Steelpan-specific pitch detection: a dataset and deep learning model

Colin Malloy
University of Victoria
3800 Finnerty Road
Victoria, BC Canada
malloyc@uvic.ca

George Tzanetakis
University of Victoria
3800 Finnerty Road
Victoria, BC Canada
gtzan@cs.uvic.ca

ABSTRACT

The steelpan is a pitched percussion instrument that although generally known by listeners is typically not included in music instrument audio datasets. This means that it is usually underrepresented when training existing data-driven deep learning models for fundamental frequency estimation. Furthermore, the steelpan has complex acoustic properties that make fundamental frequency estimation challenging when using deep learning models for general fundamental frequency estimation that are trained to work with any music instrument. Fundamental frequency estimation or pitch detection is a core task in music information retrieval and it is interesting to explore methods that are tailored to specific instruments and whether they can outperform more general methods. To address this, we present SASS-E, the Steelpan Analysis Sample Set for Evaluation that can be used to train steel-pan specific pitch detection algorithms as well as propose a custom-trained deep learning model for steelpan fundamental frequency estimation. This model outperforms general state-of-the-art methods such as PYIN and CREPE on steelpan audio - even while having significantly fewer parameters and operating on a shorter analysis window. This reduces minimum system latency, allowing for deployment to a real-time system that can be used in live music contexts.

Author Keywords

steelpan, pitch detection, fundamental frequency estimation, convolutional neural network, music information retrieval

CCS Concepts

•**Computing methodologies** → *Neural networks;* •**Applied computing** → **Sound and music computing;** •**Information systems** → *Music retrieval;*

1. INTRODUCTION

The pitch of a melodic instrument is a primary characteristic of a musical sound. Pitch detection, also referred to as fundamental frequency estimation, is an important task for audio processing and analysis. General pitch detection methods, such as PYIN and CREPE, work well in many situations, but there are many tasks for which custom-tailored methods may perform better [13], [7]. In these situations, a dataset of appropriate audio is needed in order to design a solution.

Performing pitch detection on the Caribbean steelpan is a situation in which a custom solution can outperform state-of-the-art methods such as PYIN and CREPE. In this paper we present SASS-E, the Steelpan Audio Sample Set for Evaluation, and propose Steelpan-Pitch, a deep learning model for steelpan pitch detection. The architecture for Steelpan-Pitch is designed to minimize latency so that it can be implemented in realtime processing for live audio. SASS-E is open for public use for training and analysis of steelpan audio. We evaluate Steelpan-Pitch on the test set from SASS-E and compare the results with PYIN and CREPE. The evaluation shows that our instrument-specific solution, Steelpan-Pitch, outperforms the baseline state-of-the-art solutions. To further show that the Steelpan-Pitch algorithm generalizes beyond the steelpan data represented in SASS-E, it is also evaluated on steelpan samples taken from the commercial sample library Andy Narell Steel Pans produced by Ilio. We also conduct an experiment to find the balancing point between minimizing latency and maintaining accuracy. Although Steelpan-Pitch is designed for pitch detection on steelpans, our methodology can be used as a template for developing other custom instrument-specific pitch detectors. With a suitable training dataset the Steelpan-Pitch architecture can be adapted to achieve low latency, and high accuracy pitch detection for other musical instruments.

The paper is structured as follows: Section 2 presents background information on the steelpan and pitch detection. Section 3 presents the details of the SASS-E dataset. Section 4 presents the proposed architecture of Steelpan-Pitch. The experiments and results are discussed in Section 5. Section 6 concludes the paper.

2. BACKGROUND AND RELATED WORK

2.1 The Steelpan

The steelpan was invented and developed in Trinidad and Tobago in the 1930's and 40's. Its precursors were made from old frying pans and biscuit tins, but modern versions are made from 55-gallon oil drums. The family of steelpan instruments comprises about six main voice ranges with many variations within each voice. The tenor steelpan is the highest voiced instrument consisting of a single pan. Several

NIME'23, 31 May–2 June, 2023, Mexico City, Mexico.

Figure 1: A low-C tenor steelpan.

variations of the tenor steelpan exist, but the most common layout arranges the notes in the circle of fifths as in Fig. 2. This note layout is commonly referred to as the "fourths and fifths" or spiderweb layout. In North America, tenor steelpan's typcially have a range of C4-E6 while in Trinidad it is more common for tenor steelpans to have a range of D4-F6. Other tenor steelpan layouts can have different ranges and completely different note placements. Bass steelpans are the lowest voiced instruments in the steelpan family. Different configurations consist of between 6 and 12 pans for a single instrument. Outside of Trinidad, six bass is the most common with a range of Bb1-Eb3.

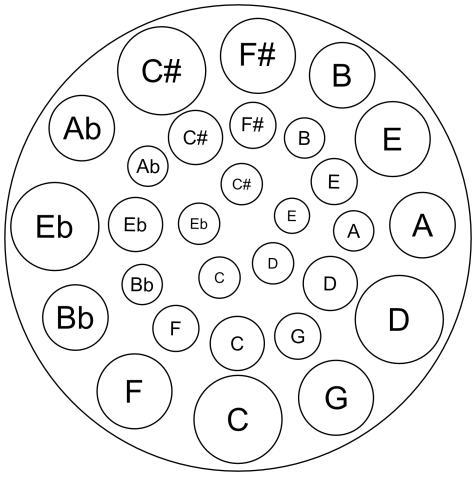

Figure 2: Low C tenor steelpan note layout.

The construction process for steelpans is complex. First, the bottom of the oil drum is sunk through hammering to create a bowl. Then the builder hammers upward on the underside of the bowl to create small convex areas for the individual notes. An "outline" of each note is scored into the metal to help acoustically isolate the notes. The pan is then heat treated and the notes are tuned. The skirt of the oil barrel is also cut to a suitable length for the instrument range – about 10 cm for tenor steelpans to nearly the full oil barrel for bass steelpans. Construction is typically finished by painting or chroming to prevent rusting [15].

The acoustics of the steelpan are complex due to several factors. The notes have flat (or semi-flat) elliptical shapes. All of the notes of a single steelpan are acoustically coupled since they share a common vibrating surface. This coupling causes significant acoustic interference between notes. Striking a note activates nearby notes that are harmonically related. There is also nonsinusoidal motion in the struck note. Due to this, the vibrational behavior of the steelpan is complicated and non-linear [1], [18]. Tuners typically tune a vibrational mode of a note to the second harmonic (an octave above the fundamental). Sometimes a second vibrational mode can be tuned to the third harmonic (an octave and a fifth above the fundamental), but, especially in the high range of the instrument, the higher vibrational modes will vibrate at unrelated partials [19].

2.2 Pitch Detection Methods

The pitch of an audio signal is a perceptual property that generally has a strong relationship to its fundamental frequency (F_0). This relationship is so strong that the two terms are often used interchangeably. As a fundamental property of a signal, pitch detection has received significant attention and many different approaches have been proposed.

There are three main categories of approaches: time-domain, frequency-domain, and data-driven. Most time-domain approaches are based on the autocorrelation function where a signal is correlated with itself at various time lags [21]. Such methods were implemented in digital hardware as early as 1976 [5]. The average magnitude difference function was proposed as a variation on autocorrelation that eliminates multiplication operations by using the absolute value of the difference between the signal and itself at various time delays [17]. The YIN algorithm was proposed as a further refinement of the autocorrelation method that combines it with various methods of error prevention to improve accuracy [4]. Subsequently, PYIN was developed as a probabilistic version of YIN that uses multiple pitch candidates and a Viterbi-decoded hidden Markov model [13]. In contrast to the autocorrelation-based methods, SWIPE compares the input signal's spectrum with the spectra of sawtooth waveforms [3]. The Cepstrum approach uses the cepstrum of a signal (the power spectrum of the logarithm of the power spectrum) to perform pitch detection [16]. In 2018, CREPE (Convolutional REpresentation for Pitch Estimation) was proposed and it was shown that a data-driven machine learning method could outpeform the more traditinal digital signal processing methods that were previously used [7].

Historically, most pitch detection methods generate pitch candidates algorithmically and use heuristics for selecting the final output. The best performing of this group of algorithms was PYIN. In 2018, CREPE presented a new approach to pitch detection by designing a convolutional neural network that takes a raw audio signal as input and outputs a fundamental frequency estimation based on how it was trained. In their landmark paper, Kim et al. show that CREPE significantly outperforms PYIN and SWIPE on the RWC-synth and MedlyDB-stem-synth audio datasets [7]. This confirms that a data-driven approach to pitch detection is a viable method. There have also been preliminary explorations of applying these techniques to the steelpan [10], [11], [12]. Singh et al. showed that the number of network parameters of CREPE could be reduced while simultaneously improving performance by increasing the filter kernel size and using residual blocks [20]. The CREPE architecture and variants of it have been re-used in other audio machine learning contexts and performed well [23].

Figure 3: Sample recording session.

3. SASS-E V1.0

The SASS-E[1] (steelpan audio sample set for evaluation) is a new audio dataset constructed for evaluating music information retrieval tasks on steelpan audio. SASS-E includes samples from three steelpans that are the personal instruments of the first author and were all recently tuned before recording the audio samples. One is a professional quality instrument built by Kyle Dunleavy in the United States. It is a semi-bore, chrome-plated steelpan. This means each note has four small holes drilled around it. Its range is C3-F6 (30 notes). The second instrument is a full-bore, nickel-plated steelpan from Trinidad. The final instrument is a non-bore, painted steelpan of unknown origin.

The three instruments were recorded at different times over the course of several years, but all in the same professional quality recording studio. Microphones used for the sessions included the Earthworks M50 measurement microphone, Beyerdynamic M 160 ribbon microphone, and the Schoeps CMC 6 small diaphragm condenser microphone with MK 4 cardioid capsule. The M 160 and CMC 6 were positioned underneath the bowl of the steelpans while the M50 was positioned above the steelpan. The microphone positions are informed by the recommendations in [14] as well as the author's own extensive experience performing on and recording steelpans. All instruments were recorded using a Focusrite RedNet 4 audio interface with a smaplerate of 96 kHz and bit depth of 24.

Approximately 50 strikes were recorded per note per instrument. The strikes were recorded at dynamic levels varying from pianissimo to fortissimo. The performer attempted to strike the notes in a variety of locations in order to capture all of the possible timbral nuances of each note. Notes in the high range of the steelpan sometimes do not fully activate when struck. Non-activating hits were not included in the dataset.

The samples from the three instruments are mixed in the dataset. The files identify their source note at the beginning of their filename. The filenames are formatted in the form `<MIDI note number>_<set>_<instrument label>_sample_<number>.wav` where `<MIDI note number>` refers to the integer MIDI value of the note's fundamental frequency, `<set>` can be "train", "val", or "test" depending on which pre-set split it belongs to, `<instrument label>` labels the source instrument, and `<number>` assigns a unique number to each sample at a given MIDI note value. The audio samples are pre-split into training, validation, and test sets with 7,931 samples in the training set, 2,680 samples in the validation set, and 2,702 samples in the test set. The dataset contains 13,313 samples for a total of 9 hours and 25 minutes of audio. The audio samples are trimmed so there is minimal leading silence and allows for the full release of each note to ring out with some trailing silence. Depending on the situation, users of the dataset should use automatic trimming on the samples. The shortest audio sample is 1.19 s and the longest is 12.46 s long.

[1] `https://doi.org/10.5281/zenodo.7803316`

4. PROPOSED ARCHITECTURE

The proposed architecture of Steelpan-Pitch[2] is based on the CREPE architecture, but modified to reduce both the number of parameters and the minimum latency of the system. Steelpan-Pitch is a deep convolutional neural network that takes a mono time-domain audio signal as input and produces a pitch estimation as output. Figure 4 shows a block diagram of the proposed architecture. The input is a 128-sample audio frame at a samplerate of 16 kHz (represented on the far left of Fig. 4). There are six convolutional layers and then the network terminates with a fully connected layer. Each triangle in Fig. 4 gives the details for the layers from left to right. The first convolutional layer consists of 512 filters of size 64. The second through fourth layers each consist of 64 filters of size 16, 16, and 8 respectively. The fifth and sixth layers increase the number of filters to 128 and 256 – both of size 4. After every convolutional layer there is a maxpooling layer of size 2. The output of the final convolutional layer is flattened to a 512-dimensional latent representation. The fully connected output layer consists of 360 neurons with sigmoid activations as in [7]. The output vector from the fully connect layer, $\hat{y}$, is used to calculate the final pitch estimate.

Based on [7], the 360 output nodes of the fully connected layer represent frequency bins with 20 cent logarithmic spacing. The 360 bins are denoted as $c_1, c_2, \ldots, c_360$ and span six octaves of audio from note C1 (32.70 Hz) to B7 (1975.5 Hz). The output of the system, $\hat{c}$, is the frequency estimation given by the weighted mean of the bin frequencies by the corresponding values of the output vector, $\hat{\mathbf{y}}$:

$$\hat{c} = \frac{\sum_{i=1}^{360} \hat{y}_i c_i}{\sum_{i=1}^{360} \hat{y}_i}. \tag{1}$$

The pitch estimate, $\hat{f}$, is then given by

$$\hat{f} = f_{\text{ref}} \cdot 2^{\hat{c}/1200} \tag{2}$$

where f_{ref} is the reference frequency of 10 Hz.

The network is trained using Gaussian blurring on the target vectors, $\hat{y}$, to encourage the system to prefer nearly correct predictions when it does make an incorrect prediction [2]. The bin corresponding to the ground truth is given a value of one and then decays over the surrounding bins with a standard deviation of 25 cents according to

$$y_i = \exp\left(-\frac{(c_i - c_{\text{true}})^2}{2 \cdot 25^2}\right). \tag{3}$$

Steelpan-Pitch uses a binary cross entropy loss function to determine the error between the target, $\mathbf{y}$, and the pre-

[2] `https://github.com/malloyca/steelpan-pitch`

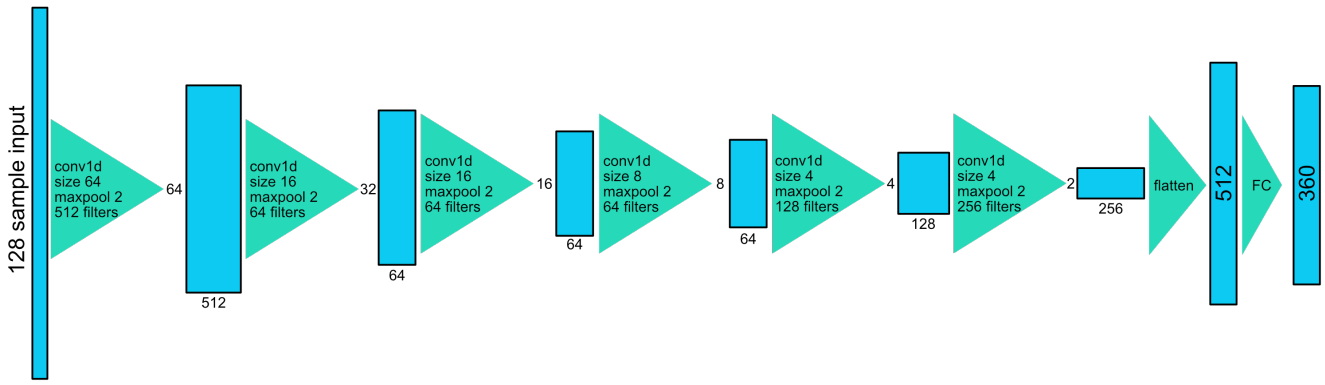

Figure 4: The network architecture of Steelpan-Pitch.

diction, $\hat{\mathbf{y}}$, as in Equation 4.

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i=1}^{360} (-y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)) \quad (4)$$

The model is optimized using the Adam optimizer with a learning rate of 0.0002 [8]. Batch normalization [6] and dropout [22] are used at each convolutional layer as well. The dropout layers have a dropout probability of 0.25. The architecture was implemented in TensorFlow using Keras and trained using Google Colab GPUs.

Although the proposed architecture is based on CREPE, there are some significant differences. One of the most important differences in the architecture is the accepted input signal. CREPE uses a 1,024-sample audio frame as input while our architecture uses a 128-sample frame. Both systems use a samplerate of 16 kHz. Thus when operating in a realtime system, this reduces the minimum latency of the system from 64 ms for 1,024 samples to 8 ms for 128 samples. In the original CREPE architecture, a stride of 4 was used at the first layer and a stride of 1 for each subsequent layer. Since the input frame to Steelpan-Pitch is smaller, it uses a stride of 1 for all layers and instead relies solely on maxpooling for downsampling in the network. The next main difference is the size of the latent representation at the end of the convolutional portion of the network. CREPE has a 2048-dimensional latent space while Steelpan-Pitch has a 512-dimensional latent space. Steelpan-Pitch also uses fewer filters per convolutional layer than CREPE. All of this results in significantly fewer parameters. The proposed architecture for Steelpan-Pitch has 1,009,320 total parameters whereas CREPE has over 22 million.

CREPE also uses a hidden Markov model with the Viterbi algorithm to probabilistically determine the most likely output. This method is the "p" in the PYIN algorithm. However, hidden Markov models are computationally expensive. We chose to omit the hidden Markov model in Steelpan-Pitch's architecture because it is designed to be useable in realtime processing.

5. EXPERIMENTS

5.1 Training Dataset

Steelpan-Pitch was trained solely on the training set from SASS-E. The recording sessions for constructing SASS-E were scheduled for shortly after the steelpans were tuned by a professional steelpan tuner. As such, we make the simplifying assumption that the notes can act as a representative of notes from their respective pitch classes. In this way, we treat this as more of a classification task despite estimating intermediate values at the final output.

The SASS-E dataset was augmented for training using pitch shifting. A copy of each note in the dataset was randomly pitch shifted up or down between 20 cents and 2 semitones in 20 cent increments. Since the steelpan samples in SASS-E are all tuned to integer MIDI values, this allowed us to train the system on intermediate values as well. Augmenting the core SASS-E training set in this way effectively doubles the training set size to nearly 19 hours of audio. Each audio sample has the leading and trailing silence trimmed and is then sliced into 128-sample audio frames (8 ms at 16 kHz samplerate) with a step size of 4 ms. This resulted in 3,521,632 training audio frames and 611,197 validation audio frames.

5.2 Methodology

SASS-E is pre-split into training, validation, and test sets using a 60/20/20 random split. We used these sets for those purposes. All three sets contain instances of every note from every instrument in SASS-E. The danger here will be that the model may overfit the dataset and not generalize to other steelpans. We address this in Section 5.5. The model is evaluated in terms of both raw pitch accuracy (RPA) and raw chroma accuracy (RCA) with 50, 25, and 10 cent thresholds. Raw Pith Accuracy (RPA) measures the percentage of estimations that are within 50, 25, and 10 cents of the target. Raw Chroma Accuracy (RCA) measures the percentage of estimations that are within 50, 25, and 10 cents of a member of the target's pitch class. In other words, RCA accounts for octave errors.

We compare Steelpan-Pitch against CREPE and pYin. These methods are the current state-of-the-art machine learning (CREPE) and time-domain (PYIN) methods available. For reference, PYIN is evaluated with frame lengths of both 1,024 samples and 128 samples. CREPE only operates with a frame length of 1,024 samples while Steelpan-Pitch only operates with a frame length of 128 samples. All three methods are evaluated on the test set from SASS-E.

5.3 Results

Table 1 shows the RPA results for the pitch detection methods on the SASS-E test set. We can see that Steelpan-Pitch significantly outperforms both PYIN (at both frame lengths) and CREPE. At the 50 cent threshold, Steelpan-Pitch beats the next best performer by 22 percentage points. A surprising result here is that PYIN performed somewhat better with the shorter frame length. Table 2 shows the raw

Model	Frame length	Params	RPA Threshold 50 cents	25 cents	10 cents
PYIN	128	-	0.761 ± 0.0009	0.688 ± 0.0010	0.523 ± 0.0012
PYIN	1024	-	0.731 ± 0.0015	0.699 ± 0.0016	0.599 ± 0.0017
CREPE	1024	22.2M	0.738 ± 0.0015	0.727 ± 0.0015	0.626 ± 0.0016
Steelpan-Pitch	128	1M	**0.982 ± 0.0003**	**0.976 ± 0.0004**	**0.948 ± 0.0005**

Table 1: Raw Pitch Accuracies and their standard deviations.

Model	Frame length	Params	RCA Threshold 50 cents	25 cents	10 cents
PYIN	128	-	0.774 ± 0.0001	0.713 ± 0.0010	0.589 ± 0.0011
PYIN	1024	-	0.739 ± 0.0015	0.709 ± 0.0016	0.620 ± 0.0017
CREPE	1024	22.2M	0.773 ± 0.0014	0.761 ± 0.0014	0.668 ± 0.0016
Steelpan-Pitch	128	1M	**0.992 ± 0.0002**	**0.988 ± 0.0003**	**0.970 ± 0.0004**

Table 2: Raw Chroma Accuracies and their standard deviations.

chroma accuracies on the SASS-E test set. In Table 2, we can see similar results as in Table 1. Steelpan-Pitch once again outperforms PYIN and CREPE by over 21 percentage points. An important point to reiterate here is that CREPE and PYIN both use Hidden Markov Models to probabilistically predict the next value.

These results are stark, but also understandable. Steelpan-Pitch was tested on audio samples from the same instruments that it was trained on. In order to show that Steelpan-Pitch has the potential to generalize beyond the training instruments, we present another experiment in Section 5.5.

5.4 Frame Length Comparison

In adapting Steelpan-Pitch from Crepe's architecture, one of the most important changes made was changing the frame length of the input signal from 1,024 samples to 128 samples. The networks are designed to work on audio with a samplerate of 16 kHz. At this samplerate, 1,024 samples is 64 milliseconds. This is well beyond the generally accepted 20 ms threshold for human perception of a signal. If one were to implement CREPE in a realtime situation, the minimum latency would be 64 ms which is readily apparent to human users. In designing Steelpan-Pitch, we decided to work towards making a system that is ready for realtime processing situations by reducing the size of the input frame length while still maintaining a high level of accuracy.

To demonstrate this, we trained different versions of the Steelpan-Pitch architecture on SASS-E. All versions of the system are based on the proposed architecture as in Fig. 4. For each different version, the size of the input layer is changed to a value from [64, 128, 256, 512, 1024] while the rest of the architecture remains the same. The change in input size causes a change in the dimensions at each layer, but the number of filters at each layer, the filter sizes, maxpooling, and other parameters are all kept consistent in order to make the models as comparable as possible. Each of the architecture were then trained on the SASS-E training set with the pitch shifting data augmentation as in 5.1 using appropriately sized audio frames.

The results of the experiment are shown in Table 3. As expected, the longer the input frame length, the better the network performs. However, we can see that there is not a significant reduction in 50-cent raw pitch accuracy until the frame length is reduced to 64 samples. With a 50-cent raw pitch accuracy of 0.982 and raw chroma accuracy of 0.992, a frame length of 128 samples was selected as the optimal balance between latency and accuracy. At the system's 16 kHz samplerate, 128 samples is 8 ms in length which is well within the tolerance of human perception for latency. Table 3 also shows that reducing the frame length can also significantly reduce the number of network parameters. Reducing the frame length from 1,024 samples to 128 results in a 56% reduction in parameters while sacrificing less than two percentage points of 50-cent threshold raw pitch accuracy. A frame length of 64 samples was not selected because, despite further reducing the latency to 4 ms, there is a significant drop in accuracy and only a 4% more reduction in parameters than a 128 sample frame length. Due to these factors we determined a frame length of 128 samples to be the optimal balance between accuracy, number of parameters, and latency.

5.5 Generalization

In order to demonstrate the ability for Steelpan-pitch to generalize to other instruments outside of SASS-E, we also evaluated it on samples recorded from the commercial sample library Andy Narell Steel Pans – The Ellie Mannette Collection produced by Ilio. These samples were recorded at eight velocity levels per note across the entire range of the instrument in Ableton Live at 48 kHz/24 bits and later downsampled to 16 kHz. This set totaled 232 audio samples. Since this is a commercial sample library, we cannot include the audio in SASS-E and only use it to demonstrate Steelpan-Pitch's performance on a steelpan it never analyzed in training.

The results of this test are shown in Table 4.

Although CREPE slightly outperforms Steelpan-Pitch at the RPA 50 cent metrics, Steelpan-Pitch performs the best for the rest of the metric categories. Furthermore, Steelpan-Pitch's performance drops off the significantly less than PYIN's and CREPE's at 10 cent thresholds. The accuracy of Steelpan-Pitch does drop somewhat from its performance on the SASS-E test set. However, the results here show that Steelpan-Pitch does generalize to other steelpans beyond those in SASS-E. Steelpan-Pitch's RCA results in particular are excellent with 95.9% accuracy within 50 cents of the chroma value. This shows a significant portion of Steelpan-Pitch's errors on this instrument are octave errors. This further validates our initial simplifying assumption for generating the training targets. The ability for Steelpan-Pitch to generalize will likely continue to improve as SASS-E is expanded and Steelpan-Pitch is trained on a wider variety of steelpans.

6. CONCLUSION AND FUTURE WORK

Frame length (samples)	Frame length (ms)	Params	Threshold (cents)	RPA	RCA
64	4	0.9 M	50	0.912 ± 0.00065	0.949 ± 0.00051
			25	0.910 ± 0.00067	0.935 ± 0.00057
			10	0.872 ± 0.00086	0.899 ± 0.00070
128	8	1.0 M	50	**0.982 ± 0.00031**	**0.992 ± 0.00021**
			25	**0.976 ± 0.00036**	**0.988 ± 0.00025**
			10	**0.949 ± 0.00052**	**0.970 ± 0.00040**
256	16	1.2 M	50	0.996 ± 0.00015	0.998 ± 0.00010
			25	0.996 ± 0.00016	0.998 ± 0.00011
			10	0.951 ± 0.00051	0.980 ± 0.00034
512	32	1.6 M	50	0.999 ± 0.00008	0.999 ± 0.00007
			25	0.999 ± 0.00008	0.999 ± 0.00007
			10	0.984 ± 0.00030	0.993 ± 0.00020
1024	64	2.3 M	50	0.999 ± 0.00016	0.999 ± 0.00015
			25	0.999 ± 0.00016	0.999 ± 0.00015
			10	0.990 ± 0.00062	0.996 ± 0.00036

Table 3: Comparison of RPA and RCA for different frame lengths (results for the chosen architecture in bold).

Model	Frame Length	Metric	Threshold 50 cents	Threshold 25 cents	Threshold 10 cents
PYIN	128	RPA	0.848 ± 0.0032	0.726 ± 0.0041	0.502 ± 0.0046
		RCA	0.858 ± 0.0032	0.762 ± 0.0039	0.594 ± 0.0044
PYIN	1024	RPA	0.825 ± 0.0097	0.765 ± 0.0108	0.607 ± 0.0125
		RCA	0.825 ± 0.0097	0.767 ± 0.0108	0.650 ± 0.0122
CREPE	1024	RPA	**0.872 ± 0.0046**	0.815 ± 0.0054	0.627 ± 0.0067
		RCA	0.916 ± 0.0039	0.838 ± 0.0051	0.662 ± 0.0066
Steelpan-Pitch	128	RPA	0.862 ± 0.0048	**0.833 ± 0.0052**	**0.723 ± 0.0062**
		RCA	**0.959 ± 0.0028**	**0.945 ± 0.0032**	**0.880 ± 0.0045**

Table 4: RPA and RCA accuracies and standard deviations of PYIN, CREPE, and Steelpan-Pitch on novel steelpan samples.

In this paper we presented a new steelpan-specific data-driven method for pitch detection and a new audio dataset consisting of steelpan audio samples. We show that by limiting the scope of the pitch detection system, it can outperform state-of-the-art systems like CREPE and PYIN despite working on short audio frames with 8 ms of audio. We further show that while the performance does suffer to an extent, Steelpan-Pitch is not simply overfitting to the dataset since it achieves acceptable results on an instance from outside the dataset.

6.1 Future of SASS-E

In the future we plan to expand the SASS-E dataset by adding more instances of tenor steelpans, instances of all members of the steelpan family, and to add more articulation types for all instruments. The tenor steelpan is the main melodic instrument in the steelpan family and the most common, but the other variations should also be represented. Since the acoustics of each different instrument and note layout vary, it will be ideal to include at least 2-3 instances of each type of steelpan as it is added to the dataset. The other primary members of the steelpan family of instruments include (from high to low voice): double tenor, double second, guitar, cello, tenor bass, and bass steelpans.

The type of mallet used to activate a steelpan note can have a drastic effect on the timbre of the instrument [9]. Steelpans are typically played with rubber tipped mallets, but there are many alternatives that are becoming increasingly common in performance such as cardboard tubing, chopsticks, brushes, yarn-wrapped mallets, and bundle rods. The dataset should also be augmented to include samples with as wide a variety of mallets as well.

6.2 Steelpan-Pitch

The accuracy results for Steelpan-Pitch demonstrate that a custom-trained neural net for instrument-specific pitch detector can significantly outperform general pitch methods such as PYIN or CREPE. The next step for Steelpan-Pitch, however, is to further train it on a wider variety of steelpans covering the full range of the steelpan family. This is necessary in order to improve the generalization of the system to all types of steelpans. While the model currently performs well on other instances of tenor steelpans, it is likely that Steelpan-Pitch will not perform well on lower voiced steelpans.

Steelpan-Pitch is trained only on recently tuned instruments, but old, "janky" steelpans have an iconic sound of their own. The overtones are often wildly out of tune from the fundamental, but humans can still generally recognize the pitches. We do not yet have any samples from such instruments in our dataset to evaluate any of the pitch detection methods on, but in all likelihood the accuracy will suffer. With a data-driven approach like in Steelpan-Pitch, we can incorporate these kinds of sounds into the representation when representative data is added to the dataset. After enough instruments of this type have been included, the system should be able to perform well on these instruments as well. Due to the esoteric nature and availability of these instruments, they are also unlikely to gain representation in general audio datasets.

The goal for Steelpan-Pitch is to maintain a well performing, but lightweight model that can be used for realtime pitch detection. The reason for this is that Colin Malloy, one of the authors, is a regular performer of electroacoustic steelpan music. His longterm goal is to build a low latency, realtime steelpan pitch transcription system that can an-

alyze his playing live and perform audio processing tasks based on what he plays.

Steelpan-Pitch is provided as a pre-trained model for analyzing steelpan audio. However, another plan is to extend the system so that another performer can easily customize the model for their specific instrument. In this case, the provided pre-trained weights would be used for transfer learning. The final training for the system would be on audio provided by the performer from their specific audio setup to improve adaptability to other instruments and audio situations.

6.3 Other Future Work

An important question raised in the course of this work, but that is beyond the scope of this research, is in regard to the relationship between a general pitch detector versus an instrument-specific pitch detector. How does the trade off between generalization and instrument-specific accuracy work? If SASS-E were incorporated into CREPE's training set, would it perform as well on steelpan audio as Steelpan-Pitch? Or, even if SASS-E were incorporated into the training corpus, would the general, mixed nature of the training set prevent it from achieving state of the art results in such a specific situation? Typically, systems that are designed to work well in the general case lose accuracy in specific cases. Whether this is necessarily the case for data-driven pitch detectors deserves further research.

7. ACKNOWLEDGMENTS

We would like to thank Andy Schloss, Kirk McNally, Jordie Shier, Keon Lee, and Jason Ye for their support related to this work. Thank you for your help over the years.

8. ETHICAL STANDARDS

This work was supported by a doctoral fellowship from University of Victoria. None of the authors reported a conflict of interest. The authors recorded and edited all of the audio samples themselves on personal instruments and took care not to include audio from the commercial sample library in the training set. No other human or animal participants took part in this research which was conducted in accordance with University of Victoria's research ethics standards.

9. REFERENCES

[1] A. Achong. The Steelpan as a System of Non-Linear Mode-Localized Oscillators, I: Theory, Simulations, Experiments and Bifurcations. *Journal of Sound and Vibration*, 197(4):471–487, 1996.

[2] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello. Deep Salience Representations For F0 Estimation In Polyphonic Music. *ISMIR*, pages 63–70, 2017.

[3] A. Camacho and J. G. Harris. A sawtooth waveform inspired pitch estimator for speech and music. *The Journal of the Acoustical Society of America*, 124(3):1638–1652, Sept. 2008.

[4] A. de Cheveigné and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, Apr. 2002.

[5] J. Dubnowski, R. Schafer, and L. Rabiner. Real-time digital hardware pitch detector. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(1):2–8, Feb. 1976.

[6] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning*, pages 448–456. PMLR, June 2015.

[7] J. W. Kim, J. Salamon, P. Li, and J. P. Bello. CREPE: A Convolutional Representation for Pitch Estimation. *arXiv:1802.06182 [cs, eess, stat]*, Feb. 2018. arXiv: 1802.06182.

[8] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, Jan. 2017. arXiv: 1412.6980.

[9] C. Malloy. Study of timbral variation of tenor steelpan mallets through spectral analysis. In *Proceedings of Meetings on Acoustics*, volume 39, page 035009, San Diego, California, 2019.

[10] C. Malloy. Improved steelpan pitch detection through audio feature extraction and machine learning. *The Journal of the Acoustical Society of America*, 149(4):A122–A122, Apr. 2021.

[11] C. Malloy. Steelpan fundamental frequency estimation through audio feature extraction and deep neural networks. *The Journal of the Acoustical Society of America*, 150(4):A175–A175, Oct. 2021.

[12] C. Malloy and J. Ye. Using convolutional neural networks to estimate pitch directly from steelpan audio signals. *The Journal of the Acoustical Society of America*, 150(4):A174–A174, Oct. 2021.

[13] M. Mauch and S. Dixon. PYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663, May 2014. ISSN: 2379-190X.

[14] F. Muddeen and B. Copeland. Microphone Placement for Tenor Pan Sound Recording: New Recommendations Based on Recent Research. *West Indian Journal of Engineering*, 35(2):95–102, 2013.

[15] L. Murr and L. White. Metallurgy of the Caribbean Steel Drum. *Percussive Notes*, 38(1), 2000.

[16] A. M. Noll. Cepstrum Pitch Determination. *The Journal of the Acoustical Society of America*, 41(2):293–309, Feb. 1967.

[17] M. Ross, H. Shaffer, A. Cohen, R. Freudberg, and H. Manley. Average magnitude difference function pitch extractor. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 22(5):353–362, Oct. 1974. Conference Name: IEEE Transactions on Acoustics, Speech, and Signal Processing.

[18] T. D. Rossing, D. S. Hampton, and U. J. Hansen. Music from Oil Drums: The Acoustics of the Steel Pan. *Physics Today*, 49(3):24–29, Mar. 1996.

[19] T. Ryan, P. O'Malley, A. Glean, J. Vignola, and J. Judge. Conformal scanning laser Doppler vibrometer measurement of tenor steelpan response to impulse excitation. *The Journal of the Acoustical Society of America*, 132(5):3494–3501, Nov. 2012.

[20] S. Singh, R. Wang, and Y. Qiu. DeepF0: End-To-End Fundamental Frequency Estimation for Music and Speech Signals. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 61–65, June 2021.

[21] M. Sondhi. New methods of pitch extraction. *IEEE Transactions on Audio and Electroacoustics*, 16(2):262–266, June 1968.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[23] J. Turian, J. Shier, H. R. Khan, B. Raj, B. W. Schuller, C. J. Steinmetz, C. Malloy, G. Tzanetakis, G. Velarde, K. McNally, M. Henry, N. Pinto, C. Noufi, C. Clough, D. Herremans, E. Fonseca, J. Engel, J. Salamon, P. Esling, P. Manocha, S. Watanabe, Z. Jin, and Y. Bisk. HEAR: Holistic Evaluation of Audio Representations, May 2022. arXiv:2203.03022 [cs, eess, stat].