.8M	RS	96.12	95.99	95.98
small vanilla <i>FNO-ResNet-9</i>	2.0M	RS	88.24	88.27	88.25
large vanilla <i>FNO-ResNet-9</i>	6.0M	RS	88.77	88.61	88.53
small <i>ResNet-9</i> ¹	0.7M	RS	95.55	95.43	95.37
large <i>ResNet-9</i> ¹	8M	RS	95.65	95.35	95.43
<i>DenseNet121</i> (Fanioudakis et al., 2018) ²	7M	SP	92.16	91.97	91.92
<i>MobileNet</i> (Fanioudakis et al., 2018) ²	2M	SP	91.41	91.14	91.20
<i>WbNet</i> (Wei et al., 2022) ²	11M	SP	87.62	91.09	90.32
ABUZZ					
Architecture	TP	Data	BVA %	TA %	ATA %
small <i>CF-ResNet-9-ID</i> (ours)	2.6M	RS	93.48	86.26	85.05
large <i>CF-ResNet-9-ID</i> (ours)	7.8M	RS	95.11	90.66	86.59
small vanilla <i>FNO-ResNet-9</i>	2.0M	RS	63.04	57.69	52.42
large vanilla <i>FNO-ResNet-9</i>	6.0M	RS	62.50	48.35	54.39
small <i>ResNet-9</i> ¹	0.7M	RS	95.11	87.36	85.49
large <i>ResNet-9</i> ¹	8M	RS	97.28	92.86	90.33
<i>DenseNet121</i> Fanioudakis et al. (2018) ²	7M	SP	99.79	97.09	95.94
<i>MobileNet</i> Fanioudakis et al. (2018) ²	2M	SP	100.00	92.98	93.69
<i>WbNet</i> Wei et al. (2022) ²	11M	SP	62.50	73.63	67.25
FRUITFLIES					
Architecture	TP	Data	BVA %	TA %	ATA %
small <i>CF-ResNet-9-ID</i> (ours)	2.6M	RS	98.49	97.99	97.96
large <i>CF-ResNet-9-ID</i> (ours)	7.8M	RS	98.49	97.96	98.00
small vanilla <i>FNO-ResNet-9</i>	2.0M	RS	93.60	93.86	93.45
large vanilla <i>FNO-ResNet-9</i>	6.0M	RS	93.63	93.61	93.53
small <i>ResNet-9</i> ¹	0.7M	RS	98.16	97.75	97.64
large <i>ResNet-9</i> ¹	8M	RS	98.32	97.71	97.67
<i>DenseNet121</i> Fanioudakis et al. (2018) ²	7M	SP	92.65	92.68	92.91
<i>MobileNet</i> Fanioudakis et al. (2018) ²	2M	SP	91.73	91.01	91.57
<i>WbNet</i> Wei et al. (2022) ²	11M	SP	86.37	86.15	86.67
INSECTS					
Architecture	TP	Data	BVA %	TA %	ATA %
small <i>CF-ResNet-9-ID</i> (ours)	2.6M	RS	85.27	85.44	85.10
large <i>CF-ResNet-9-ID</i> (ours)	7.8M	RS	85.31	85.27	85.24
small vanilla <i>FNO-ResNet-9</i>	2.0M	RS	75.59	74.98	74.81
large vanilla <i>FNO-ResNet-9</i>	6.0M	RS	76.00	75.62	75.28
small <i>ResNet-9</i> ¹	0.7M	RS	84.91	84.63	84.73
large <i>ResNet-9</i> ¹	8M	RS	85.26	85.18	85.16
<i>DenseNet121</i> Fanioudakis et al. (2018) ²	7M	SP	81.77	81.59	81.36
<i>MobileNet</i> Fanioudakis et al. (2018) ²	2M	SP	79.25	79.28	78.79
<i>WbNet</i> Wei et al. (2022) ²	11M	SP	76.29	78.11	77.91

large *CF-ResNet-9-ID* model were 85.05%, 98.00% and 85.24% on the *Abuzz*, *Fruitflies* and on the *Insects* datasets respectively.

¹According to another accepted article by the authors.

²Reevaluated experiment solving data leak in the original implementation.

These five results were obtained by saving the model with the best validation accuracy during the training process and then evaluating them on the test set. Importantly, while *CF-ResNet-9-1D* does not achieve the highest average test accuracy on the *Abuzz* dataset, it outperforms the other models on the remaining two datasets. The evaluation on the validation set was performed on every 1000 training minibatches in each epoch for the *Wingbeats* dataset, while for the *Abuzz*, the *Fruitflies* and the *Insects* datasets, the model was evaluated after every 30, 100 and 200 minibatches, respectively. The evolution of the accuracies on the training and the validation sets is shown in Figures 4.

The exact details about the test results based on each species can be seen in Table 7. These results were obtained from the models with the best validation accuracy. In Figures 3, we present the confusion matrices for the large *CF-ResNet-9-1D* model. It can be observed that the structures are strongly diagonal. The classes are arranged by genus in pairs, i.e. *Anopheles*, *Aedes* and *Culex* for the *Wingbeats* and the *Abuzz* datasets. There are also two species of the same genus in *Fruitflies*, namely *Drosophila*. Most misclassifications occur between species of the same genus. For the *Insects* dataset, the larger *CF-ResNet-1D* network is not significantly more efficient than other models as it is for *Wingbeats* and *Fruitflies*, perhaps the spectral properties of the recordings are not as dominant in this case. Nevertheless, we can see from Table 4 that both training and inference are substantially faster with our proposed *CF-ResNet-1D* models compared to, for example, the second-best performing *ResNet* model. We can also observe from Table 7 that the *CF-ResNet-1D* network tends to overly predict two classes in both cases, specifically the *Cx. quinq. (male)* and *Cx. tarsalis (male)* classes at the expense of accuracy. In the case of the *Insects* dataset, most of the confusion occurred between the *Cx. quinquefasciatus (male)* and *Cx. tarsalis (male)* classes. Further complicating the classification here is the presence of samples from the same species that included both female and male individuals in the dataset.

Table 3 also illustrates that the conventional convolution with a kernel size of 1 in the original approach proposed by Li et al. (2021) is insufficient for time series processing. We refer to this network as vanilla *FNO-ResNet-9* here, in order to clearly distinguish it from the *CF-ResNet-9-1D* architecture we propose. As the results indicate Convolutional Fourier Layers improve significantly on traditional FNO layers.

In general, our proposed model performed better for the *Wingbeats*, *Fruitflies* and *Insects* datasets compared to the *Abuzz* dataset. Of course, this may also be due to the fact that these datasets are significantly larger than the *Abuzz*. It is also important to note that the *Wingbeats*, *Fruitflies* and *Insects* audio recordings were short, only 0.65s and 0.1s long, and captured using advanced audio equipment, while the *Abuzz* data were longer and recorded using mobile devices in noisy environments, varying up to 5 minutes in length. Therefore, *Abuzz* audio recordings were divided into multiple 10s long segments.

4 CONCLUSION

In this study, we presented a novel approach for incorporating the Fourier Neural Operator in Deep Learning models for time series classification, while also providing a definition of the operator more fitting to this task.

We proposed *CF-ResNet-9-1D* a ResNet-like model that consists of Convolutional Fourier (CF) Layers that highly improve on the original Fourier Neural Operators by adding a parallel local convolution. Experimental evaluation is carried out on insect wingbeat sound classification datasets where our models achieve a new state-of-the-art result on the *Wingbeats*, *Fruitflies*, and *Insects* datasets, outperforming previously published models, while also maintaining a competitive inference and training speed. To aid mosquito-borne disease control and intervention efforts we release our implementation and pre-trained models.

As combining FNOs with wide-spread deep learning operators empirically yielded improvements over current state-of-the-art, exploring the effects of data augmentation, and more complex Fourier domain processing, such as adaptive filtering, attention, or Short-Time Fourier Transform-based solutions are promising questions for further research.

5 REPRODUCIBILITY

We set all possible random seeds during our numerical experiments. To account for the variability of random processes we repeat each experiment 5 times and report the best and average scores in Section 3. To retain anonymity the supporting code for our experiments and the open-source datasets are made available as supplementary material only. The camera-ready version will include a public repository with the same material.

6 ETHICS STATEMENT

This research does not include any potentially harmful datasets, methods, or bias, nor any human-related experiments. We declare that the authors have no conflicting interests or funding, that could influence the research procedure or results.

REFERENCES

- Benjamin Arthur, Kevin Emr, Robert Wytenbach, and Ron Hoy. Mosquito (*aedes aegypti*) flight tones: Frequency, harmonicity, spherical spreading, and phase relationships. *The Journal of the Acoustical Society of America*, 135, 01 2014. doi: 10.1121/1.4861233.
- Hector Caraballo and Kevin King. Emergency department management of mosquito-borne illness: malaria, dengue, and west nile virus. *Emergency medicine practice*, 16(5):1—23; quiz 23—4, May 2014. ISSN 1524-1971. URL <http://europepmc.org/abstract/MED/25207355>.
- Xiao Chaozhao, Xiaoqian Zhu, Xiaoqun Cao, Fukang Yin, Jun Nie, and Fujia Hu. Pa_casualstm: A new time series prediction network with the physical constraint and adjusted fourier neural operator for the time-dependent partial differential equation. *Frontiers in Physics*, 10, 09 2022. doi: 10.3389/fphy.2022.1004417.
- Yanping Chen, Adena Why, Gustavo Batista, Agenor Mafra-Neto, and Eamonn Keogh. Flying insect classification with inexpensive sensors. *Journal of Insect Behavior*, 27, 03 2014a. doi: 10.1007/s10905-014-9454-4.
- Yanping Chen, Adena Why, Gustavo Batista, Agenor Mafra-Neto, and Eamonn Keogh. Flying insect classification with inexpensive sensors. *Journal of Insect Behavior*, 27, 03 2014b. doi: 10.1007/s10905-014-9454-4.
- Eleftherios Fanioudakis, Matthias Geismar, and Ilyas Potamitis. Mosquito wingbeat analysis and classification using deep learning. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 2410–2414, 2018. doi: 10.23919/EUSIPCO.2018.8553542.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.
- Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs). *arXiv e-prints*, art. arXiv:1606.08415, June 2016.
- Williamson Johnny, Hatzinakis Brigido, Marcelo Ladeira, and Joao Carlos Felix Souza. Fourier neural operator for image classification. In *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–6, 2022. doi: 10.23919/CISTI54924.2022.9820128.
- Samira Kabri, Tim Roith, Daniel Tenbrinck, and Martin Burger. Resolution-invariant image classification based on fourier neural operators. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 236–249. Springer, 2023.

- Veerayuth Kittichai, Theerakamol Pengsakul, Kemmapon Chumchuen, Yudthana Samung, Patchara Sriwichai, Natthaphop Phatthamolrat, Teerawat Tongloy, Komgrit Jaksukam, Santhad Chuwongin, and Boonsang S. Deep learning approaches for challenging species and gender identification of mosquito vectors. *Scientific Reports*, 11, 03 2021. doi: 10.1038/s41598-021-84219-4.
- Zhijie Li, Wenhui Peng, Zelong Yuan, and Jianchun Wang. Fourier neural operator approach to large eddy simulation of three-dimensional turbulence. *Theoretical and Applied Mechanics Letters*, 12(6):100389, 2022. ISSN 2095-0349. doi: <https://doi.org/10.1016/j.taml.2022.100389>. URL <https://www.sciencedirect.com/science/article/pii/S2095034922000691>.
- Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021. URL <https://openreview.net/forum?id=c8P9NQVtmnO>.
- Jose Luna-Gonzalez, Daniel Robles-Camarillo, M. Nakano-Miyatake, Humberto Lanz-Mendoza, and Hector Perez-Meana. *A CNN-Based Mosquito Classification Using Image Transformation of Wingbeat Features*. 09 2020. ISBN 9781643681146. doi: 10.3233/FAIA200559.
- Daniel Motta, Álisson Santos, Bandeira, Bruna Aparecida Souza Machado, Otavio Gonçalves Vicente Ribeiro-Filho, Luis Octavio Arriaga Camargo, Matias Alejandro Valdenegro-Toro, Frank Kirchner, and Roberto Badaró. Optimization of convolutional neural network hyperparameters for automatic classification of adult mosquitoes. *PLOS ONE*, 15(7):1–30, 07 2020. doi: 10.1371/journal.pone.0234959. URL <https://doi.org/10.1371/journal.pone.0234959>.
- Haripriya Mukundarajan, Felix Jan Hein Hol, Erica Araceli Castillo, Cooper Newby, and Manu Prakash. Using mobile phones as acoustic sensors for high-throughput mosquito surveillance. *eLife*, 6:e27854, oct 2017. ISSN 2050-084X. doi: 10.7554/eLife.27854. URL <https://doi.org/10.7554/eLife.27854>.
- Wm. H. Offenhauser and Morton Charles Kahn. The sounds of disease-carrying mosquitoes. *Journal of the Acoustical Society of America*, 21:462–463, 1949.
- Kazushige Okayasu, Kota Yoshida, Masataka Fuchida, and Akio Nakamura. Vision-based classification of mosquito species: Comparison of conventional and deep learning methods. *Applied Sciences*, 9(18), 2019. ISSN 2076-3417. doi: 10.3390/app9183935. URL <https://www.mdpi.com/2076-3417/9/18/3935>.
- Tai-Hsien Ouyang, En-Cheng Yang, Joe-Air Jiang, and Ta-Te Lin. Mosquito vector monitoring system based on optical wingbeat classification. *Computers and Electronics in Agriculture*, 118: 47–55, 10 2015. doi: 10.1016/j.compag.2015.08.021.
- John Palmer, Aitana Oltra, Francisco Collantes, Juan Delgado, Javier Lucientes, Sarah Delacour, Mikel Bengoa, Roger Eritja, and Frederic Bartumeus. Citizen science provides a reliable and scalable tool to track disease-carrying mosquitoes. *Nature Communications*, 8, 10 2017. doi: 10.1038/s41467-017-00914-9.
- Ilyas Potamitis and Iraklis I. Rigakis. Large aperture optoelectronic devices to record and time-stamp insects’ wingbeats. *IEEE Sensors Journal*, 16:6053–6061, 2016. doi: 10.1109/JSEN.2016.2574762.
- Iraklis Rigakis, Ilyas Potamitis, Nicolaos-Alexandros Tatlas, Ioannis Livadaras, and Stavros Ntalampiras. A multispectral backscattered light recorder of insects’ wingbeats. *Electronics*, 8(3), 2019. ISSN 2079-9292. doi: 10.3390/electronics8030277. URL <https://www.mdpi.com/2079-9292/8/3/277>.
- Leslie N. Smith and Nicholay Topin. Super-convergence: very fast training of neural networks using large learning rates. In *Defense + Commercial Sensing*, 2017. doi: 10.1117/12.2520589.

Dinarte Vasconcelos, Nuno Nunes, Miguel Ribeiro, Catia Prandi, and Alex Rogers. Locomobis: a low-cost acoustic-based sensing system to monitor and classify mosquitoes. pp. 1–6, 01 2019. doi: 10.1109/CCNC.2019.8651767.

Xutong Wei, Md Hossain, and Khandaker Asif Ahmed. A resnet attention model for classifying mosquitoes from wing-beating sounds. *Scientific Reports*, 12:10334, 06 2022. doi: 10.1038/s41598-022-14372-x.

Myat Su Yin, Peter Haddawy, Borvorntat Nirandmongkol, Tup Kongthaworn, Chanaporn Chaisumritchoke, Akara Supratak, Chaitawat Sa-ngamuang, and Patchara Sriwichai. A lightweight deep learning approach to mosquito classification from wingbeat sounds. In *Proceedings of the Conference on Information Technology for Social Good, GoodIT '21*, pp. 37–42, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384780. doi: 10.1145/3462203.3475908. URL <https://doi.org/10.1145/3462203.3475908>.

Kai Zhang, Yuande Zuo, Hanjun Zhao, Xiaopeng Ma, Jianwei Gu, Jian Wang, Yongfei Yang, Chuanjin Yao, and Jun Yao. Fourier Neural Operator for Solving Subsurface Oil/Water Two-Phase Flow Partial Differential Equation. *SPE Journal*, 27(03):1815–1830, 06 2022. ISSN 1086-055X. doi: 10.2118/209223-PA. URL <https://doi.org/10.2118/209223-PA>.

A APPENDIX

Table 4: Sample processing speeds of different architectures using the largest power of two batch sizes which fit into 24GB memory. *TP* denotes the number of the trainable parameters, *BS* marks the batch size, *TTPS* is training time per sample in milliseconds, *ITPS* denotes inference time per sample in milliseconds, *RS* means raw samples and *SP* shortens spectrogram. Processing speeds were averaged over a full epoch of training and validation.

<i>WINGBEATS</i>					
Architecture	<i>TP</i>	Features	<i>BS</i>	<i>TTPS</i>	<i>ITPS</i>
small <i>CF-ResNet-9-ID</i> (ours)	2.6M	<i>RS</i>	512	0.72	0.21
large <i>CF-ResNet-9-ID</i> (ours)	7.8M	<i>RS</i>	512	0.92	0.29
small <i>ResNet-9</i> ³	0.7M	<i>RS</i>	512	0.45	0.14
large <i>ResNet-9</i> ³	8M	<i>RS</i>	256	1.83	0.43
<i>DenseNet121</i> Fanioudakis et al. (2018) ⁴	7M	<i>SP</i>	256	0.85	0.30
<i>MobileNet</i> Fanioudakis et al. (2018) ⁴	2M	<i>SP</i>	512	0.38	0.14
<i>WbNet</i> Wei et al. (2022) ⁴	11M	<i>SP</i>	128	1.20	0.38
<i>ABUZZ</i>					
Architecture	<i>TP</i>	Features	<i>BS</i>	<i>TTPS</i>	<i>ITPS</i>
small <i>CF-ResNet-9-ID</i> (ours)	2.6M	<i>RS</i>	64	13.42	3.84
large <i>CF-ResNet-9-ID</i> (ours)	7.8M	<i>RS</i>	32	17.78	5.53
small <i>ResNet-9</i> ³	0.7M	<i>RS</i>	64	3.00	1.08
large <i>ResNet-9</i> ³	8M	<i>RS</i>	32	20.97	5.11
<i>DenseNet121</i> Fanioudakis et al. (2018) ⁴	7M	<i>SP</i>	32	13.08	4.20
<i>MobileNet</i> Fanioudakis et al. (2018) ⁴	2M	<i>SP</i>	32	5.40	1.86
<i>WbNet</i> Wei et al. (2022) ⁴	11M	<i>SP</i>	8	22.04	5.11
<i>FRUITFLIES</i>					
Architecture	<i>TP</i>	Features	<i>BS</i>	<i>TTPS</i>	<i>ITPS</i>
small <i>CF-ResNet-9-ID</i> (ours)	0.7M	<i>RS</i>	512	0.69	0.21
large <i>CF-ResNet-9-ID</i> (ours)	8M	<i>RS</i>	512	0.96	0.28
small <i>ResNet-9</i> ³	0.7M	<i>RS</i>	512	0.44	0.14
large <i>ResNet-9</i> ³	8M	<i>RS</i>	256	1.92	0.40
<i>DenseNet121</i> Fanioudakis et al. (2018) ⁴	7M	<i>SP</i>	256	1.67	0.56
<i>MobileNet</i> Fanioudakis et al. (2018) ⁴	2M	<i>SP</i>	512	0.37	0.13
<i>WbNet</i> Wei et al. (2022) ⁴	11M	<i>SP</i>	128	1.22	0.42
<i>INSECTS</i>					
Architecture	<i>TP</i>	Features	<i>BS</i>	<i>TTPS</i>	<i>ITPS</i>
small <i>CF-ResNet-9-ID</i> (ours)	2.6M	<i>RS</i>	8192	0.13	0.03
large <i>CF-ResNet-9-ID</i> (ours)	7.8M	<i>RS</i>	4096	0.16	0.04
small <i>ResNet-9</i> ³	0.7M	<i>RS</i>	8192	0.08	0.04
large <i>ResNet-9</i> ³	8M	<i>RS</i>	2048	0.24	0.07
<i>DenseNet121</i> Fanioudakis et al. (2018) ⁴	7M	<i>SP</i>	512	0.57	0.18
<i>MobileNet</i> Fanioudakis et al. (2018) ⁴	2M	<i>SP</i>	4096	0.09	0.03
<i>WbNet</i> Wei et al. (2022) ⁴	11M	<i>SP</i>	128	13.24	0.46

³According to another accepted article by the authors.

⁴Reevaluated experiment solving data leak in the original implementation.

Table 5: Element numbers of each class in the *Wingbeats* and in the *Fruitflies* datasets.

Species	Wingbeats	Abuzz
<i>Ae. aegypti</i>	85553	324
<i>Ae. Albopictus</i>	20231	197
<i>An. Gambiae</i>	49471	171
<i>An. Arabiensis</i>	19297	95
<i>Cu. pipiens</i>	30415	66
<i>Cu. quinquefasciatus</i>	74599	62

Table 6: Element numbers of each class in the *Fruitflies* dataset.

Species	Fruitflies
<i>Dr. melanogaster</i>	6,064
<i>Dr. suzukii</i>	10,142
<i>Zaprionus</i>	18,312

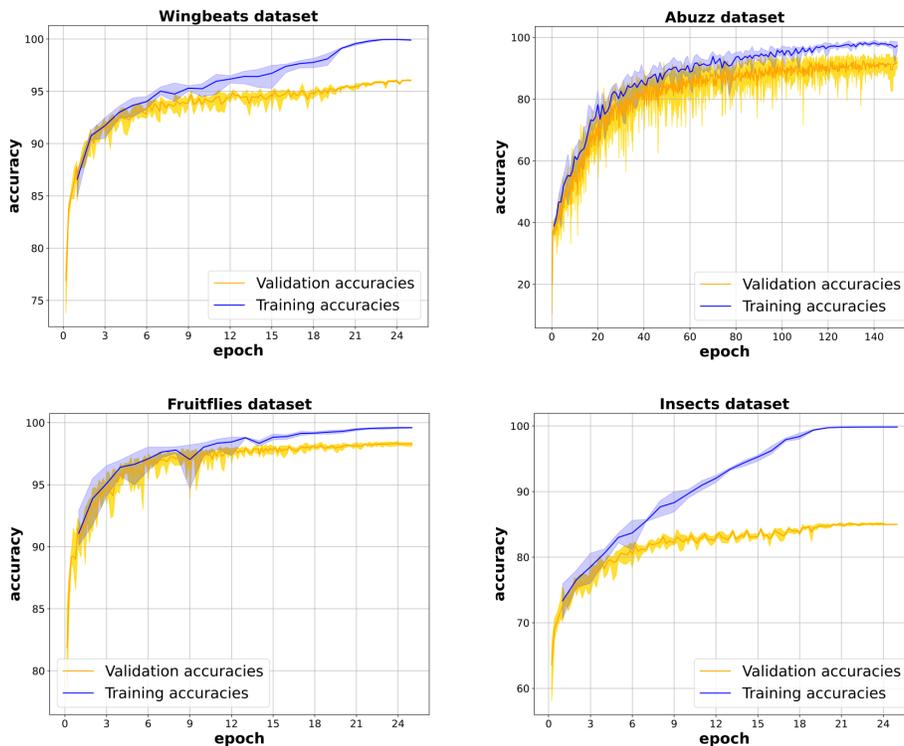
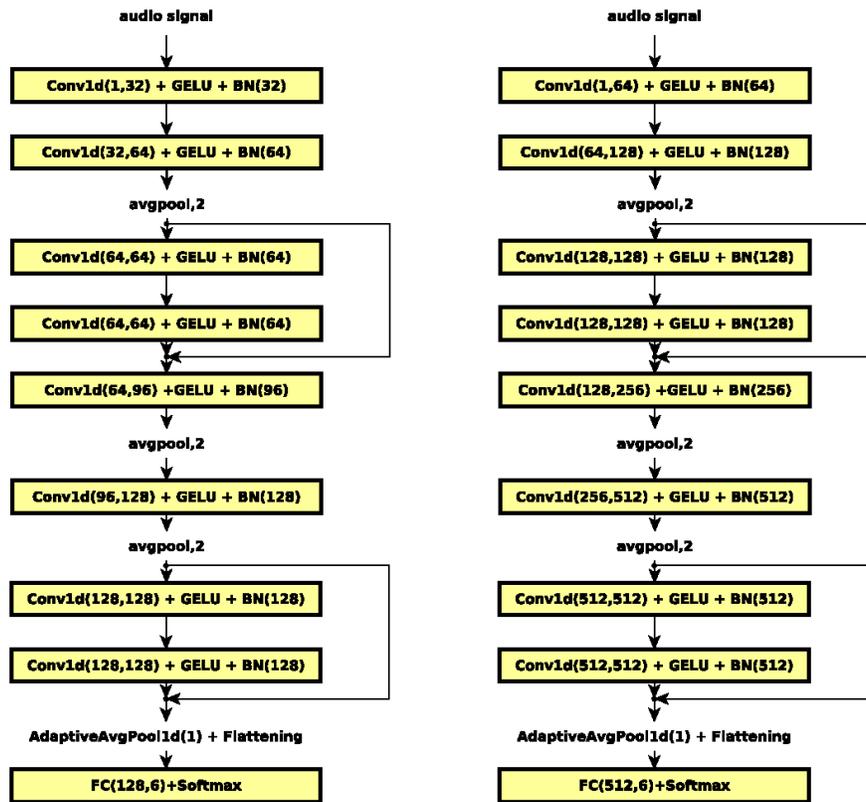
Figure 4: Performance on the different datasets for the accuracy generated from 5 independent runs by the large *CF-ResNet-9-1D* model. The shaded region is enclosed between the maximum and minimum values over the runs, while the boldface curve displays the average.

Table 7: The *CF-ResNet-9-ID* models evaluation metrics on the different datasets.

<i>WINGBEATS</i>						
Species	Small <i>CF-ResNet-9-ID</i>			Large <i>CF-ResNet-9-ID</i>		
	Precision	Recall	F1-score	Precision	Recall	F1-score
<i>Ae. aegypti</i>	97.07	97.83	97.45	97.07	97.84	97.45
<i>Ae. Albopictus</i>	94.28	91.32	92.78	94.20	91.20	92.68
<i>An. Gambiae</i>	93.18	94.55	93.91	93.70	94.68	94.19
<i>An. Arabiensis</i>	84.41	81.24	82.79	85.13	82.33	83.70
<i>Cu. pipiens</i>	97.67	97.25	97.46	97.66	97.22	97.44
<i>Cu. quinq.</i>	98.92	99.12	99.02	98.78	99.07	98.93
<i>ABUZZ</i>						
Species	Small <i>CF-ResNet-9-ID</i>			Large <i>CF-ResNet-9-ID</i>		
	Precision	Recall	F1-score	Precision	Recall	F1-score
<i>Ae. aegypti</i>	88.24	88.24	88.24	93.75	88.24	90.91
<i>Ae. Albopictus</i>	77.78	73.68	75.68	85.00	89.47	87.18
<i>An. Gambiae</i>	81.16	86.15	83.58	85.51	90.77	88.06
<i>An. Arabiensis</i>	88.89	82.05	85.33	94.44	87.18	90.67
<i>Cu. pipiens</i>	100	100	100	100	100	100
<i>Cu. quinq.</i>	100	100	100	100	100	100
<i>FRUITFLIES</i>						
Species	Small <i>CF-ResNet-9-ID</i>			Large <i>CF-ResNet-9-ID</i>		
	Precision	Recall	F1-score	Precision	Recall	F1-score
<i>Dr. melanog.</i>	94.20	94.59	94.40	94.63	93.85	94.24
<i>Dr. suzukii</i>	96.83	96.60	96.72	96.43	96.89	96.66
<i>Zaprionus</i>	99.94	99.94	99.94	99.97	99.97	99.97
<i>INSECTS</i>						
Species	Small <i>CF-ResNet-9-ID</i>			Large <i>CF-ResNet-9-ID</i>		
	Precision	Recall	F1-score	Precision	Recall	F1-score
<i>Ae. aegypti (female)</i>	81.53	81.78	81.65	81.13	80.54	80.83
<i>Ae. aegypti (male)</i>	92.09	92.86	92.48	92.68	92.97	92.82
<i>Drosophila simulans</i>	86.61	87.65	87.13	89.07	88.45	88.76
<i>Musca domestica</i>	86.59	87.54	87.06	88.56	87.24	87.90
<i>Cx. quinq. (female)</i>	83.72	78.00	80.76	81.58	79.84	80.70
<i>Cx. quinq. (male)</i>	76.84	78.34	77.58	75.73	75.51	75.62
<i>Cx. stigma. (female)</i>	80.38	84.36	82.32	80.89	82.57	81.72
<i>Cx. stigma. (male)</i>	94.90	93.98	94.44	93.67	94.85	94.26
<i>Cx. tarsalis (female)</i>	96.21	94.18	95.19	94.00	95.89	94.94
<i>Cx. tarsalis (male)</i>	76.46	76.02	76.24	75.54	75.24	75.39

Figure 5: The architecture of the small and large *ResNet-9* models.