AutoTrace: An automatic system for tracing tongue contours

Gustave Hahn-Powell [1]
Diana Archangeli [1,2]
Jeff Berry[1] Ian Fasel[3]

[1] *Department of Linguistics, University of Arizona, USA*
[2] *Linguistics, University of Hong Kong, HK*

October 27, 2014

[1] Original authors of software

Why an Automated Method?

Manual tracing of tongue contours is impractical

- . . . ultrasound images are captured at 30 - 100+ fps[2]
- . . . but an expert takes two seconds or more to trace one frame![3]
- . . . for five minutes of speech recorded at 30 fps, it would take an expert **50**+ hours to trace all **9,000** frames!

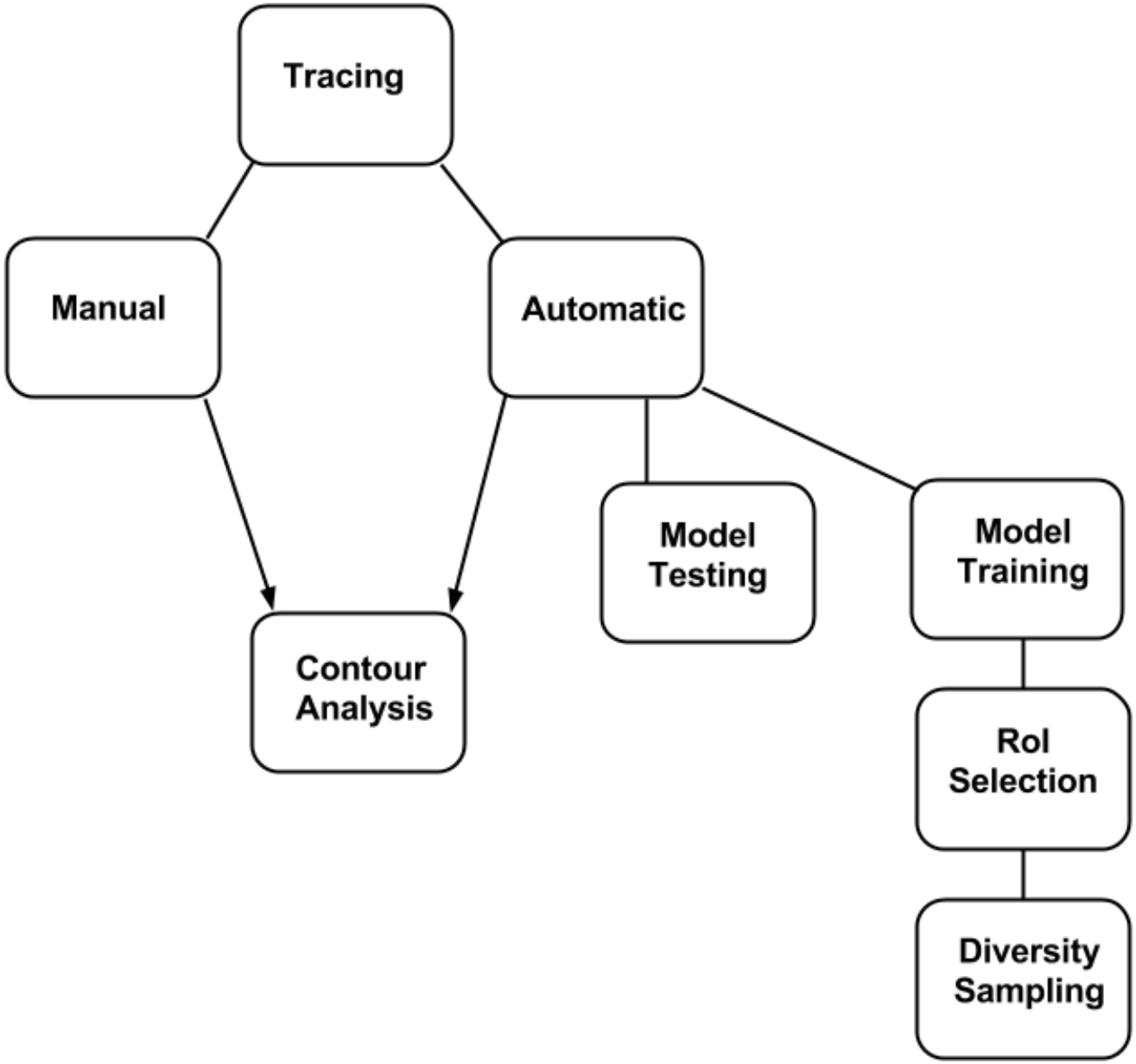
We need an automated system with performance that rivals human experts.

[2] 30 fps for Sonosite Titan
[3] (Berry, 2012, 28)

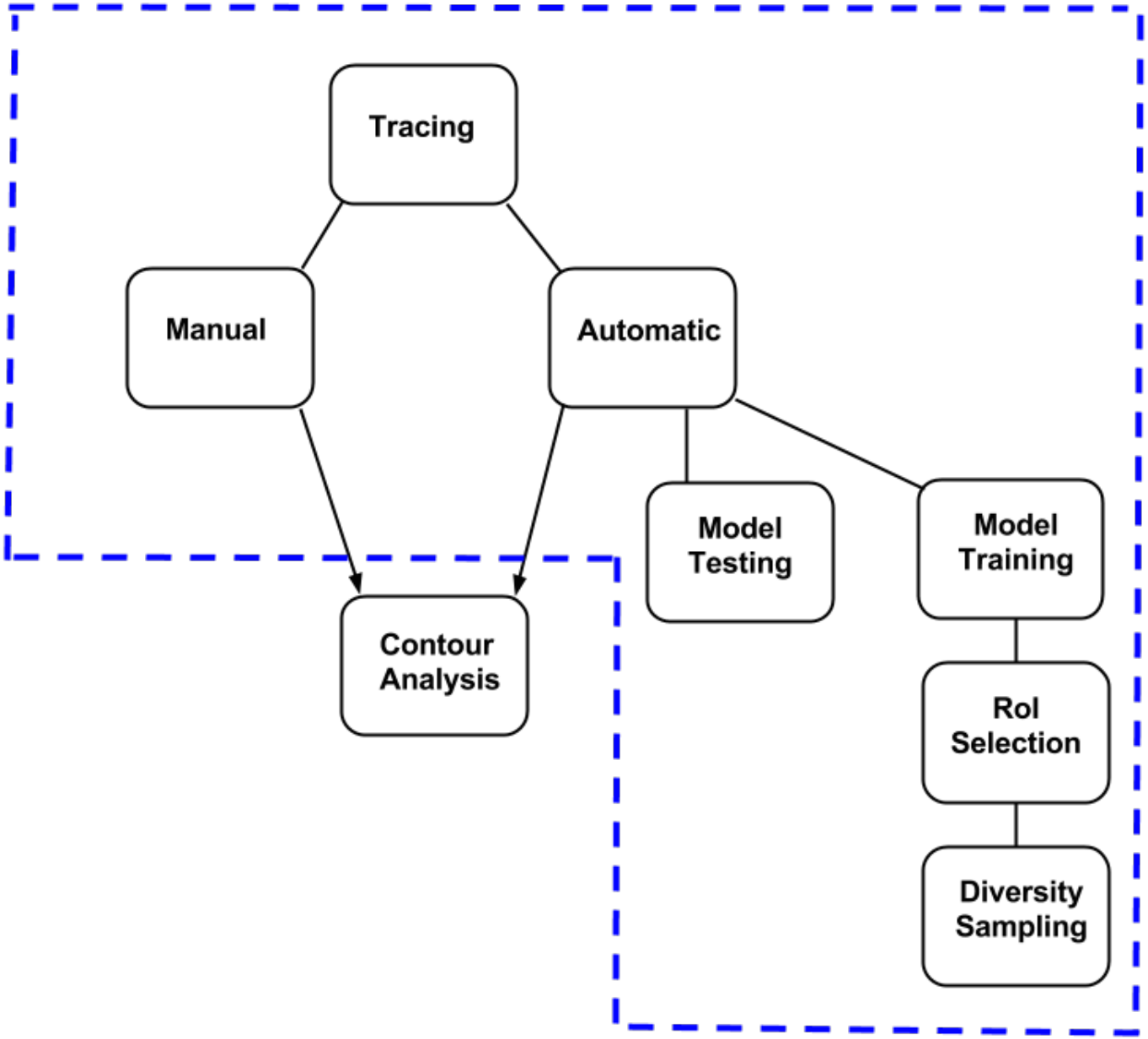
What is AutoTrace?

- Uses a *translational* Deep Belief Network...
  - state-of-the-art ANN
- Model (network) requires training
- State-of-the-art automated method
  - see Csapó and Lulich (2014)

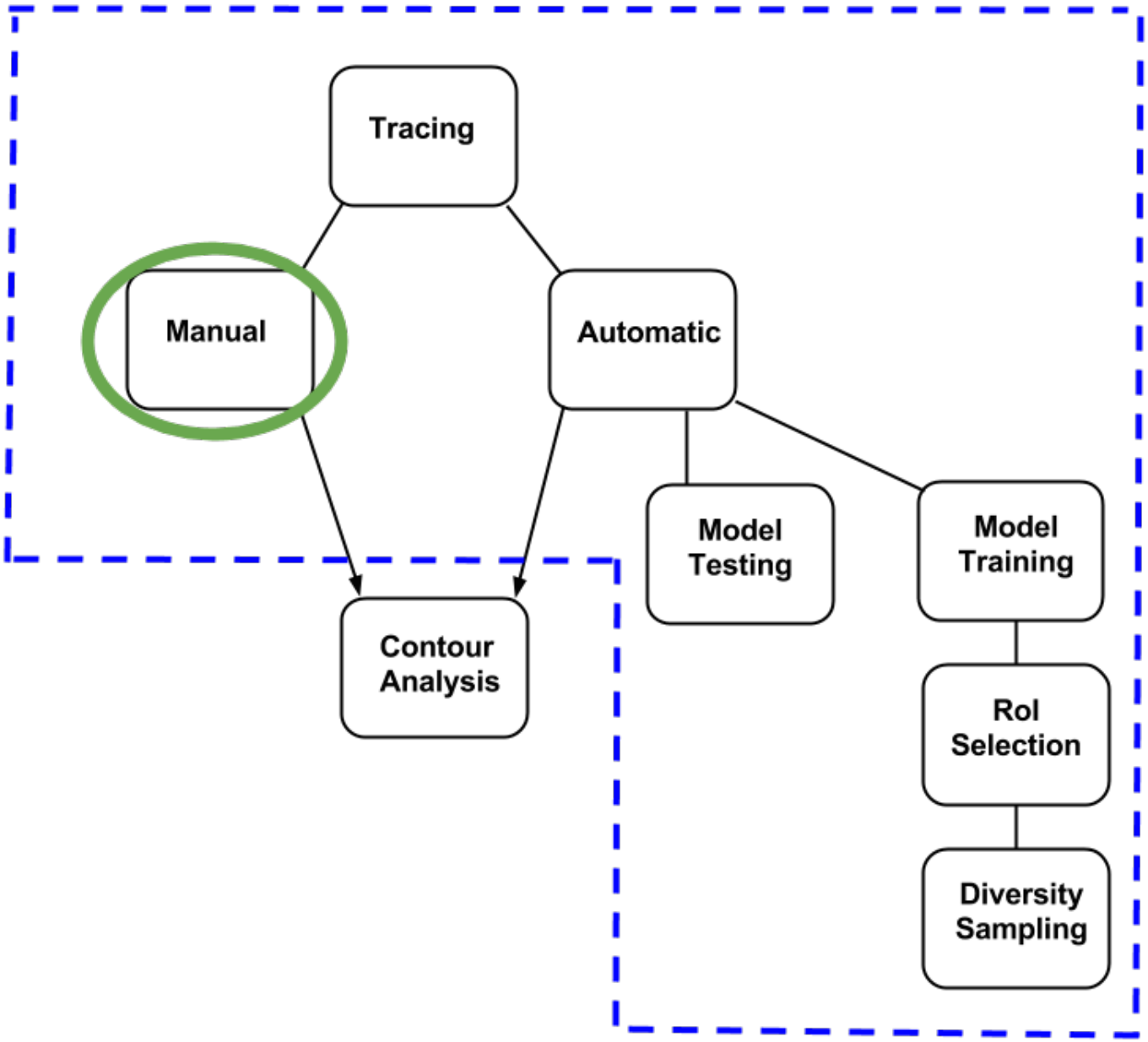
What is the AutoTrace Toolkit?



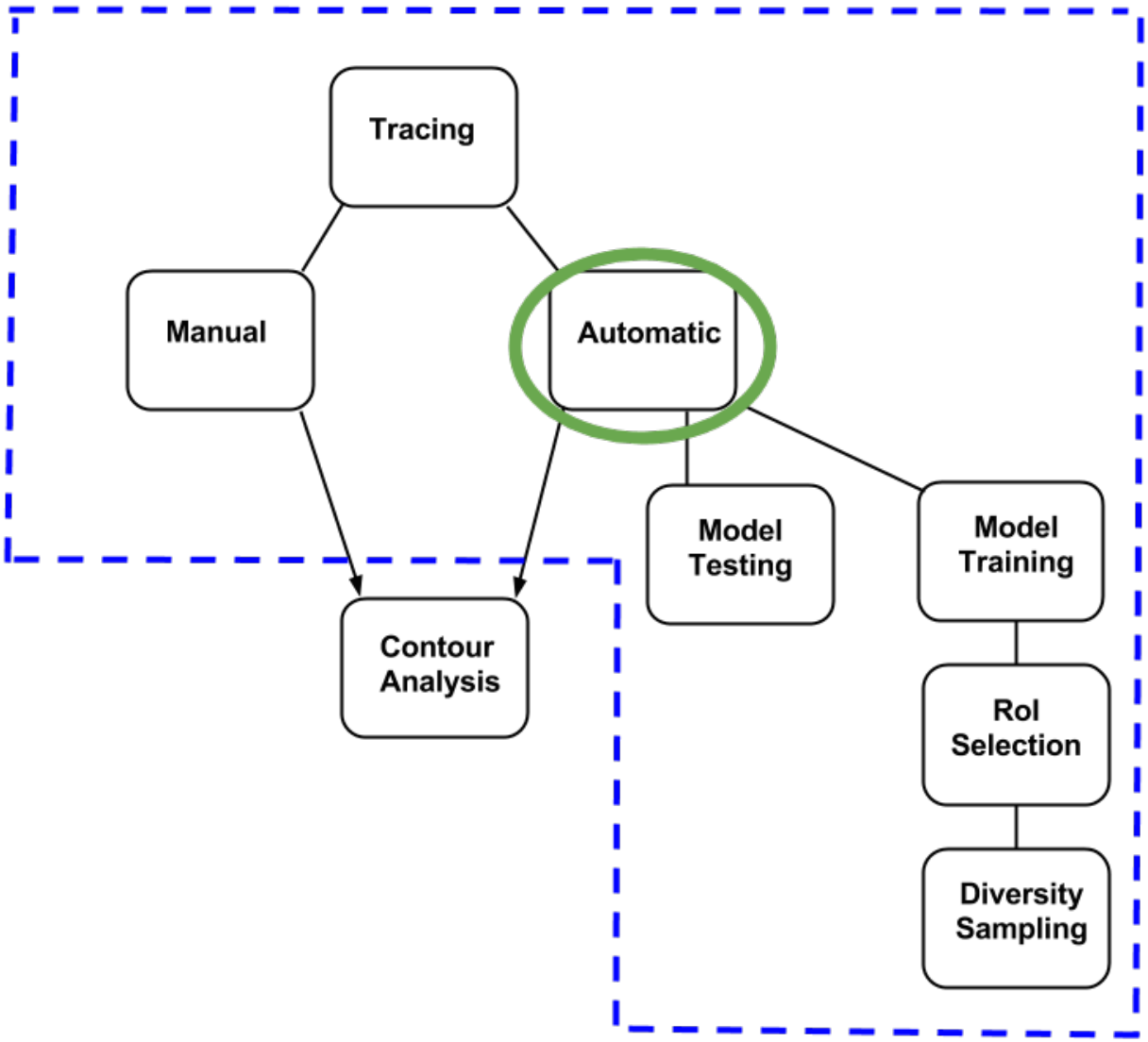
Today's Focus



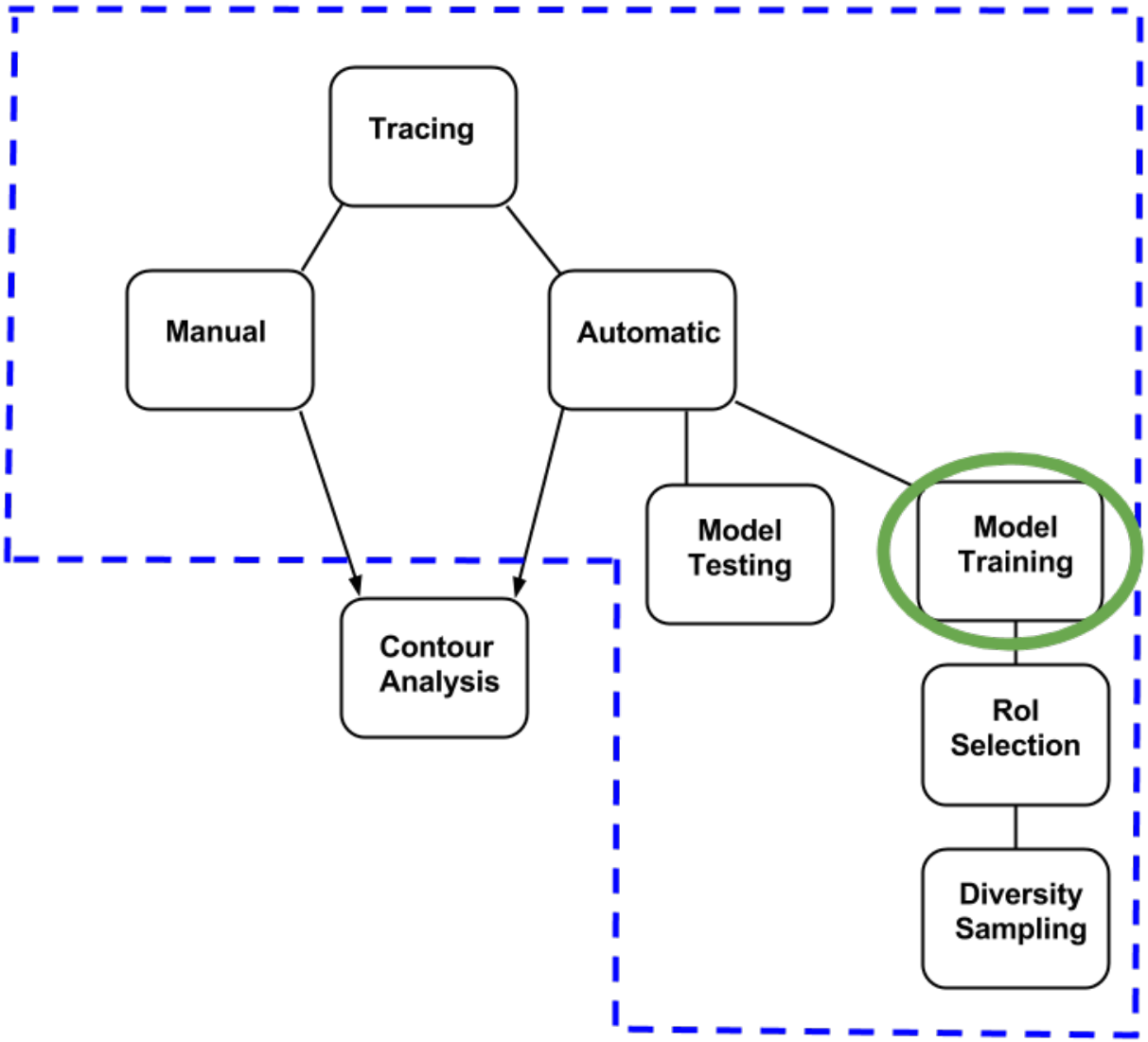
Manually Tracing Contours



Automatic Tracing



Why train your own network?



Training AutoTrace

1. Select (traced) images for training
   1.1 Specify region of interest (RoI)
   1.2 Calculate diversity scores
2. Train network (model)
   - network is fed ultrasound image and corresponding trace

Selecting images for training

"Informed Undersampling" (Liu et al. (2009))

- **Problem**: there are too many images...
- **Solution**: sample from the "**most diverse**" images

Berry (2012)

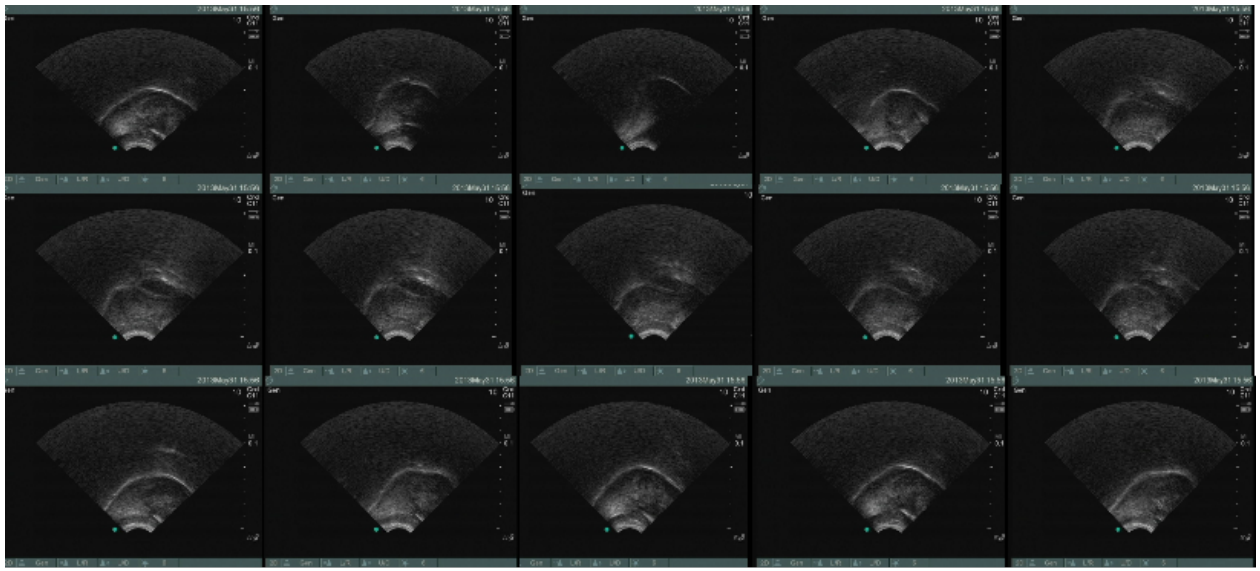
- training set of most diverse images outperforms a random sampling of the same size
  - MSD from high entropy < MSD from random sampling
    - * ($p < 0.0001$)

Most-Diverse Training I

Training Design

1. Select a pool of images

Figure: Examples of ultrasound frames



Most-Diverse Training II

2. Determine RoI

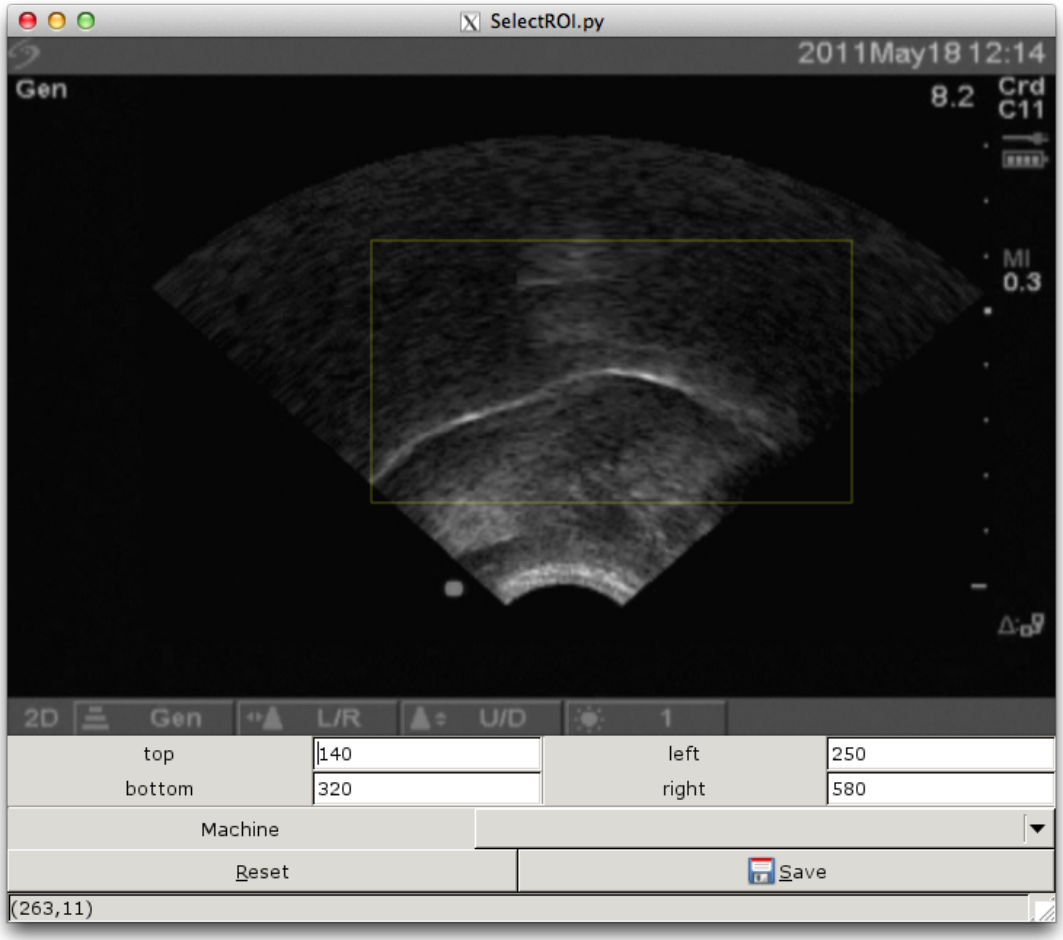


Figure: Example of software used for RoI selection

Most-Diverse Training III

3. Calculate an average image (averaging pixel values)

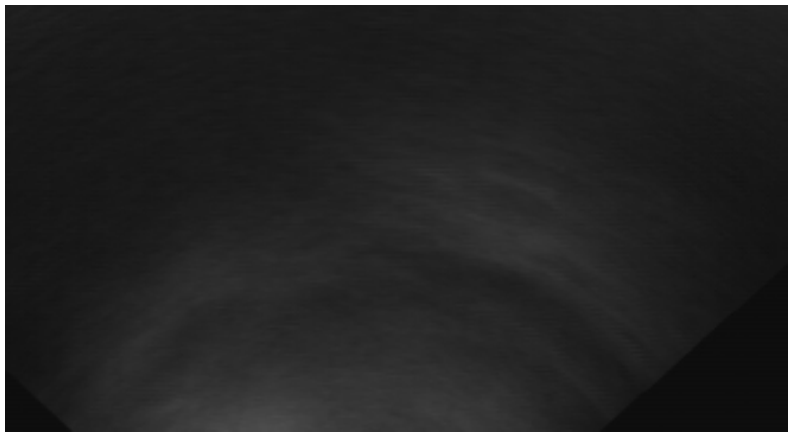


Figure: RoI-constrained averaging of pixel values

Most-Diverse Training IV

4. For each image in the training pool, measure distance from the average image

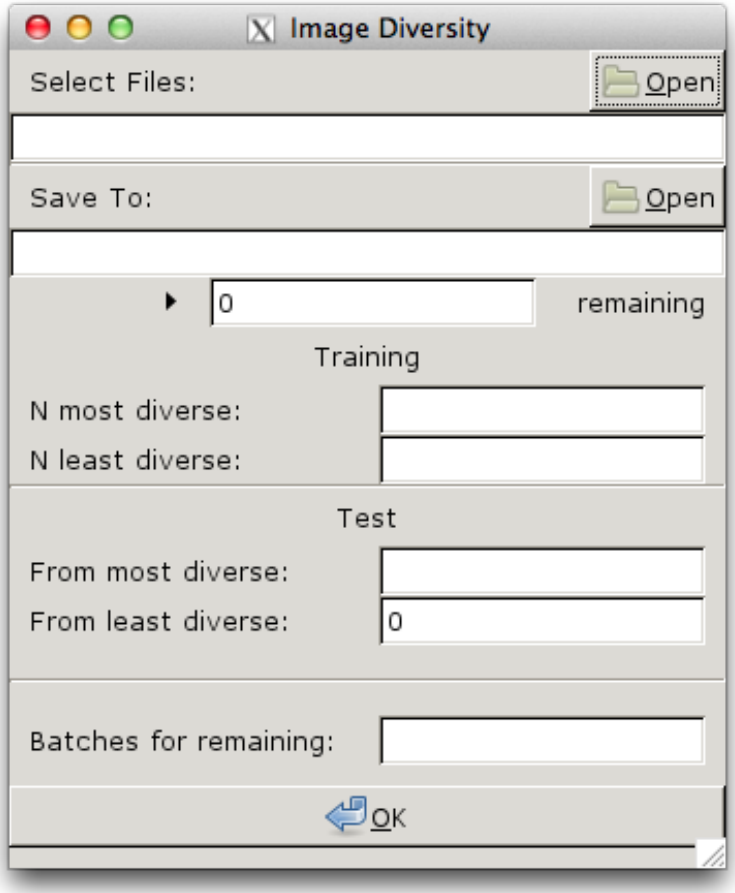
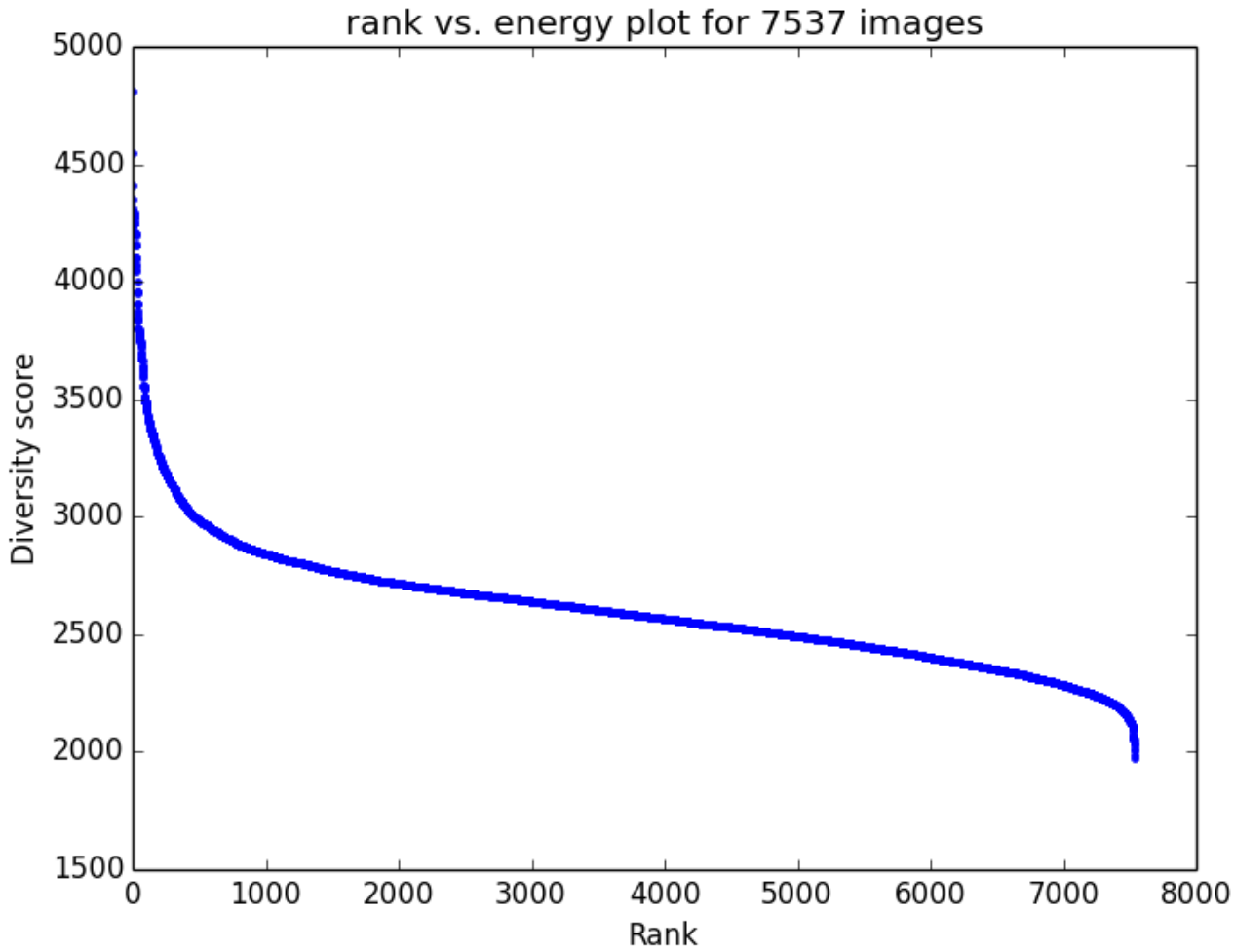


Figure: UI for generating diversity set

Most-Diverse Training V

5. Rank images



Most-Diverse Training VI

6. Select images
   - *MDI* most-diverse (most distant) images
   - *LDI* least-diverse (least distant) images
   - $MDI \gg LDI$

Most-Diverse Training VII

7. Trace selected images

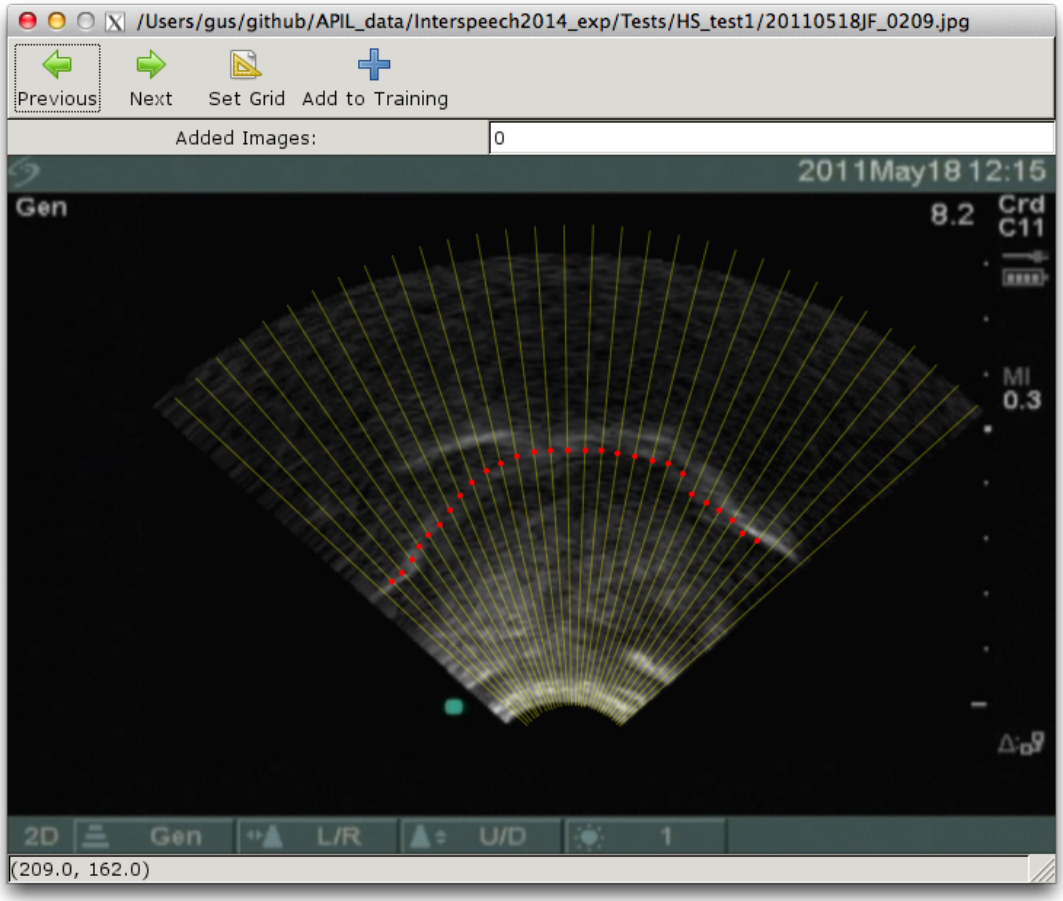


Figure: UI for tracing

Most-Diverse Training VIII

8. Prepare data directory

Most-Diverse Training IX

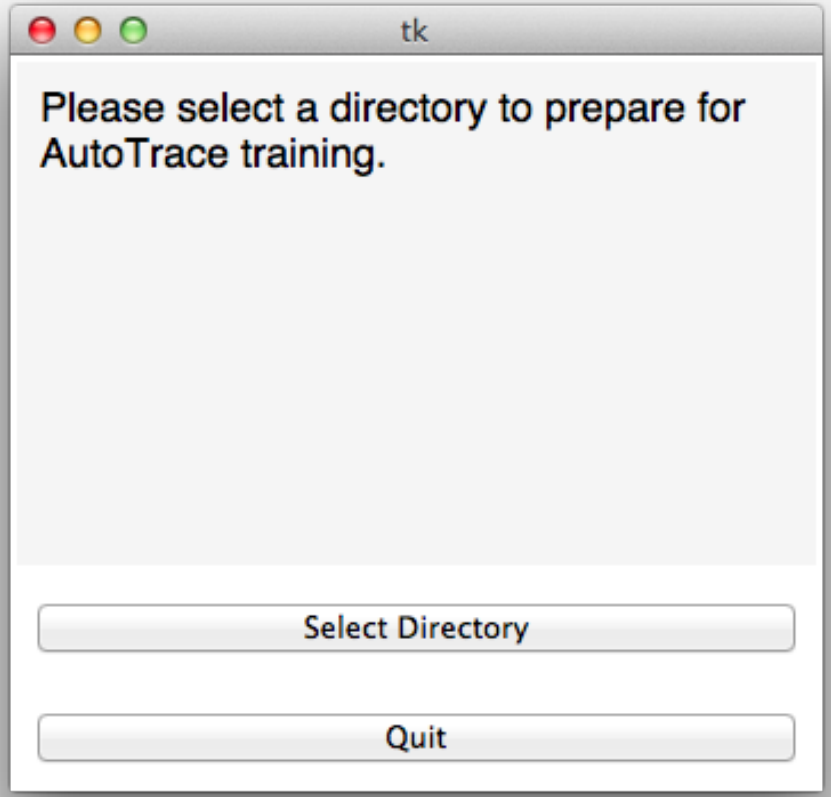
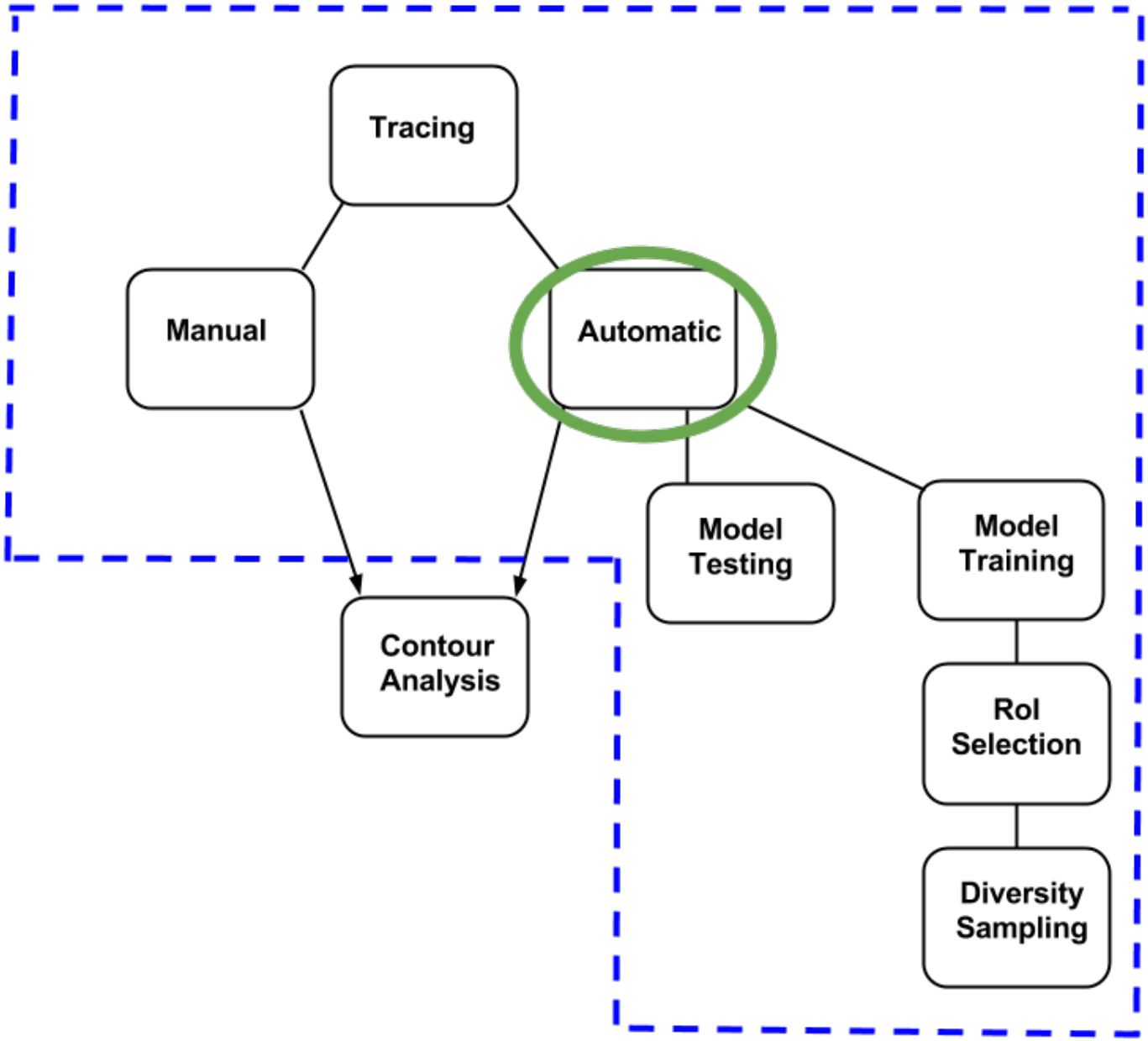


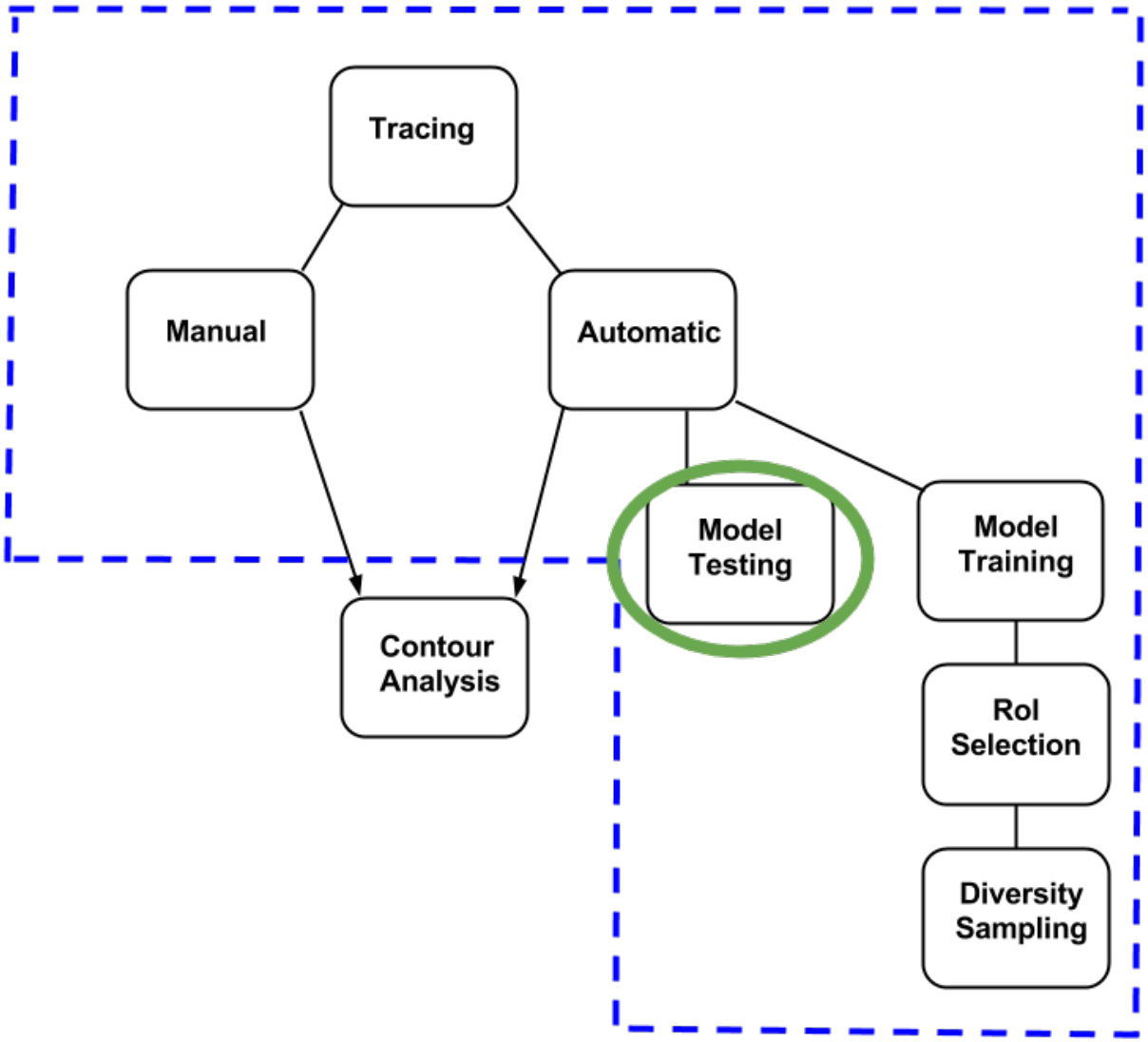
Figure: data folder is configured automatically via a Python script

9. Train network

Automatic Tracing



Evaluation I



Evaluation II

Compare Contours

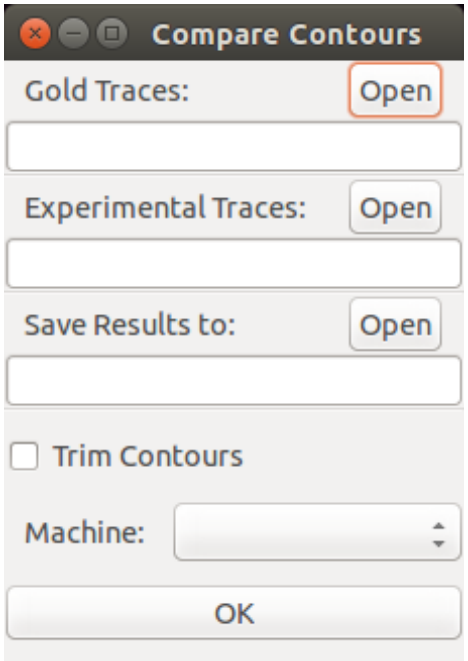


Figure: CompareContours.py

Evaluation III

Easy evaluation

- AutoTrace all of your remaining data
- Check automatically generated traces
- Manually correct some portion and evaluate

Evaluation IV

Evaluation

- use Mean Sum of Distances (MSD) metric to measure performance
  - the *lower* the MSD, the better performance
  - Li et al. (2005b)

Figure: Mean Sum of Distances

$$\mathrm{MSD}(U, V) = \frac{1}{2n}\left(\sum_{i=1}^{n} \min_{j} |v_i - u_j| + \sum_{i=1}^{n} \min_{j} |u_i - v_j|\right)$$

U and V are vectors representing pixel values at points along the tongue contour

Results I

- Sampling from both most and least is essential
- Performance positively correlated with size of most diverse set
  - As *MDI* increases, MSD decreases
  - *LDI* is kept constant

Results II

Table: Results for small data set

Most Diverse	Least Diverse	AutoTrace MSD
200	50	7.52
300	50	7.316
...	...	...
700	50	5.308
800	50	4.778

- *Our inter-annotator discrepancy*:
  - **4.077 MSD**
- *Discrepancy reported by Li et al. (2005a)*:
  - **2.47 - 3.77 MSD**
  - Two experts
  - three phrases

How to improve AutoTrace performance?

Iterative Retraining I

Use errors in retraining

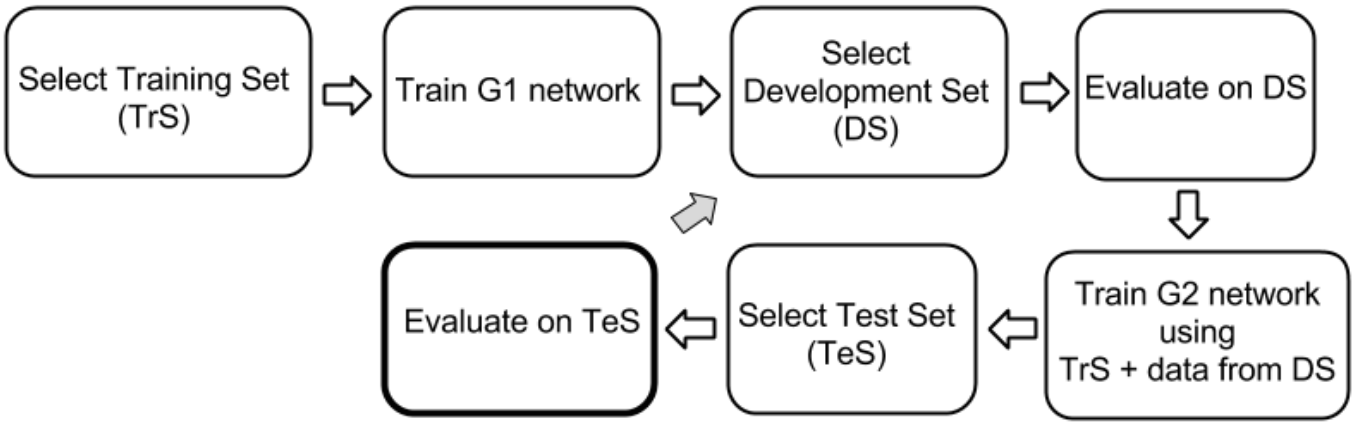


Figure: Network training procedure (see Berry et al. (2012) & Sung and Archangeli (2013))

What's left...

Image selection for training

- Alternative diversity measures
- Preprocessing images
- Improved RoI selection

Improving the network through retraining

- ASA poster

Identifying (and correcting) likely errors

- What traces are likely to be problematic?

Try it out yourself

Public Repository

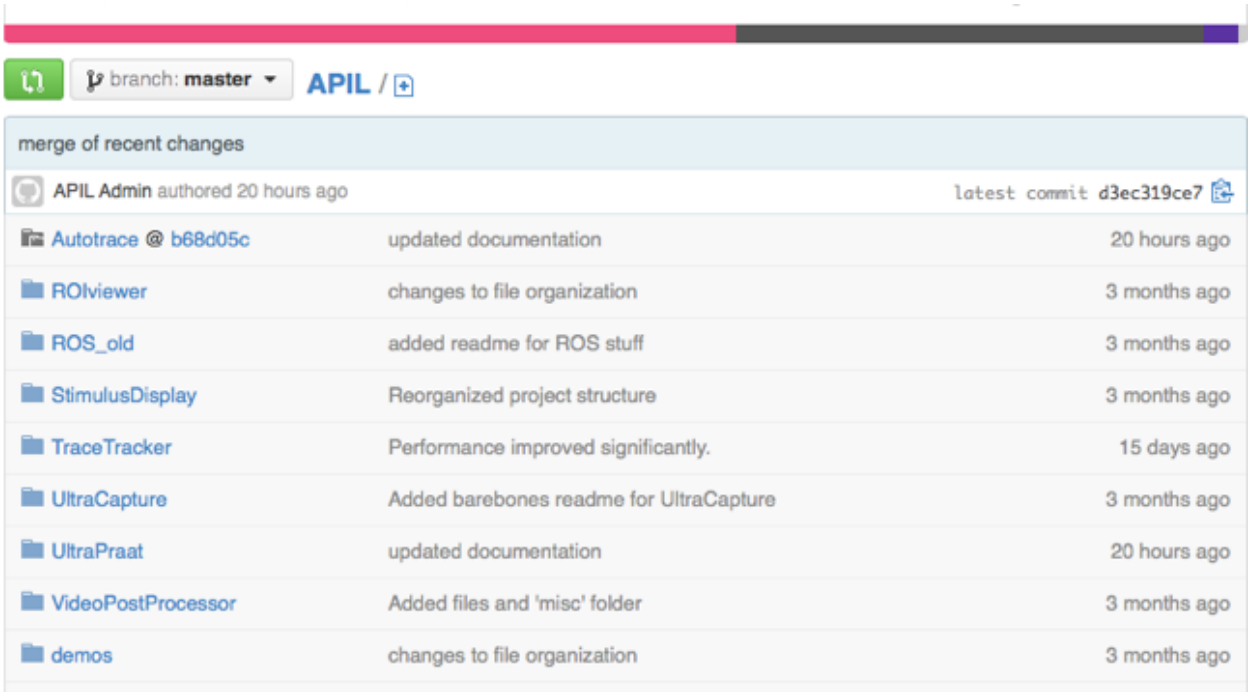
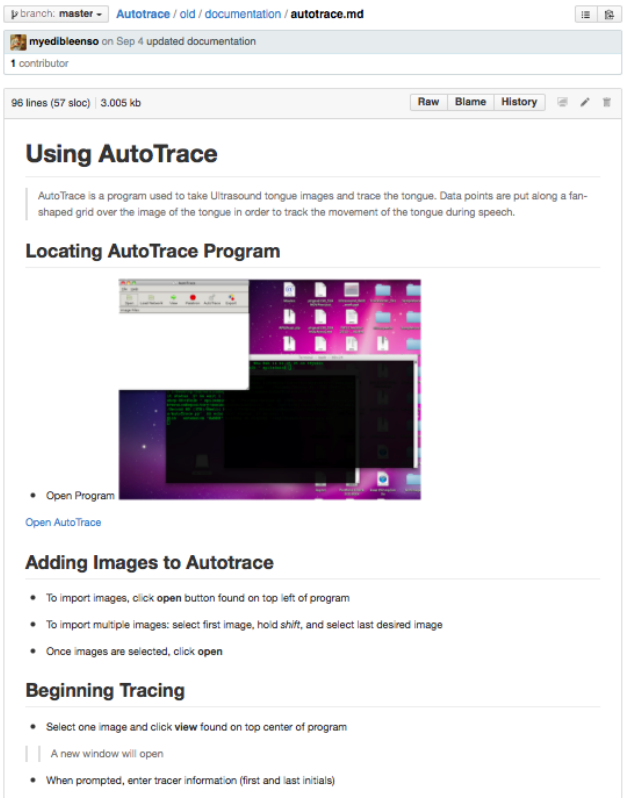


Figure: `https://github.com/myedibleenso/APIL`

We welcome contributions!

Documentation



branch: master · Autotrace / old / documentation / autotrace.md

myedibleenso on Sep 4 updated documentation

1 contributor

96 lines (57 sloc) | 3.005 kb — Raw | Blame | History

Using AutoTrace

> AutoTrace is a program used to take Ultrasound tongue images and trace the tongue. Data points are put along a fan-shaped grid over the image of the tongue in order to track the movement of the tongue during speech.

Locating AutoTrace Program

- Open Program

Open AutoTrace

Adding Images to Autotrace

- To import images, click **open** button found on top left of program
- To import multiple images: select first image, hold *shift*, and select last desired image
- Once images are selected, click **open**

Beginning Tracing

- Select one image and click **view** found on top center of program

> A new window will open

- When prompted, enter tracer information (first and last initials)

Figure: `https://github.com/jjberry/Autotrace/tree/master/old/documentation`

A work in progress...

Installing the AutoTrace toolkit

1. Install Matlab

Linux (Ubuntu)

2 Paste the following line into Terminal:

```
curl https://raw.githubusercontent.com/jjberry/
    Autotrace/master/ubuntu_autotrace_installer.
    sh > lui.sh; (sh lui.sh; rm lui.sh)
```

Mac OS X (Snow Leopard - Mavericks)

2 Install Xcode from the App Store

3 Paste the following line into Terminal:

```
curl https://raw.githubusercontent.com/jjberry/
    Autotrace/master/mac_autotrace_installer.py >
     mai.py; (python mai.py; rm mai.py)
```

Windows

- Limited support (no GUI)

Many thanks to...

National Science Foundation

- Grants 1059266 and 1244687

James D. McDonnell Foundation

- (grant awarded to Diana Archangeli)

References I

Berry, J. (2012). *Machine learning methods for articulatory data*. PhD thesis, The University of Arizona.

Berry, J., Fasel, I., Fadiga, L., and Archangeli, D. (2012). Training deep nets with imbalanced and unlabeled data. In *INTERSPEECH*.

Csapó, T. and Lulich, S. (2014). Tongue contour tracings from 2d ultrasound image sequences: quantification of measurement error using manual and automatic tracing methods. Unpublished manuscript.

Li, M., Kambhamettu, C., and Stone, M. (2005a). Automatic contour tracking in ultrasound images. *Clinical linguistics & phonetics*, 19(6-7):545–554.

Li, M., Kambhamettu, C., and Stone, M. (2005b). Automatic contours tracking in ultrasound images. *Clinical Linguistics and Phonetics*, 19:545–554.

Liu, X.-Y., Wu, J., and Zhou, Z.-H. (2009). Exploratory undersampling for class-imbalance learning. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(2):539–550.

Sung, J.-H., B. J. C. M. H.-P. G. and Archangeli, D. (2013). "testing autotrace: A machine-learning approach to automated tongue contour data extraction". Edinburgh. UltraFest VI.

Questions?

hahnpowell@email.arizona.edu