SPNet: Object Detection of Antinode Regions in Oscillating Steelpan Drums

Scott H. Hawley[1] and Andrew C. Morrison[2]

[1]Belmont University, Nashville, TN
[2]Joliet Junior College, Joliet, IL

Thanks for computer hardware:
Robert Magruder & Thom Spence (Belmont bosses)
Ralph Muehleisen (Argonne NL allocation
"Deep Learning Applications for Musical Acoustics and Audio”)

Acoustical Society of America 178th Meeting, San Diego CA, Dec 4, 2019

Talk Outline

Work in Progress: No photos, please.

Morrison's previous work

- What are [Caribbean] steelpan drums?
- What's interesting about them?
- Videoing oscillations via hi-speed ESPI
- Annotating the images (frames)

Hawley's contribution

- Strategy: Use humans' annotations to train a machine learning (ML) model
- How the model works (YOLO + mods)

Results

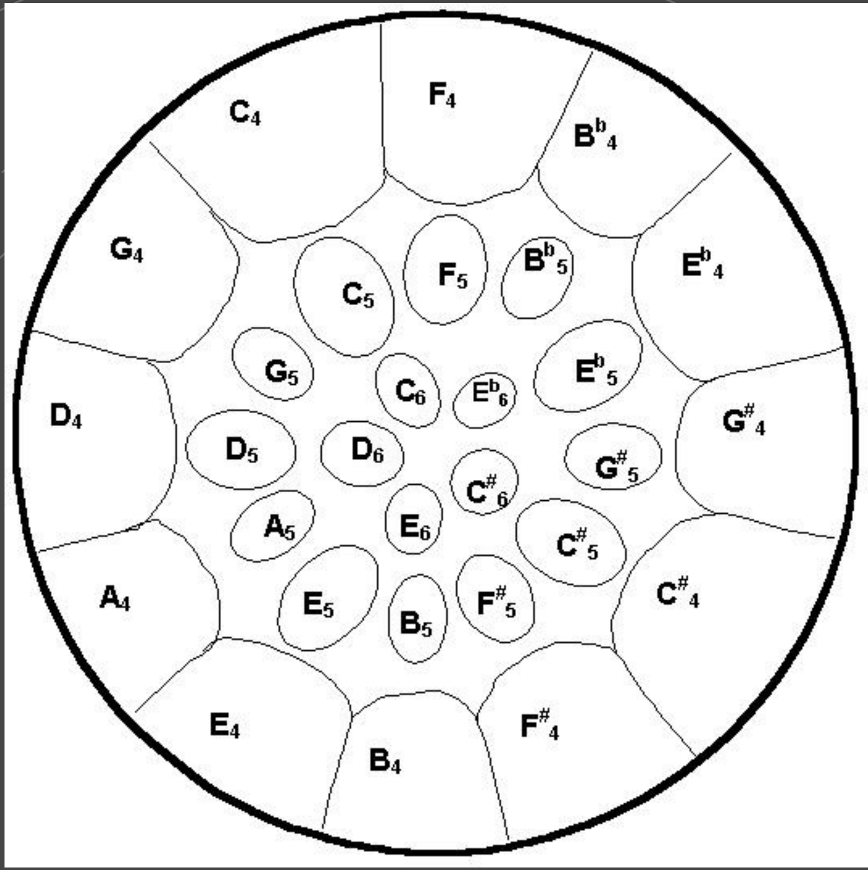
Future Work

Steelpan History

(Next 7 slides Morrison's, +errors by Hawley)

- "Probably the most significant acoustic instrument invented in the past century."
- Only been around ~75 years.
- Originated in Trinidad and Tobago, islands in South Caribbean.
- Although the steelpan appears to be simple instrument, it is deceptively complex

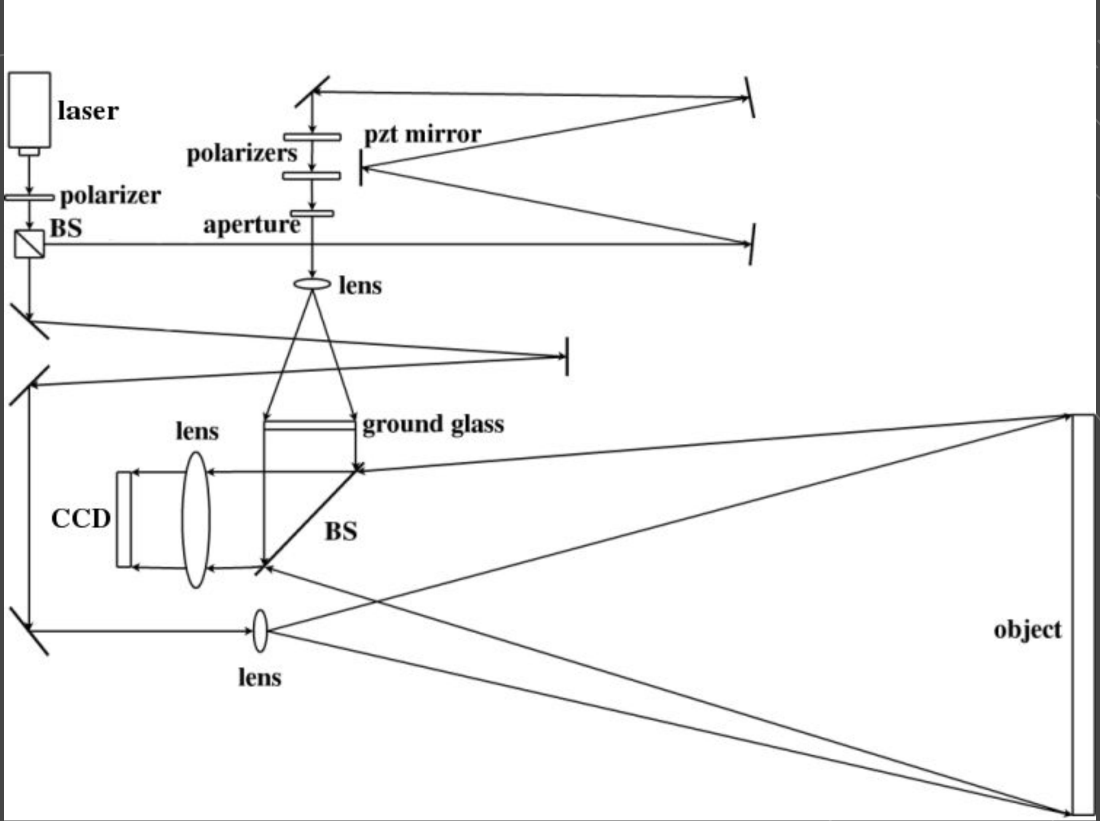
Steelpan Layout & Tuning



- Different domains/regions also called "notes"
- Shown: Low tenor steelpan, in "fourths and fifths" layout
- Tuned by hand by Bertrand Kellman

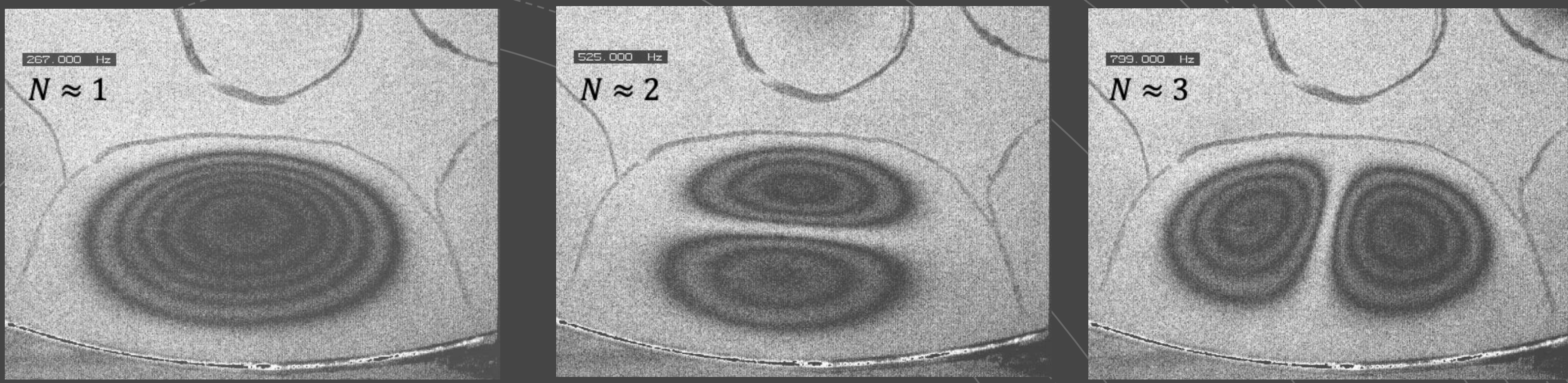
Illuminated via ESPI

Electronic Speckle Pattern Interferometry (Thom Moore)



Oscillation Modes

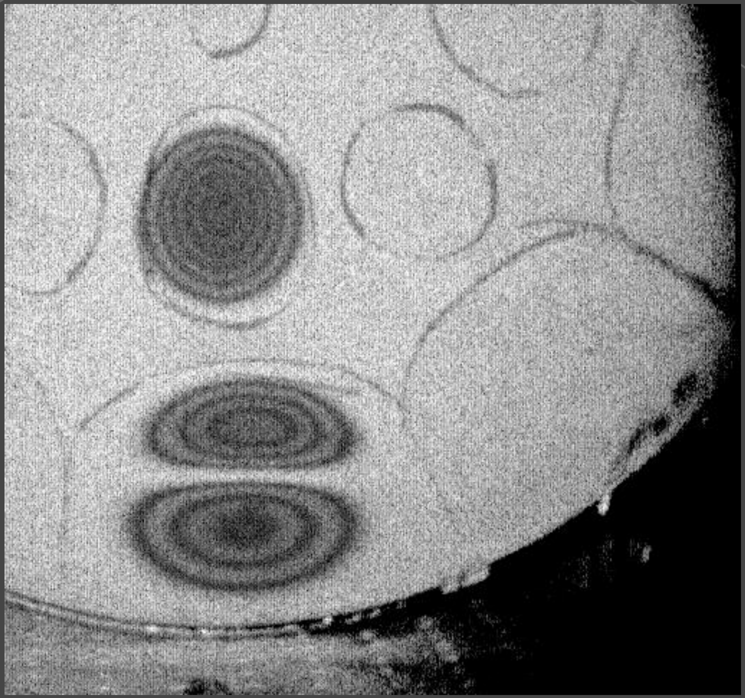
First three resonances of a single note [region]:



- Interference fringes/”rings” show contours of constant phase-difference between laser beams, correlate with contours of deviation of the surface (height)
  - similar to lines on a topographic map
- Number/density of rings is correlated with amplitude, not frequency [of sound]
- But the change in fringes over time can be revealed via high-speed video...

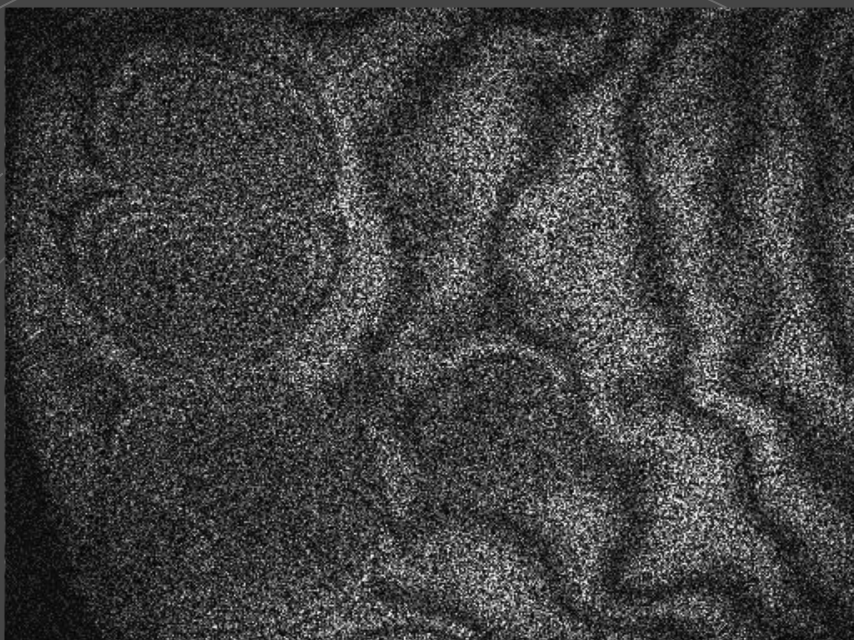
What makes the steelpan unique?

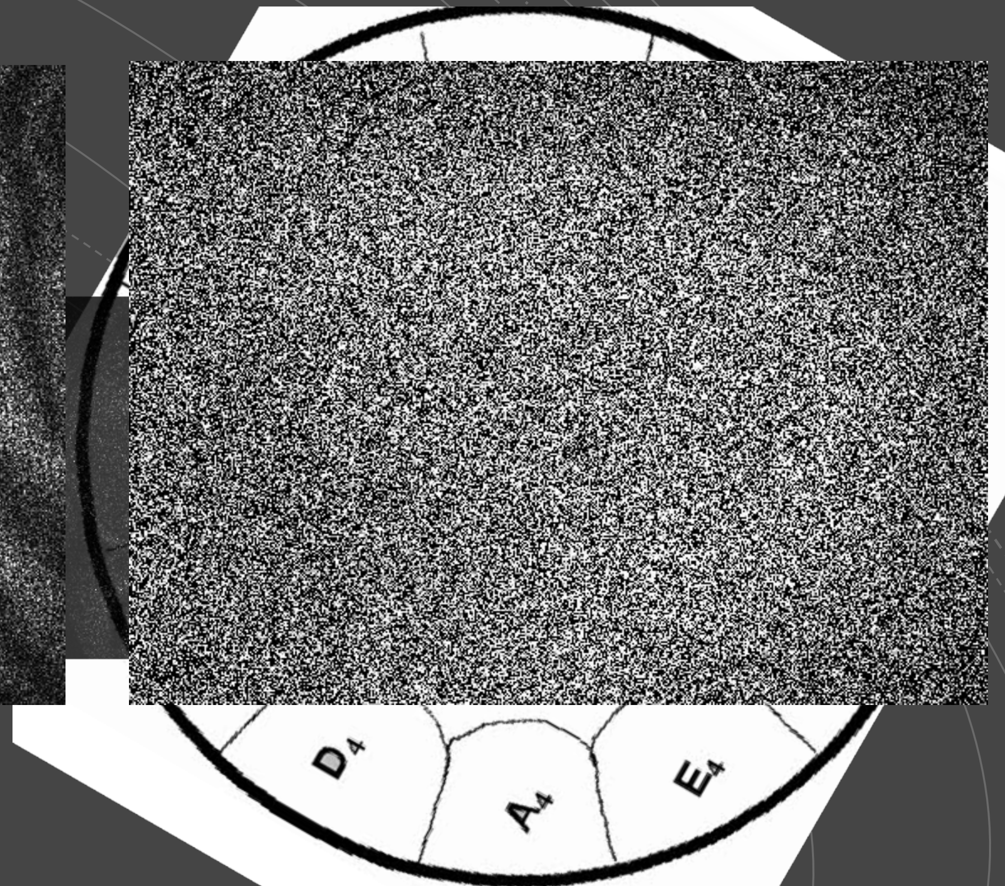
- Diverse set of couplings between “note” regions
- Strike it in one place, yet oscillations also appear elsewhere



High speed imaging!

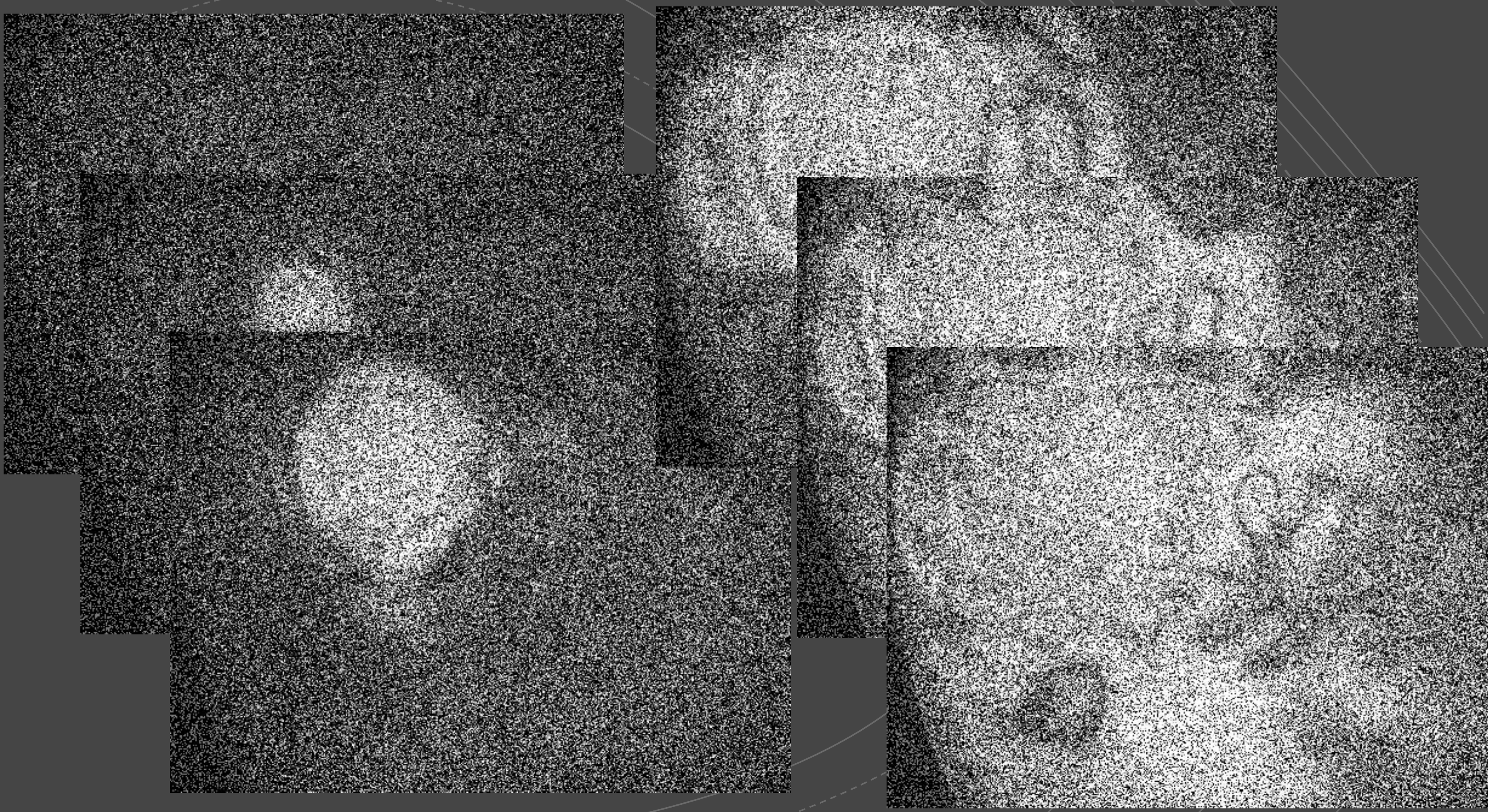
Morrison & Moore



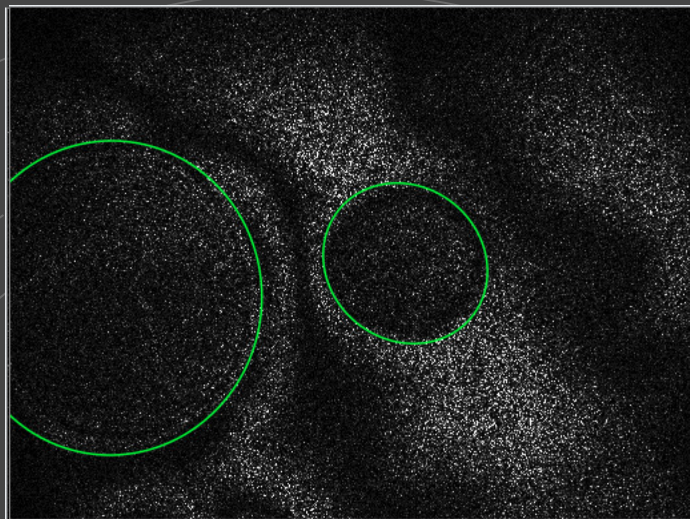


Looking at individual frames

Complex structure arising from a single strike...

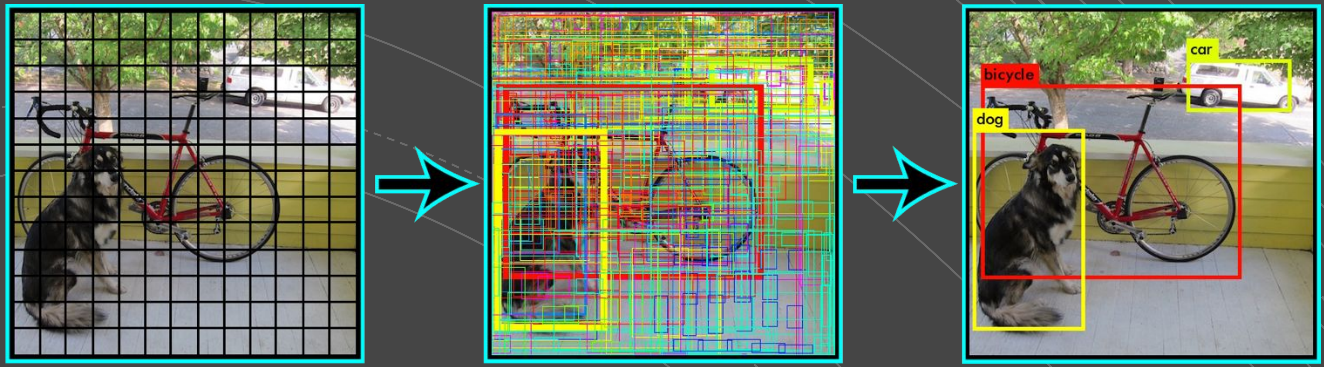


Annotating Images (by hand)



- To better understand drums' dynamics, track/analyze features in videos
- Crowdsourced to human volunteers, via Zooniverse.org, https://www.zooniverse.org/projects/achmorrison/steelpan-vibrations
- Users draw ellipses around ring groups -- antinode regions -- and count rings
- ...for each frame in video (~40k frames per video)
  - Slow-going: After several months, only had ~1000 annotated images
  - Reliability?: Some users entered 'junk.' Need multiple users' annotations per image in order to average, etc.
- Hawley: "*Someone* could probably train a ML model to do that"

Object Detection

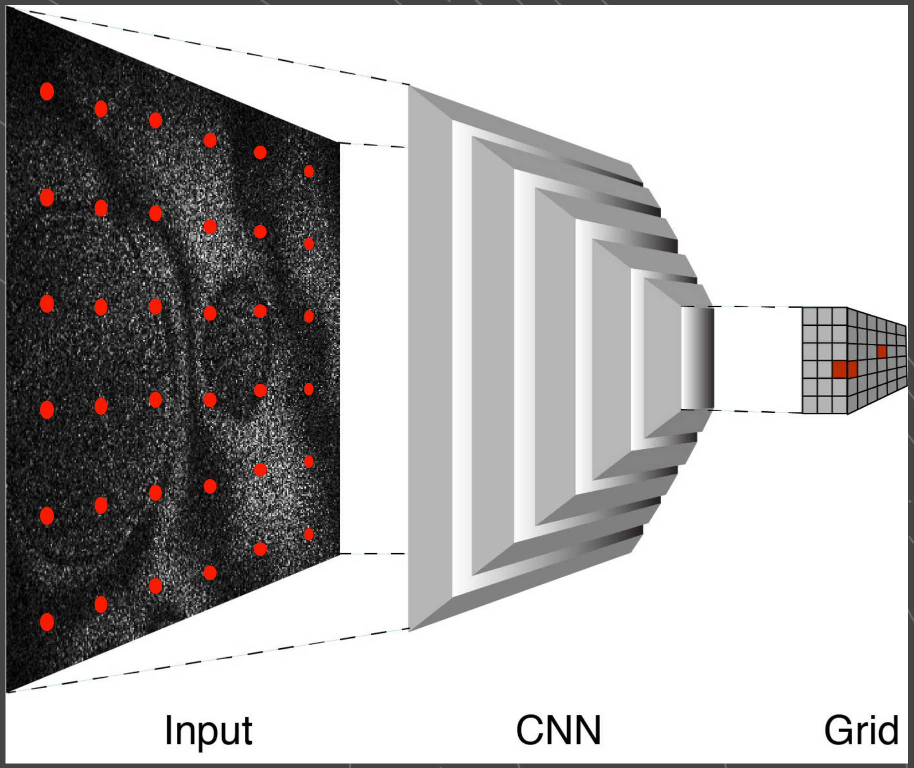


Source: YOLOv2, pjreddie.net

- Computer Vision problem
- Staple of surveillance capitalism! ;-)
- Usually predicts rectangular bounding boxes with (classification) labels
- But we want rotated ellipses and ring-counts
- ...so had to write custom code

SPNet Overview

- Based on YOLOv2 ("You Only Look Once") method, + tweaks
- Convolutional Neural Net (CNN) outputs a (6x6x2*) grid of "predictors"
- Dataset consists of input images + target output (aggregated) human annotations of...
  - **Center of ellipse(s) (x,y)**
  - **Semimajor/minor axes (a,b)**
  - **Rotation angle of ellipse (θ)**
  - **Number of rings (N)**
- but these target outputs are modified prior to training (to make it work better)...



^ e.g. ResNet, DenseNet, NASNetMobile,...

*6x6x2 chosen via experimentation/tweaking – e.g. 6x6x3 shown in picture!

Modified Target Outputs

- Instead of:

  1. Center of ellipse (x,y)

  1. Semimajor/minor axes (a,b)

  1. Rotation angle of ellipse (θ)
  2. Number of rings (r <= 11)

- We train using:

  0. **Existence of object (p = 0 or 1)**
  1. **Offset of ellipse center (x,y) relative to center of (nearest) grid-predictor**
  2. **Scaling of semimajor/minor axes (a,b) relative to default size**
  3. **$c = \cos(2\theta)$ and $s = \sin(2\theta)$**
  4. **Number of rings (r <= 11)**

  ...mapped “internally” onto [-0.5, 0.5] (“zero mean, unit variance”)

- Also, per image, per grid predictor, target outputs are ordered left-to-right first, then top-to-bottom (for “uniqueness” in training)
- Training model ⇔ minimizing loss function between predictions & targets...

Loss Function, v1.0

- Simplest, and it works: Mean Squared Error (MSE), with special weightings λ ("regularization parameters")
- Notation: Squared Error $\Delta_q^2 \equiv (q - \hat{q})^2$, where $q \in \{p, x, y, a, b, c, s, r\}$ is target ("true") output value, $\hat{q}$ is corresponding prediction
- For each grid predictor j, the loss L_j is

$$\begin{aligned} L_j &= \lambda_p \, \Delta_p^2 \\ &\quad + p\left[\lambda_{center} \left(\Delta_x^2 + \Delta_y^2\right) + \lambda_{size}\left(\Delta_a^2 + \Delta_b^2\right)\right. \\ &\quad \left. + \lambda_{angle} (a - b)^2 (\Delta_c^2 + \Delta_s^2) + \lambda_r \, \Delta_r^2\right] \end{aligned}$$

don't bother if doesn't exist ->

$p = 0$ or 1

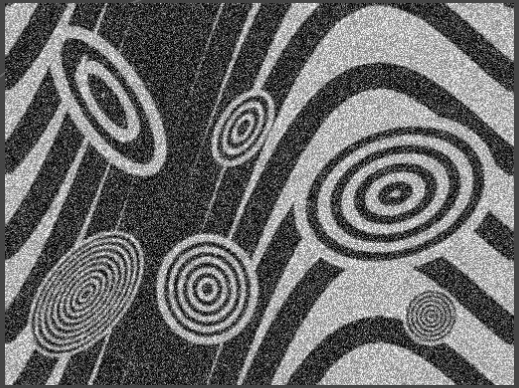
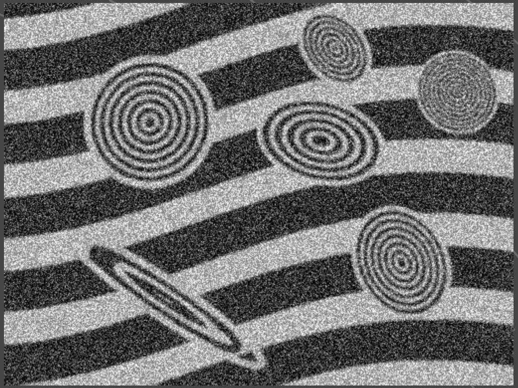
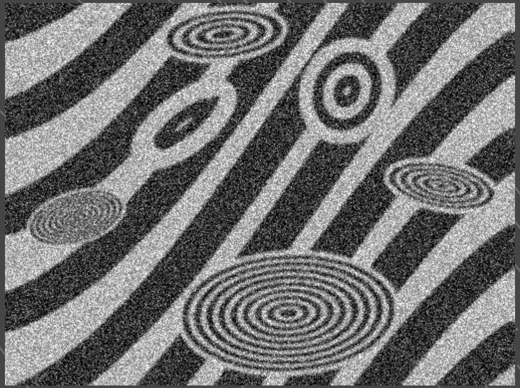
<- the more circular, the less we care about rotation

- "Total" loss L over all N=6x6x2=64 predictors is the mean

$$L = \frac{1}{N} \sum_j L_j$$

Fake Dataset

- Useful for developing ML system while real dataset small/unclean
- Samples:

- Model does well on fake data
- ...but not similar enough to real data to be useful for transfer learning or data augmentation to/on real dataset
- Future work: fake data via Generative Adversarial Network (GAN)?

(Real) Data Augmentation

- For 1000 images, need more variance to avoid overfitting
    - flip horizontal/vertical
    - change contrast/brightness,
    - add noise,
    - cut out regions, add "salt'n'pepa" noise
    - affine transformations: rotate, translate,…
- Note that for some must also change target outputs to match
    - More difficult than augmentation for mere classification
    - So do "hard" aug's before training: 1000 images → 50,000 images!
    - "Easy" aug's (that don't change targets) done "on the fly" Type equation here.

Implementation Details

Code

- Python + Keras
- Lets us swap in CNN models: NASNetMobile works
  (rescale input images to 331x331 pixels)
- Adam optimizer, "1cycle" learning rate schedule
- GitHub repo is private, but public when we publish

Hardware

DIY desktop builds: (no budget-line-item for cloud)

- 2017: NVIDIA GTX 1080 GPU, 32GB RAM
- 2018: Dual Titan X GPUs, 64GB RAM
- 2019: Dual RTX 2080 Ti GPUs, 128GB RAM

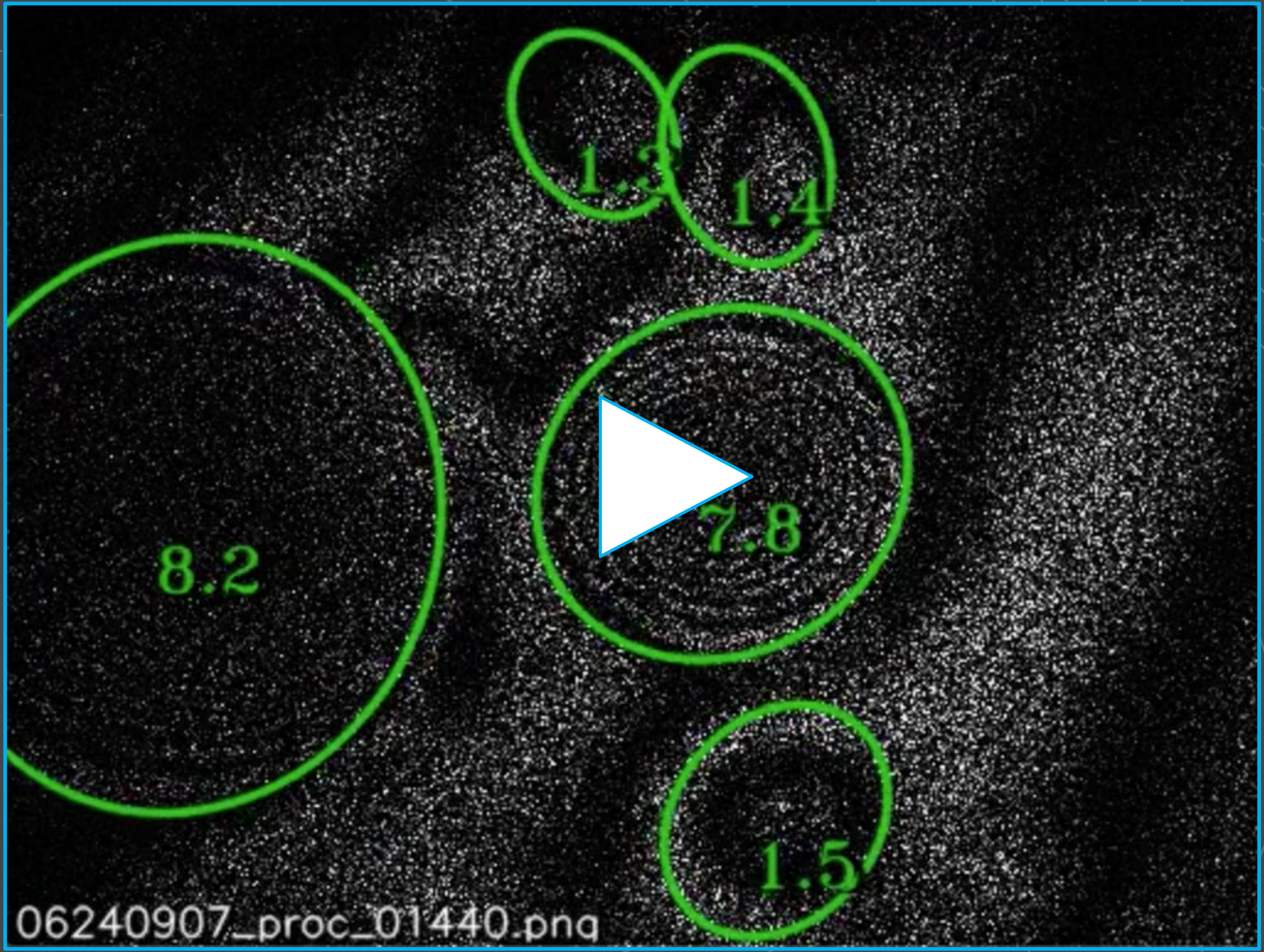
Performance

Depends on hardware

- Typical training runs 6-12 hours
- Inference runs at 300-500 FPS

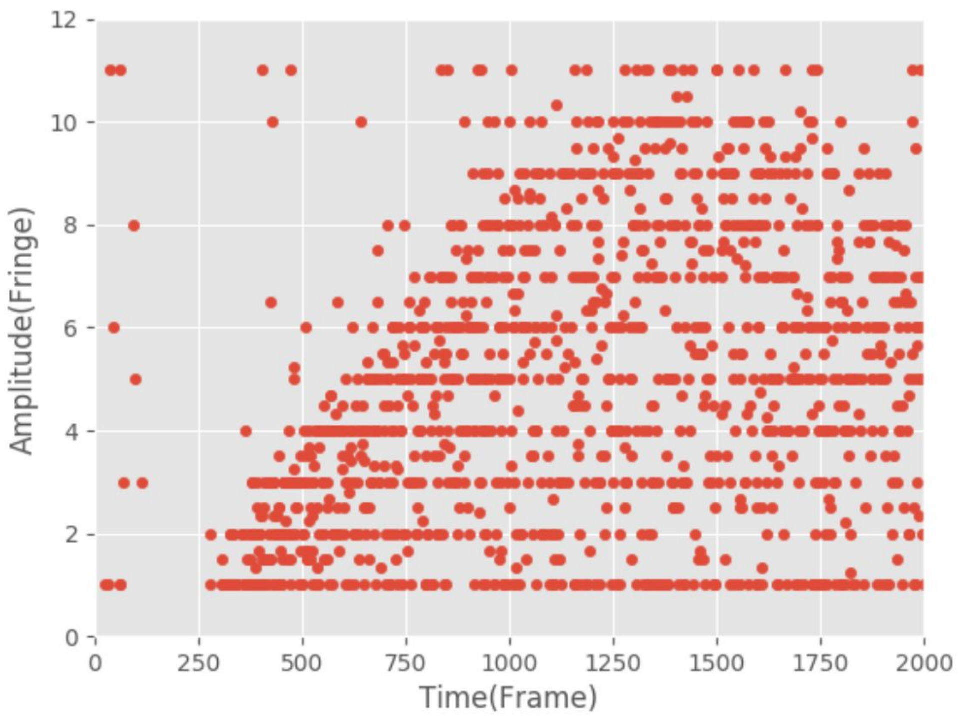
Results: Movie

http://hedges.belmont.edu/~shawley/steelpan_demo/spnet_steelpan_movie.mp4



Extracting Physics: Initial analysis

Morrison

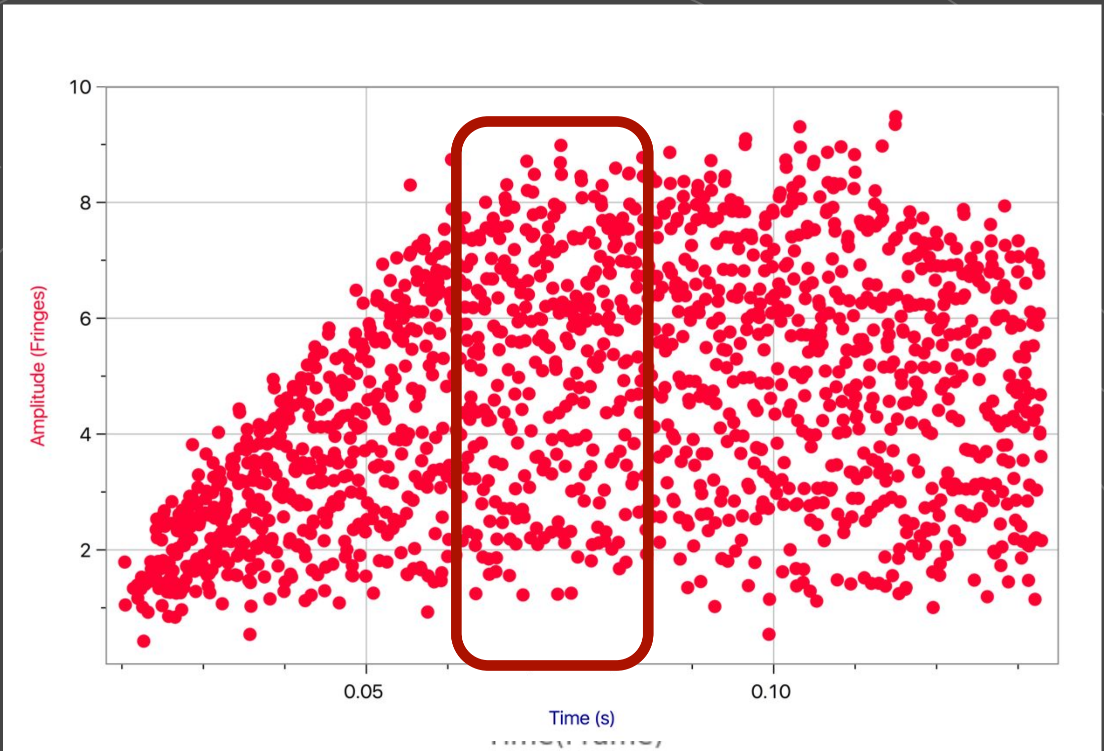


(a) Measures amplitude by the number of fringes the antinode contains.

Fairly noisy, but hints of a trend.

Extracting Physics: Updated analysis

Morrison



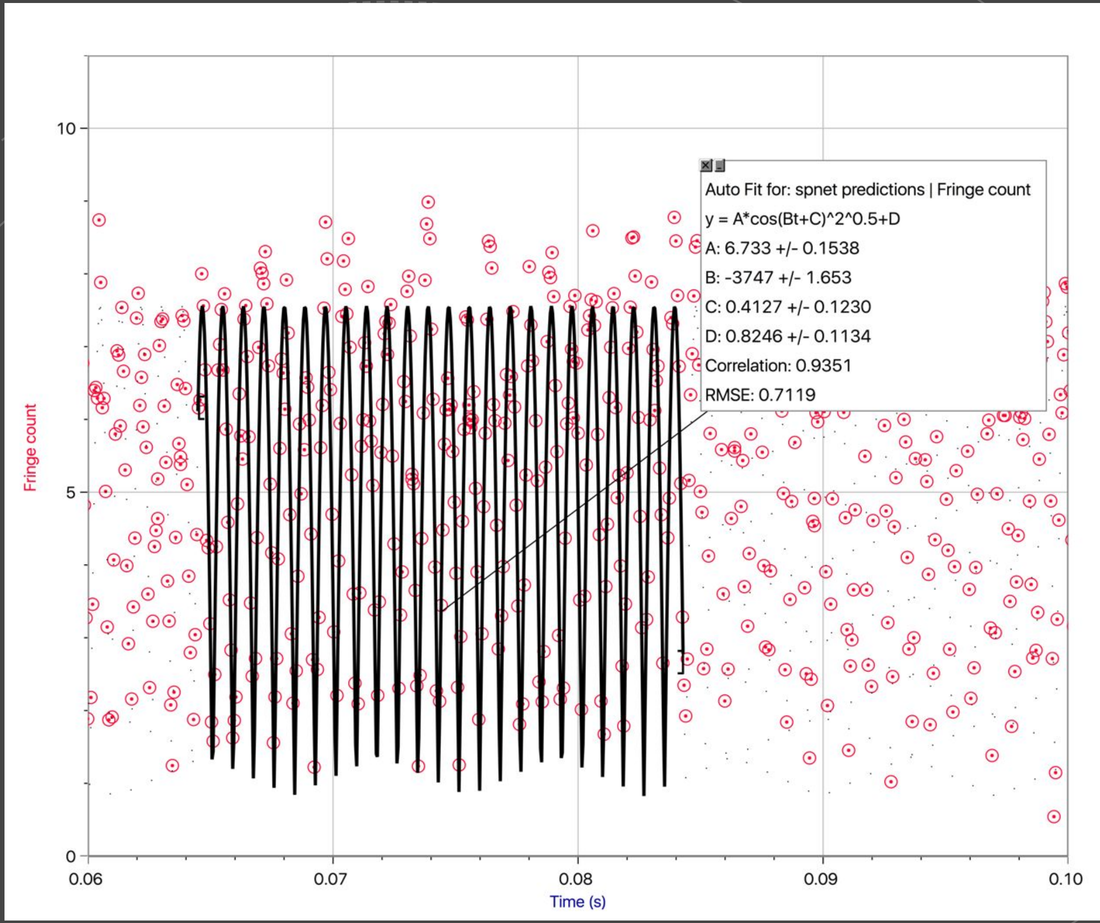
(a) Measures amplitude by the number of fringes the antinode contains.

Still noisy?

Look at smaller range of data

Physics Extracted!

Morrison



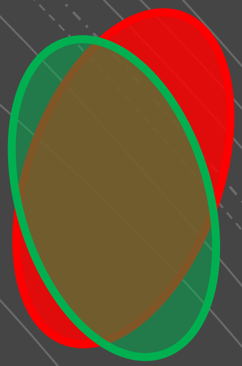
Fits $|\cos \omega t|$

$f = 596$ Hz

This is D_5 (close)

Results: ML Metrics

- Graph of Loss (goes down)
- “Accuracy” (goes up but levels off)
- Intersection-Over-Union (IOU) scores? □
- How well does it generalize?
- ...oh no we’re almost out of time! ;-) Working on these! “Work in progress”



Future Work

- Publish "as is", in POMA or special JASA issue on Musical Instruments!
- Better OD scheme, e.g. "Loss function v2.0": MSE for "regression" variables $\{x, y, a, b, c, s, r\}$, plus cross-entropy for "classification" variables (p,...and r?)

$$L_j = -\lambda_p[p \log(\hat{p}) + (1-p)\log(1-\hat{p})]$$
$$+ p\left[\lambda_{center}\left(\Delta_x^2 + \Delta_y^2\right) + \lambda_{size}\left(\Delta_a^2 + \Delta_b^2\right) + \lambda_{angle}(a-b)^2(\Delta_c^2 + \Delta_s^2) + \lambda_r \Delta_r^2\right]$$

$p = 0$ or 1
$0 < \hat{p} < 1$

...but training "crashes" after a while if I do this.

- GAN for "better fake" training data?
- Try on other system(s)? This was a very specific problem. Won't generalize to non-ellipse shapes (e.g. guitars, violins). Maybe could try image segmentation via U-Net, etc.
- Use time-dependent model. Currently we only process single images, but including prior (video) frames in inputs should help

Open Questions / Applicability

- This project was very specific: Replicate what Morrisons' human volunteers do, only faster & consistently.
- Not intended as a generic 'product' for all instruments.
- So, what about other instruments?
  - Do you have a dataset as ambitious as Morrisons?
  - Could one do transfer learning from this model & dataset (to reduce need for data on new instruments)
- And other antinode shapes?
  - This model only works on oval shapes.
  - What about other shapes? (e.g. "peanuts")
    - Try "image segmentation" instead of "object detection"