Tiny trainable instruments

by

Aarón Montoya-Moraga

B.S., Pontificia Universidad Católica de Chile (2014)
M.P.S, New York University (2017)

Submitted to the Program of Media Arts and Sciences
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

July 2021

Author . . . .
Program of Media Arts and Sciences
July 2021

Certified by. . . . .
Tod Machover
Muriel R. Cooper Professor of Music and Media
Thesis Supervisor

Accepted by . . . .
Tod Machover
Academic Head, Program in Media Arts and Sciences

Tiny trainable instruments

by

Aarón Montoya-Moraga

Submitted to the Program of Media Arts and Sciences
on July 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

Tiny trainable instruments is a collection of instruments for media arts, using machine
learning techniques and deployed in microcontrollers.

Thesis Supervisor: Tod Machover
Title: Muriel R. Cooper Professor of Music and Media

Acknowledgments

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

This thesis is the capstone project of my master's program, between the academic years 2019-2021 at MIT Media Lab. During these years I have been a Research Assistant at the groups Opera of the Future and Future Sketches.

The work presented here has been developed mostly working remotely during the COVID19 pandemic, while at home in Boston MA.

TODO: include photograph of my desk at home.

This thesis is a collection of media arts instruments, using tiny machine learning, and with a strong emphasis on digital ethics and Do-It-Yourself (DIY). Its main audience is beginners and artists, and it is my hope that this work can inform the discourse and practice of a new generation of artists, designers, educators, programmers, policy makers, activists, and enthusiasts.

Machine learning can be really difficult and often relies on proprietary software and

hardware. They often need to be trained in borrowed and rented resources, such as the service Google Colab, which allows you to share and train ML models on the cloud. This thesis shows users how to build machine learning art systems that can circumvent this and allow for more control of their data.

Artists have an unprecedented access to new tools for making new tools and instruments, and this thesis intends to be a foundation for a new generation of instruments for manipulating audiovisual material, using machine learning. This approach is really exciting because it allows beginners and artists to train their instruments instead of programming them, by inputting data for tuning, instead of having to write lines of code and fixed thresholds for changing their behavior.

Machine learning algorithms are written by humans, and then trained on biased data, and they are deployed to the world, where they affect our lives. These imperfections are highlighted over the course of this thesis, by being explicit about the assumptions and quantizations performed when working with data.

TODO: Add example of biased dataset that is commonly used.

The proliferation of surveillance tools and now of microcontrollers allows for an even more pervasive surveillance and data leaks by governments and corporations. This thesis allows for beginners and artists to develop their own databases, by self sensing and surveillance. TODO: explain self-sensing and surveillance, and explain harms of surveillance, with an example.

Low power and repurposable Tiny machine learning is defined as machine learning performed with low-power devices TODO: add citation or mention that it is my own definition. The proposal of media arts instruments that can run on little power, rechargeable batteries is a friendly use of resources for experimentation in arts. This is in contrast with the high power use and critique of other emerging fields such as Non Fungible Tokens (NFTs) and cryptoart. The proposal of these scriptable open source instruments allows for users to continuously tweak and modify their instruments,

11

repurpose the hardware, and enable sharing.

TODO: Add a reference to an early tiny ML paper.

1.2   Objectives

With the release of TensorFlow Lite Micro, the TinyML Foundation, new avenues have been opened for creative expression using machine learning in microcontrollers.

TODO: Explain what is TensorfLow Lite Micro, what does it do, the same for TinyML Foundation

COMMENT: add in a paragraph here that combines the points you make in the first paragraphs -> the main contributions of YOUR thesis... which it seems is basically just addressing a bunch of the context section? Blakeley Payne's thesis does a good job of outlining contributions. It's also good to do here to understand where you are going

Homebrew examples, teach people how to build their own databases, and how to circumvent corporations view,

TODO: Terms and conditions comic book. If people were to read terms and conditions, it would take them X years

You can train an instrument to only detect your voice, your accent, your living conditions.

TODO: add blurb main inspiration about trail building and skatepark, and playpens, sandbox

TODO: add Marina Berry's paper on playgrounds, child development

TODO: Technology and Public Art with Rafael Lozano-Hemmer | June 3, 2020 https://www.youtube.com/watch?v=QgVdEmqmuEE 36m28s Rafael Lozano-Hemmer: "Face recognition needs to be banned in all applications except art."

TODO: add Zoom example of garbage speech to text TODO: Add in a glossary of terms to go over the basics of machine learning TODO: Add contextual stories to frame each aspect TODO: More pictures TODO: Add how everything is public, even the errors TODO: Raspberry Pi example, it's cheap but it needs a lot of extra hardware to be used TODO: Also add examples about April Fools Day and Terms and Conditions, about owning your soul TODO: link https://www.youtube.com/watch?v=jOQ-9S3lOnM&t=125s TODO: Terms and conditions citation: (copied this in from my thesis, but check out McDonald and Cranor 2008) https://kb.osu.edu/handle/1811/72839 "Just considering privacy policies shown to people online in one year, it's been estimated that consumers would need approximately 244 hours to read, not skim, privacy policies shown to them in one calendar year."

1.3   Outline

This thesis will cover the following chapters:

1. Chapter 2: Background: the inspiration and context of this thesis. TODO: add literature that informs my work.

2. Chapter 3: Early experiments: my earlier work that led to this thesis, in the topics of media arts education, microcontrollers, and machine learning.

3. Chapter 4: Tiny Trainable Instruments: description of design strategies for the software and hardware, description of the support team working on this thesis.

4. Chapter 5: Project evaluation: user feedback, field notes.

5. Chapter 6: Future work: next iterations of the instruments, and their proposed use for educators and artists.

Chapter 2

Background

As part of the research that directly informed this thesis, I want to highlight the courses, projects and studies I have undertaken while at MIT Media Lab.

COMMENT ABOUT CLASSES: it may be better to do a lit review of sorts and discuss the material that informs your work here rather than specific classes

2.0.1 Classes

1. Comparative Media Studies, by Sasha Costanza-Chock

2. Recreating The Past, by Zach Lieberman

3. Sound: Past and Future, by Tod Machover

2.0.2 Projects

Some other projects I created during these years include:

1. SiguesAhi: an instrument to detect when oppressive institutions have ceased to exist. It is achieved with microcontrollers with Internet connectivity.

2. Open Drawing Machine, with Gaurav Patekar: an open source low cost programmable drawing machine

3. Introduction to networks for artists: a series of tutorials for beginners, to learn how to set up their own networks and collaborate in peer-to-peer ways for making art.

2.1   Media arts instruments

In particular, here I will highlight some media arts instruments that have inspired my research, because of their use and promotion of open source software and hardware, scripting capabilities, and other design considerations.

TODO: maybe add my own personal experience with these instruments.

2.1.1   Bast

The Czech company Bastl Instruments have released the Kastle series of instruments based on the Arduino Uno's chip ATMEGA 328P, and their source code is released on GitHub, with alternate firmware for different behaviors.

Currently they offer the Kastle v1.5, a melodic and drone synthesizer, and the Kastle Drum, for rhythmic work. The only difference between these synthesizers is the firmware and the faceplate, and it is encouraged to write new firmware to modify their behavior. Also they are forgiving instruments, its inputs and outputs are robust enough to allow for mistakes in connections, in an electrical and mechanical way.

This instrument inspired many characteristics present in Tiny Trainable Instruments, including the use of breadboard jumper cables for making connections, their low cost (90 USD), and their portability by use of batteries.

Add an example of how Moog synthesizers in the 1970s were as expensive as a house! And then as a car.

Mention how accessible the different components of the instruments are, how you can get everything from adafruit for not that much money, etc.

Add explanation about how this is not just on GitHub, but also published on the arduino IDE library, and how examples are worked already, and how things are pre-compiled.

Lower barriers include releasing a Bill of Materials, on Adafruit, inspired by the Drawdio project, which can be bought as a kit from Adafruit. TODO: explain why Adafruit is cool and easier to use than Digikey etc.

2.1.2   Critter & Guitari

This company has released standalone scriptable computers for arts, which run Linux operating system + Pure Data software.

ETC and EYESY computers for visuals, scriptable, Linux operating system + Python / pygame environment or openFrameworks.

They can run on power supplies, and are also portable by the use of batteries.

2.1.3   monome

Aleph: earlier sound computer.

Norns: sound computer, currently on its second iteration, with expanded hard drive. Also there is a DIY version which is cheaper and runs on a Raspberry Pi.

Norns is a Linux machine, running SuperCollider for the sound engine, and Lua scripts. It has spawned a community that continually releases new scripts and software updates.

2.1.4   Shbobo

Peter Blasser has released several collections / companies of musical instruments, the most famous one being Ciat-Lonbarde. Peter also runs Shbobo, which to date has two different instruments, the Shnth and the Shtar.

Both run on microcontrollers, and they use a new proposed language called Shlisp, based on Lisp, and also they can be programmed using the Fish IDE.

As of 2021, they became open source, which is available at github.com/pblasser/shbobo.

COMMENT ON OPEN SOURCE: something being open source doesn't necessarily mean it is accessible to a wider audience. is one of the goals of your work to create instruments that are accessible to a wider audience?

They promote computer-centric approaches to making sound, such as the use of integers and metaphors of finite state machines, and also allow for different ways of playing and sensing, such as the use of antennas for detecting hand distance, a microphone for detecting speech and whistling, and wooden bars with piezos for detecting pressure.

TODO: write how this inspired the new interactions i am inventing or appropiating for Tiny Trainable Instruments, such as a drum machine you can talk to, Alexa spinoff.

2.2   Education

This thesis is inspired by the work of the research group Lifelong Kindergarten at MIT Media Lab, led by professor Mitchel Resnick. On the book with the same title, he builds on Seymour Papert's work, and proposes that educational projects should have "Low floor, wide walls, high ceilings", and that learners thrive when they engage in the 4 Ps: "Peers, projects, passion, play" TODO: add how this project addresses the 4Ps

2.3   Creative Machine learning

COMMENT: what is the main argument of the whole piece and how does each independent part connect to that? you should start with a story from previous experiences that is particularly relevant as to why you were inspired to do this work. think "papert and the gears"

While being a graduate student and research resident at NYU ITP I saw how quick things changed in terms of machine learning. I saw how the project deeplearn.js allowed for people to train and deploy machine learning on their browsers, and how this library was acquired by Google and repurposed as TensorFlow.js, a JavaScript version of their machine learning framework TensorFlow.

In turn, at NYU ITP a team of artists and programmers built the library and community of ml5.js, with the 5 being an homage to p5.js. Technically, ml5.js is a wrapper for TensorFlow.js, in the same spirit that p5.js is a wrapper for HTML5 elements such as the canvas.

Another huge contribution to the landscape of machine learning for arts has been the release of the app Runway, which started as Cristóbal Valenzuela's thesis, and is now

a company with Alejandro Matamala and Anastasis Germanidis.

After leaving NYU ITP, I was a student at the month-long workshop "Autonomous Generative Spirit" taught by Gene Kogan and Andreas Refsgaard at the School of Machines, Make and Make Believe in Berlin 2018. We experimented with quick and cheap methods for machine learning, such as the k-nearest neighbors algorithm using the artist and beginner-friendly app Wekinator by Rebecca Fiebrink, and also more computer-intensive algorithms, which sometimes required proprietary hardware such as NVIDIA graphics cards to be trained in a matter of hours, instead of days or weeks using our computers.

TODO: add what does the speed enable you to do on a personal level? why is it important to share that speed with others?

The last spark that led me to this thesis was the release of two libraries for machine learning on the Arduino platform: The currently beta version Arduino KNN, which allows for on-device training and resembles my earlier studies with Wekinator, in a more portable and private way, no data leaves the microcontroller, and the whole neural network can be wiped with one click of a button.

At a more complex level, I am also working with the TensorFlow Lite Micro, which I learned from Arduino blogs, and which currently is supported by the hardware Arduino Nano 33 BLE Sense.

COMMENT TO THE ABOVE PARAGRAPH: you may not even need to include the specific details here, but just highlight in larger overviews the types of projects you've worked on or the educational fields that influence your work

In late 2020 and early 2021 I completed the just released series of 3 courses of the TinyML Professional Certificate by Harvard at the online platform edx.org

Newer books and references that this thesis was inspired by include the books "You

Look Like a Thing and I Love You: How Artificial Intelligence Works and Why It's Making the World a Weirder Place" by Janelle Shane (2019), and the book "Making Pictures with Generative Adversarial Networks" by Casey Reas (2019).

Also Yining Shi created a new class at NYU ITP in 2020, at the intersection of machine learning and physical computing.

2.4   Digital rights

Machine learning algorithms need data to be trained on. I think it's a human right to not be surveilled, and I hope my thesis can put a positive spin on the gathering of data, by letting users perform auto surveillance, like the Ai Weiwei piece WeiweiCam, a 2021 project where the artist installed cameras for self surveillance as a protest against the Chinese government.

A huge inspiration for my thesis has been the Guardian Project by the Electronic Frontier Foundation, and the Design Justice Network and the book Design Justice by Sasha Costanza-Chock.

TODO QUESTION: is any of the methodology you use to develop all of this work specifically related to either of these? you might connect specific aspects of the EFF project and Sasha's work that are connected

2.5   Instruments

The table 2.1 is an example of media arts instruments that are scriptable.

Maker	Instrument	Year	Computing	Software
BASTL	microGranny 2	YEAR	microcontroller	Arduino
BASTL	Kastle v1.5	YEAR	microcontroller	Arduino
Critter & Guitari	Organelle	YEAR	computer	Linux?
Critter & Guitari	EYESY	YEAR	computer	Linux?
monome	aleph	YEAR	computer	Linux?
monome	norns	YEAR	computer	Linux
Shbobo	Shnth	YEAR	microcontroller	ARM Cortex
Shbobo	Shtar	YEAR	microcontroller	ARM Cortex

Table 2.1: Table of media arts scriptable instruments

2.5.1   BASTL

BASTL Kastle, two iterations and a spinoff: Kastle, Kastle v1.5, Kastle Drum.

Based on Arduino, GitHub repository with alternate firmware.

Breadboard patching with jumper cables, inputs and outputs robust enough to allow for mistakes in connections.

2.5.2   Critter & Guitari

Organelle computer for sound, scriptable, Linux operating system + Pure Data software.

ETC and EYESY computers for visuals, scriptable, Linux operating system + Python / pygame environment or openFrameworks.

2.5.3   monome

Aleph: sound computer

Norns: sound computer, currently on its second iteration, with expanded hard drive.

Also there is a DIY versionm which is cheaper and runs on a Raspberry Pi. Norns is a Linux machine, running SuperCollider for the sound engine, and Lua scripts.

2.5.4   Shbobo

Peter Blasser's Shbobo

Shnth and Shtar

Shlisp language and Fish IDE.

github.com/pblasser/shbobo

2.6   Education

Mitch Resnick's book Lifelong Kindergarten

Low floor, wide walls, high ceiling

Peers, projects, passion, play

Gene Kogan and Andreas Refsgaard

2.7   Machine learning

ml5.js

Runway

TinyML Professional Certificate HarvardX

2.8 Digital rights

Electronic Frontier Foundation

Edward Snowden

Design Justice Network

Chapter 3

Early experiments

3.1 Microcontrollers

My first exposure to Arduino was as an undergraduate student of electrical engineering back home in Chile. The Arduino Uno was a very powerful device, and I saw its applications to arts, when with a friend we created a rudimentary automatic tuner for guitar, that performed pitch detection and then controlled a motor to move the tuning machine on the guitar to achieve the desired tuning, with a PID controller.

I didn't use it too much, because they were relatively hard to obtain, and I was more interested in software at the time.

Fast forward to 2015, I became a graduate student at New York University's Interactive Telecommunications Program, where on my first semester I took the amazing class Introduction to Physical Computing, with one of Arduino's co-creators Tom Igoe.

While freelancing in New York, I was introduced to an Arduino off-shoot, the Teensy, which captivated me by its USB MIDI capabilities, which allowed for standalone op-

eration without a host computer, and by its audio library, which allowed me to create interactive standalone experiences, triggering samples and applying audio effects on device.

While at MIT Media Lab, I was delighted by the newer versions of Teensy, which are even faster and more powerful, and which led me to start designing handheld samplers for field recordings.

This in turn led me to review the current NYU ITP materials for physical computing, where they currently stopped using the now classic Arduino Uno, and have incorporated

3.2 Machine learning

Class at School of Machines by Gene Kogan and Andreas Refsgaard on 2018.

ml5.js

Machine learning for artists

Piano Die Hard with Corbin Ordel at the alt-ai conference, with Wekinator and KNN algorithm.

ml5.js is a wrapper for Tensorflow.js, NYU ITP. Browser based

Runway ML by Alejandro Matamala, Anastasis Germanidis, and Cris Valenzuela.

Casey Reas' book for GANs.

Chapter 4

Tiny trainable instruments

4.1 Design principles

1. Cheap

2. Privacy

4.2 Technology

This project is based on an Arduino Nano 33 BLE Sense.

Arduino microcontroller

Arduino library KNN

TensorFlow Lite Micro

4.3 Programmable / remix

All examples included with this library were written with the aim of showing the fundamentals of how to build the instruments and different machine learning enabled manipulation of multimedia material, so that people could build on top of it and make it it their own, by changing the values of variables and adding more functionalities.

4.4 Philosophy and experience

4.5 Inputs

Enumerate sensors from the Arduino Nano 33 BLE Sense.

We are using the RGB color, proximity, gyroscope, accelerometer, and microphone sensors on the microcontroller, in order to capture the Inputs

1. Color

2. Gesture

3. Speech

4.5.1 Color

This approach uses the RGB color sensor from the microcontroller, with the auxiliary help from the proximity sensor, that is used to capture color information at a certain distance threshold.

The data is passed to a k-Nearest-Neighbor algorithm, programmed using the Arduino KNN library.

4.5.2 Gesture

This input uses the information from the Inertial Measurument Unit (IMU) of the microcontroller, including a gyroscope and accelerometer. It captures data after a certain threshold of movement is detected.

The data is passed to a TensorFlow neural network, programmed using the Arduino TensorFlow Lite library, and based on the included magic_wand example.

4.5.3 Speech

This input uses the information from the microphone of the microcontroller.

The data is passed to a TensorFlow neural network, programmed using the Arduino TensorFlow Lite library, and based on the included micro_speech example.

4.6 Outputs

The different outputs were picked, because of their low cost, ubiquity, and possibilities of expansion and combining them.

4.6.1 Buzzer

This output creates pitched sound, by using a PWM output.

4.6.2 Servo motor

This output creates movement and through that, rhytmic sounds.

The main inspiration for this output was the emerging use of motor-activated percussive instruments, such as the Polyend Perc.

4.6.3 MIDI

We wrote functionalities to manipulate MIDI innstruments, and included examples to interface with some popular and cheap MIDI instruments, such as the Korg volca beats.

We included examples for rhythmic and melodic elements, using two very ubiquitous and inexpensive MIDI musical instruments, which are the Korg volca beats, and the Korg volca keys.

4.6.4 Thermal printer

A thermal printer is the basis for creating written and literary output, inspired by the field of computational poetry.

We used the popular Adafruit Thermal printer kit, which is documented on their website and includes a software library, distributed over GitHub and Arduino IDE, and also as a submodule on this project's TinyTrainable software library.

4.7 Development

This thesis has been developed with the invaluable help of undergrad researchers Peter Tone and Maxwell Wang.

They have cloned both repositories, the main one and the Arduino library one, and have continuously submitted pull requests with their contributions.

Peter Tone has helped with research in data structures, library writing, and we have shared back and forth code, going from experimental proofs of concepts, and has also helped with the design of the user-facing library.

Maxwell Wang has proofread our code, has ran the examples, and has helped with the writing of the documentation for self-learners and for the workshops.

We all share a Google Drive folder, where we all share notes about our research and development of the library and the educational material.

4.8 Code

This thesis is distributed as a repository, hosted on the GitHub platform, and available at github.com/montoyamoraga/tiny-trainable-instruments.

The auxiliary files, such as the LaTeX project for this document, and the auxiliary Jupyter notebooks, and documentation and tutorials are included on this repository.

The main software component of this project is the TinyTrainable library, available at github.com/montoyamoraga/TinyTrainable and also through the Arduino IDE.

The code included on this library is distributed on the folders:

1. examples/

2. src/

4.8.1 src/

The source code for where there is a TinyTrainable.h and TinyTrainable.cpp file where we included all the basic functionality of the library. Additional subfolders include

inputs/

Base class Input and inherited classes for each one of the other inputs.

outputs/

Base class Output and inherited classes for each one of the other outputs.

tensorflow/

Auxiliary files, copied from the Arduino TensorFlow examples that we are building on top of. The files here, unless otherwise noted, are included without modifications and distributed through the Apache License included on each file's headers.

4.9 Opera of the Future projects

During the development of this thesis, I have been fortunate to collaborate on different capacities with other thesis by classmates at Opera of the Future, which has directly

informed my work.

Squishies, Hannah Lienhard's master's thesis, novel squishable interfaces for musical expression. We shared discussions about low-level sound design, code reusability, sound art education, digital instruments. We were part of a master's thesis working group, facilitated by Roya Moussapour with two other Media LAb classmates, where we workshopped drafts of our thesis. This practice and feedback has been critical in shaping the language and discourse of my thesis document.

Fluid Music, Charles Holbrow's PhD thesis, library design, documentation for contributors. The design of the interface, documentation, and scope of the thesis were a direct influence on my documentation and API coding style.

Chapter 5

Project evaluation

5.1 Digital release

This thesis lives on the internet on repositories, and at the MIT library.

The repositories are hosted on GitHub, to promote collaboration, and people can file issues and pull requests.

GitHub repository

Arduino library

PDF zine for explaining, reference as the PDF booklet for monome norns

5.2   Audience engagement

5.3   Workshop

For user testing and sharing this thesis, some workshops were conducted during TODO, with support from a grant at CAMIT for teaching the workshops in English in USA, and in Chile in Spanish, remotely over teleconferencing software and after being approved by the MIT COUHES TODO explain.

The workshop instructions are documented on the docs/ folder of the repository available at https://github.com/montoyamoraga/tiny-trainable-instruments

Each workshop consists of 2 sessions of 2 hours each, spread over a weekend.

On the first session we will first help people with installation of the software, and then move on to start wiring the materials on the electronic breadboard material. We will concentrate on the simpler examples with color input. We will also collect data of gesture and speech to create custom databases and use them to train other slow machine learning models, that will keep on running on the student's workshops after the workshop is over.

On the second session we will use the result of the trained models to create more advanced instruments that react to gesture and speech. We will also show the participants the other

5.4   Multimedia show

Livestreamed show with multiple artists incorporating Tiny Trainable Instruments to their practice.

Chapter 6

Conclusion

This thesis project is a

6.1 Future work

6.1.1 Education

I hope that this thesis project is adopted by educators, to introduce students to machine learning, physical computing, media arts, and ethics.

6.1.2 Artist workflow

Training instead of programming.

6.1.3   Packaging

Low hanging fruit is to package a Tiny Trainable Instrument with a set of particular outputs, on a perfboard or PCB.

The next step would be to create enclosures.

6.1.4   Gallery

Bibliography