IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 September 2022, accepted 9 October 2022, date of publication 19 October 2022, date of current version 25 October 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3215972*

APPLIED RESEARCH

# Audiogram Digitization Tool for Audiological Reports

**FRANÇOIS CHARIH AND JAMES R. GREEN, (Senior Member, IEEE)**
Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada

Corresponding author: James R. Green (jrgreen@sce.carleton.ca)

This work was supported by the Workplace Safety Insurance Board of Ontario.

**ABSTRACT** Multiple private and public insurers compensate workers whose hearing loss can be directly attributed to excessive exposure to noise in the workplace. The claim assessment process is typically lengthy and requires significant effort from human adjudicators who must interpret hand-recorded audiograms, often sent via fax or equivalent. In this work, we present a solution developed in partnership with the Workplace Safety Insurance Board of Ontario to streamline the adjudication process. We present a flexible and open-source audiogram digitization algorithm capable of automatically extracting the hearing thresholds from a scanned or faxed audiology report as a proof-of-concept. The algorithm extracts most thresholds within 5 dB accuracy, allowing to substantially lessen the time required to convert an audiogram into digital format in a semi-supervised fashion, and is a first step towards the automation of the adjudication process. The source code for the digitization algorithm and a desktop-based implementation of our NIHL annotation portal is publicly available on GitHub https://github.com/GreenCUBIC/AudiogramDigitization.

**INDEX TERMS** Machine vision, pattern recognition, deep learning, audiology.

## I. INTRODUCTION

Noise-induced hearing loss (NIHL) is a common consequence of long-term exposure to noise in the workplace. In fact, a Canadian study [1] recently found, through a series of over 3,500 interviews, that 42% of respondents were exposed to hazardous levels of noise in the workplace. Moreover, 20% of respondents who reported being asked to wear hearing protective equipment by their employer admitted to not following this rule consistently. It is therefore not surprising that numerous occupational NIHL-related claims are received by public and private insurance companies yearly.

The Workplace Safety Insurance Board of Ontario (WSIB) reports receiving several thousands of NIHL-related claims every year which can take several months to process. The audiological reports received must be carefully interpreted by adjudicators who apply a series of rules to determine the eligibility of the claim. This is a time-consuming process that contributes to the lengthy adjudication process.

The audiogram is a critical component of a NIHL-related claim. An audiogram plots the hearing *threshold* (minimal perceivable amplitude) in dB across a range of frequencies. Different standard audiological symbols are used to indicate whether a hearing threshold correspond to the left or right ear, whether the threshold was obtained through air or bone conduction, and whether masking was used or not to prevent the non-test hear from hearing tones delivered to the contralateral ear. The shape of this audiometric curve is pivotal in establishing the etiology of the hearing loss. For instance, individuals with NIHL tend to have a notch in their air and bone conduction audiometric curves (worse hearing) between 3,000 and 6,000 Hz [2].

While audiometers are now fully capable of generating digital versions of audiograms, many hearing professionals still plot audiograms by hand in audiological reports. While these reports differ slightly in their layout and content, they typically consist of a single page with one or two audiogram plots (combined or separated by ear), a brief summary of the findings, and potentially, results from other tests (*e.g.*, tympanogram, *etc.*).

Because the reports are received by fax or as image files sent electronically, eligibility rules can only very rarely be directly applied to reports submitted to insurance companies or compensation boards. As such, an algorithm capable of

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague.

automatically extracting hearing thresholds from the audiograms on scanned or faxed audiological reports to add to the client's electronic record could significantly reduce the time required to process a claim.

In this work completed in partnership with the WSIB, we present an audiogram digitization algorithm that combines traditional image processing and deep neural networks trained with transfer learning to digitize audiograms. We outline how the training data for this algorithm were collected, the training procedure, and show how the algorithm can be useful in the adjudication of NIHL-related claims.

## II. RELATED WORK

Automated audiogram acquisition and interpretation has been an active area of research over the past two decades. Several automated audiometry tools were developed and marketed to counterbalance the predicted gap between the availability and need for hearing professionals [3]. For instance, multiple tablet-based audiometers were designed to make hearing assessment possible outside a soundproof booth [4], [5], [6]. These mobile audiometers are now deployed worldwide to afford access to basic hearing assessments to people in remote locations where hearing professionals are few or absent.

To generate actionable insight, the audiograms generated with these automated tools must be interpreted by a trained hearing professional or some other thoroughly validated automated method. Useful information that can be extracted from audiograms include the type of hearing loss (*e.g.*, sensorineural, conductive or mixed), the severity of the hearing loss, and whether there is a need for immediate or long-term intervention(s), for example. A number of audiogram interpretation models have been developed to address the challenge of automatically extracting such information from audiograms. One of the first well-known methods to achieve this is the AMCLASS system [7]. It is a rule-based system that was designed by hand to classify the hearing loss by severity, symmetry, configuration and type of hearing loss. Several groups leveraged large audiological databases and clustering-based approaches to generate a set of ''canonical'' audiograms against which new ones can be compared [8], [9]. More recently, we described a data-driven model trained on hundreds of expert-annotated audiograms to take an audiogram as an input, and generate a concise summary of the severity, configuration and symmetry across ears of the hearing loss [10]. Crowson et al. trained and compared several convolutional neural network architectures for the task of identifying the type of hearing loss from *undigitized* audiogram images, achieving quite impressive results (>95% accuracy) [11]. It is unclear whether their model generalizes well, however, given that all the audiograms were generated at the same instution. They argue that bypassing the digitization step simplifies the process of extracting insight from audiograms, but this comes at a cost in terms of flexibility. For example, it makes it more difficult to classify audiograms based on a set arbitrary rules defined by an institution. It is true that a model trained on audiogram images can implicitely learn the rules from a large number of audiograms annotated with the rules, but this will not outcompete the perfect accuracy of digitized audiograms upon which classification rules are applied. Furthermore, if digitization is accurate on a wide variety of audiology reports, then subsequent audiogram interpretation tasks (*e.g.*, hearing loss type identification, determination of the best intervention, hearing aid tuning, *etc.*) become agnostic to the actual way the audiogram was plotted (*i.e.*, handwritten, grid dimensions, image quality, etc.), and therefore require fewer audiograms to train.

These automated audiogram classification models typically, though not always [11], rely on digital audiometry data to analyze the subject's hearing, hence the need for audiogram digitization tools capable of accurately turning images of audiograms into a list of thresholds are often needed. The only other method that we are aware of that can achieve this is the *Multi-stage Audiogram Interpretation Network (MAIN)* of Li et al., 2021 [12]. The authors trained several convolutional neural networks to extract audiograms, symbols and axis labels from audiogram images, to finally reconstruct the audiogram digitally.[1] There are a couple limitations associated with MAIN. First, it can only process unmasked air conduction thresholds, *i.e.*, only two of the eight commonly encountered types of measurements. Second, it was trained and tested on a small dataset called *Open Audiogram* consisting of 64 unique audiograms, all generated with the same system. Multiple photos of each audiogram were taken in different conditions to augment it. MAIN will require validation on a larger and more diverse dataset to demonstrate its generality.

## III. DIGITIZATION STRATEGY

In order to extract hearing thresholds from an audiological report, the following elements (shown in Figure 1) must be located and identified:

1) **Audiogram:** to identify the region(s) of interest in the report.
2) **Axis labels and grid lines:** important for the conversion of pixel coordinates to frequency-threshold coordinates.
3) **Audiological symbols:** the hearing thresholds of the individual.

A few techniques were considered to tackle the object detection problem. For instance, we initially considered applying template matching to locate the audiological symbols within the report, but rapidly realized that several problems made this approach less than ideal. First, the symbols are often drawn by hand, and a given symbol (*e.g.*, a cross) will be drawn differently depending on the person recording the audiogram. Next, overlapping symbols are a frequent occurrence, and as such, it is not uncommon to see 2 or 3 overlapping symbols. For these reasons, the number of templates

[1] The approach of Li et al., 2021 shares striking similarities with ours, and was uploaded to the arXiv pre-print server in the time between the drafting of this paper and its submission. We were not aware of it as we were developing our own method.

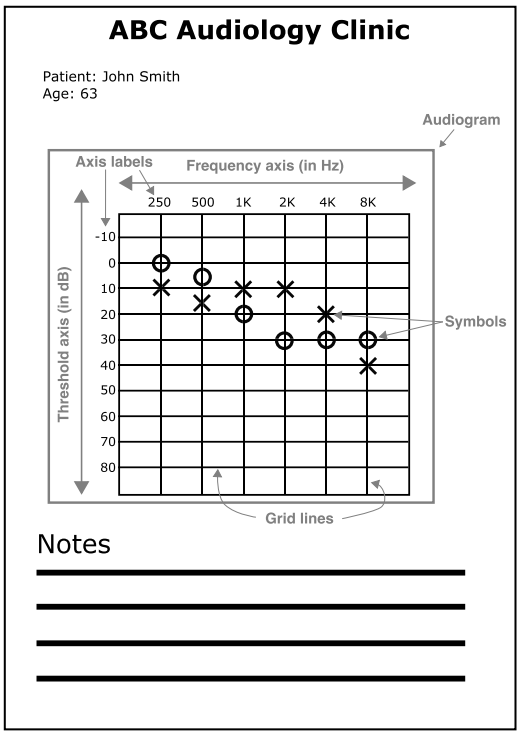


**FIGURE 1.** **Components of interest within a typical audiology report. A typical audiology report contains general information about the patient, the audiogram, and a section for notes. Occasionally, the report will also contain tympanometry data in tabular or graphical form. The elements of interest that are extracted for the purpose of this work are identified in gray.**

that would be required to handle all possible symbols and symbol combinations would be prohibitively large.

We finally settled for the strategy illustrated in Figure 2. This strategy combines transfer learning (object detection) and traditional image processing techniques. Pre-trained deep convolutional neural networks are fine-tuned for detection of audiograms, axis labels, and symbols within a report. Line detection techniques are applied to correct for audiogram rotation and to compute the pixel-to-frequency/threshold transform. The algorithm is described in greater details later in this paper.

## IV. DATA COLLECTION

Knowing that transfer learning would be employed to train object detectors for elements composing the audiological report, we sought to assemble a dataset of several thousands of annotated audiological reports.

The WSIB provided a large dataset of approximately 3,200 anonymized audiological reports from claims received in 2006 or later. The reports all consisted of a single page on which the audiograms were drawn in one or two plots. A qualitative assessment of the dataset revealed a highly heterogeneous dataset where reports varied greatly in terms of layout, resolution, handwriting, and completeness.

To collect the annotations required to develop and evaluate the strategy presented previously, we developed the *NIHL Portal*, a web portal specifically tailored

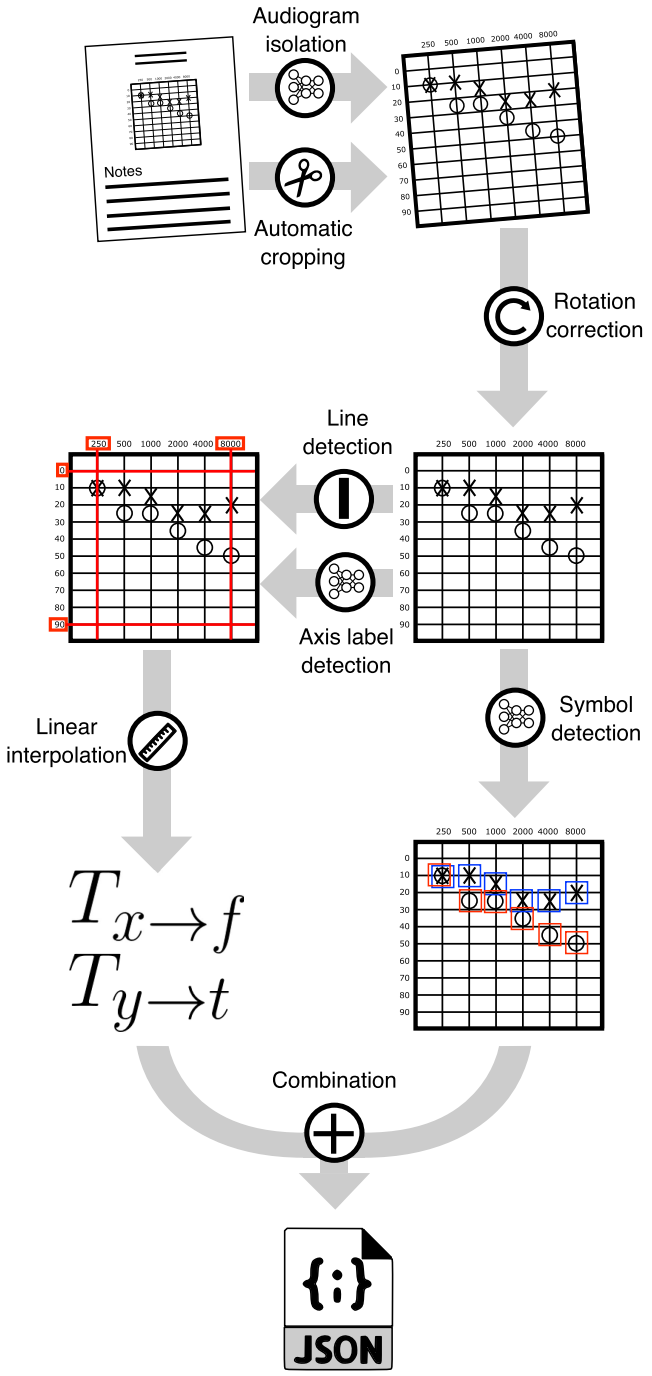


**FIGURE 2.** **Audiogram digitization strategy. Our strategy combines a series of deep learning-based object detection tasks with traditional image processing procedures (line detection, cropping, rotation). These steps enable the derivation of pixel-to-frequency and pixel-to-threshold transforms through linear interpolation. These transforms are subsequently applied to the pixel coordinates of detected audiological symbols to generate a JSON document listing the individual hearing thresholds.**

to collect the required data rapidly and ergonomically. The user interface was developed using React.js (https://reactjs.org) while its backend was implemented using Flask (https://flask.palletsprojects.com) as the server and PostgreSQL (https://www.postgresql.org/) as the database.

```
[
  {
    "boundingBox": {
      "x": number,
      "y": number,
      "width": number,
      "height": number
    },
    "corners": [
      {
        "x": number,
        "y": number,
        "position": {
          "vertical": string,
          "horizontal": string
        },
        "frequency": number,
        "threshold": number,
      }, ...
    ],
    "labels": [
      {
        "boundingBox": {
          "x": number,
          "y": number,
          "width": number,
          "height": number
        },
        "value": string
      }, ...
    ],
    "symbols": [
      {
        "boundingBox": {
          "x": number,
          "y": number,
          "width": number,
          "height": number
        },
        "response": boolean,
        "measurementType": string
      }, ...
    ]
  }, ...
]
```

**FIGURE 3.** Schema of a JSON annotation produced by the NIHL Portal. Annotations are stored as an array of one or two items, one for each audiogram in the report. The locations of the audiogram bounding box, corners of the audiogram, axis labels, and symbols are collected.

Annotators were asked to draw, using the interface shown in Figure 4, bounding boxes for the audiogram, the axis labels, and the individual threshold symbols. Annotators were also asked to indicate the location of the audiogram corners along with the corresponding frequency/threshold coordinates, so that a ground truth audiogram could be computed from the annotation. The annotation process was completed over a period of approximately 4 months. The JSON schema of the annotations collected is presented in Figure 3.

## V. DIGITIZATION ALGORITHM

### A. AUDIOGRAM DETECTION MODEL AND ROTATION CORRECTION

We used the YOLOv5s architecture developed by Ultralytics [13] pre-trained on the COCO dataset [14] to train an audiogram detection model. The main purpose of this component of the digitization algorithm is to isolate the audiogram(s) in the report so as to: 1) determine whether there are audiograms in the report and 2) prevent the detection of symbols that are outside the bounds of an audiogram plot.

The audiogram detection model was fine-tuned as per the instructions described by the architecture's authors [13] on a computer with a NVIDIA P100 GPU and 64G RAM. We used 80% of the 3,000 reports for training, while the remaining 20% was used for validation. We set 206 reports aside for testing. The model was trained for 100 epochs, and we selected the model that had the lowest generalized intersection over union (GIoU) [15] in validation. The procedure was repeated 3 times with different seeds to emulate 3-fold cross-validation, allowing us to better estimate the performance of the model and to quantify the uncertainty in that estimate.

When deployed, this model is used to crop the image of a report around the audiogram(s). It is often the case that the orientation of the audiogram plot must be corrected to make its grid lines horizontal and vertical. To this end, we framed the problem of rotation correction as an optimization one.

The first step of the procedure involves detecting lines in the audiogram with the conventional Hough Transform [16]. Lines that do not intersect another line roughly perpendicularly ($\pm 1°$) are not considered by the algorithm, as they are unlikely to belong to the audiogram grid. The correction angle, $\theta_{corr}$, is then obtained by minimizing the sum of the deviations of all lines from the horizontal (0°) or vertical axis (90°), whichever is closest in terms of angle:

$$\underset{\theta_{corr}}{\operatorname{argmin}} \left( \sum_{v \in V} |90° - (\theta_v + \theta_{corr})| + \sum_{h \in H} |(\theta_h + \theta_{corr})| \right) \quad (1)$$

where $V$ is the set of lines assumed to be vertical in the actual, unaltered report, i.e. $\theta_v \in (45°, 135°)$, and $H$ is the set of all other lines, assumed to be horizontal in the original, unaltered report, i.e. $\theta_h \in (-45°, 45°)$.

### B. AUDIOLOGICAL SYMBOL DETECTION

The audiological symbol detection model was trained similarly. We used 2803 *audiograms* (not reports) for training/validation, and 273 audiograms for testing, and repeated the experiment 3 times with different seeds. We defined a total of 8 different classes corresponding to different audiological symbols accounting for the 4 different types of measurement (Table 1).

**TABLE 1.** Relevant audiological symbols considered in this study.

| | Measurement type | | Ear | |
|---|---|---|---|---|
| | Conduction | Masking | Left | Right |
| 1 | Air | No | × | ○ |
| 2 | Air | Yes | □ | △ |
| 3 | Bone | No | > | < |
| 4 | Bone | Yes | ] | [ |

Once trained, the model can detect the symbols and provide the pixel coordinates of the center of any detected symbol's

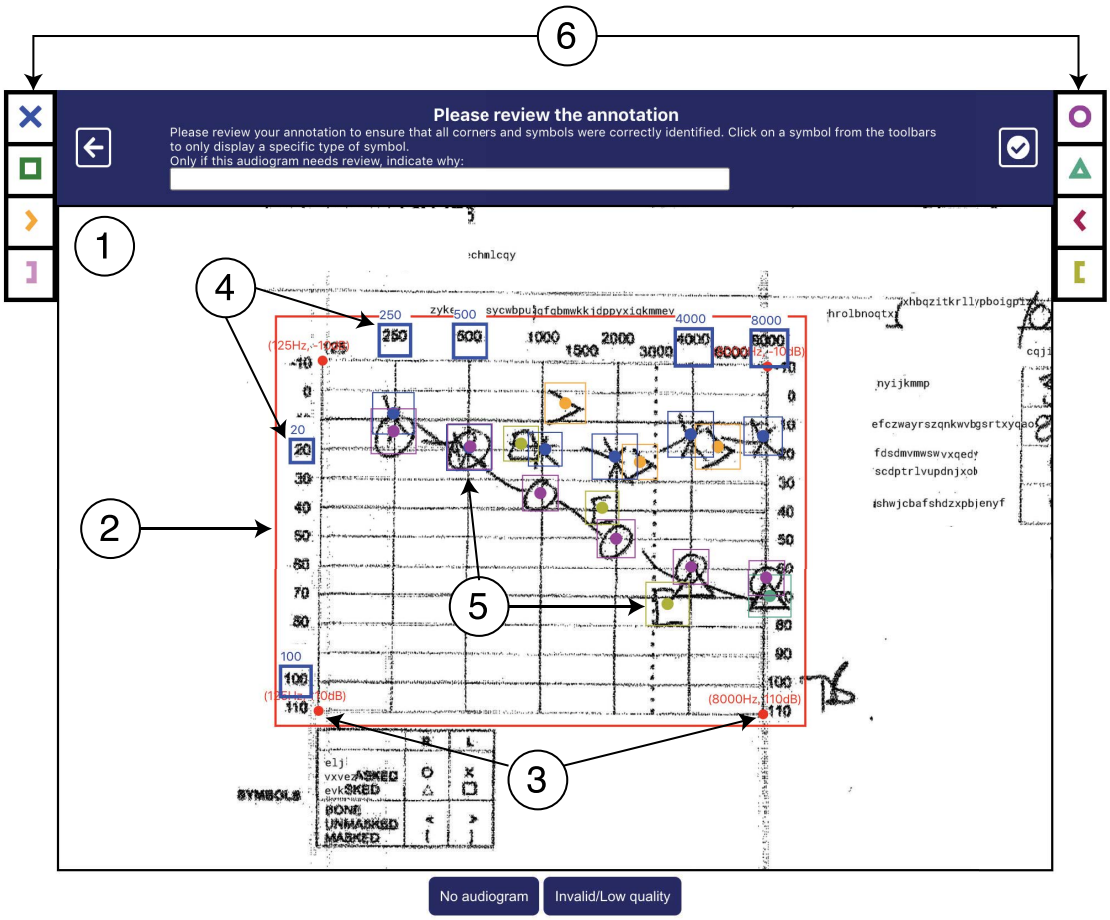


**FIGURE 4.** NIHL Portal. The annotator is shown the image of an audiological report in the viewport (1) and is asked to draw the bounding box around the audiogram (2). The annotator then proceeds to annotate the outermost corners (3), the axis labels (4), and the audiological symbols (5) by selecting the appropriate symbols from a tool bar (6).

bounding box. However, to be clinically meaningful, these pixel coordinates must be converted into frequency-threshold pairs.

### C. AXIS LABEL DETECTION

Given that the audiogram grids vary slightly in layout from one report to another, we sought to identify the axis labels so that they could be associated to the lines that make up the audiogram grid. This association allows for the derivation of pixel-to-frequency and pixel-to-threshold mappings. The derivation of these transforms using linear interpolation requires a minimum of two correct grid-line associations per axis. Ideally, these grid-line associations are at the opposite ends of the axes.

We initially attempted to apply optical character recognition to detect axis labels using Google's open-source Tesseract engine [17], but failed to obtain reliable results despite our best efforts to preprocess the images by adjusting parameters such as the contrast and brightness or by applying techniques like thresholding and dilation/erosion. However, given that the set of labels encountered is finite, we framed this as an object detection problem to be solved in the same way as audiogram detection, and collected axis label annotations for 506 *audiograms* (not reports), and used 465 for training/validation and 41 for testing. In the same fashion as for the other two predictors, the process was repeated three times with different seeds.

We fine-tuned the same YOLOv5s model on a dataset of audiograms with annotations for the following frequency axis labels: 250 Hz, 500 Hz, 4,000 Hz and 8,000 Hz. We also included classes for equivalent representations of the same frequencies (*e.g.*, "0.25", "0.5", "4K", "8", *etc.*). The classes for the decibel axis labels included "20", "60", "80" and "100". We replicated the procedure described in the previous section to train the axis label detection model.

### D. DERIVING AND APPLYING THE PIXEL DOMAIN-TO-AUDIOLOGICAL DOMAIN TRANSFORMS

While it is trivial for human interpreters to identify the coordinates of thresholds in the *audiological domain* (in terms of frequencies and threshold values) by visual inspection of the symbol and the surrounding axes, this task is far from trivial for computers, which operate in the *pixel domain*. One could presume that given enough annotated audiograms, deep

neural networks could be trained to extract this information from images of audiograms. However, given the wide variety of audiograms one may encounter, all unique in terms of their layout, font, size, aspect ratio, it is highly unlikely that a couple of thousands of audiograms would suffice.

Fortunately, one can make use of visual landmarks in the audiogram to derive the transforms that convert pixel values along the $x$ axis to frequencies (in Hz) and pixel values along the $y$ axis as thresholds (in dB). The most relevant landmarks are the axis labels and the audiogram grid lines. *Isn't that how we, humans, interpret this type of visual information after all?*

To derive the pixel-to-threshold transform (i.e. the $y$ axis), we pair detected axis labels with the horizontal line ($\pm 1°$) that intersects it and that is closest to the center of its bounding box. The lines are detected with the Hough transform [16]. From this, we may generate a sorted list of tuples of the form

$$\{(y_1, t_1), \ldots, (y_n, t_n)\} \tag{2}$$

where $y_i$ is the $y$ coordinate (in pixels) of the $i$-th horizontal line and $t_i$ is the threshold value (in dB) of the associated label.

Provided that we are able to make two such associations, a transform can be derived. If more than two such associations exist, the ones that are farthest apart are used for increased resolution. The transform $T_{y\to t}$ is simply:

$$T_{y\to t}(y) = t_1 + \frac{(t_n - t_1)(y - y_1)}{y_n - y_1} \tag{3}$$

We may proceed similarly for the pixel-to-frequency transform along the $x$ axis, except that one must account for the logarithmic nature of the frequency scale. To do so, we use the linear *octave scale*, where 125 Hz is the $0^{th}$ octave, 250 Hz is the $1^{st}$ octave, 500 Hz is the $3^{rd}$ octave, and so on. We can convert the frequency to its octave value with the equation:

$$o(f) = \frac{\ln(f/125)}{\ln(2)} \tag{4}$$

where $f$ is the frequency and $o(f)$ is the octave value of frequency $f$.

Then proceeding similarly as for the pixel-to-threshold transform derivation, we may generate a sorted list of tuples by associating $x$ axis labels with the vertical lines ($\pm 1°$):

$$\{(x_1, o_1), \ldots, (x_n, o_n)\} \tag{5}$$

where $x_i$ is the $x$ coordinate (in pixels) of the $i$-th horizontal line and $t_i$ is the octave value of the associated frequency label.

Then, we may use linear interpolation to derive the pixel-to-frequency tranform $T_{x\to f}$, which converts back the octave value generated into a frequency value by applying the reciprocal of Equation 4:

$$T_{x\to f}(x) = 125 \times 2^{o_1 + \frac{(o_n - o_1)(x - x_1)}{x_n - x_1}} \tag{6}$$

The frequency-threshold pairs generated by applying these transforms to the coordinates of detected symbols virtually always yield values that approximate the clinical measurements. Knowing that thresholds are measured at a set of standard octave or semi-octave frequencies and that thresholds are measured in increment of 5, we may correct the measurement by "snapping" the frequency value to the nearest standard octave or semi-octave and the threshold value to the nearest increment of 5. For instance, an initial measurement of 53.3 dB at 1,136 Hz would be corrected to 55 dB at 1,000 Hz. The rounded measurements can then be used to populate a list of thresholds in JSON format (Figure 5).

```
[
  {
    "frequency": number, // in Hz
    "threshold": number, // in dB
    "masking": bool,
    "ear": string, // "left" or "right"
    "conduction": string // "air" or "bone"
  }, ...
]
```

**FIGURE 5.** JSON schema of a digitized audiogram. The JSON document produced by the algorithm is a list of JSON objects that describe the threshold of hearing, which ear was tested, the measurement type (air or bone) as well as whether masking was used or not for each frequency tested.

## VI. RESULTS AND DISCUSSION

### A. PERFORMANCE OF INDIVIDUAL REPORT COMPONENT DETECTORS

To evaluate each independent object detector, we used a 3-fold cross-validation-like strategy. We computed the mean recall, mean precision and mean average precision at 0.5 IoU (mAP@0.5) for each model. The metrics are averaged over the different classes for the axis label and symbol models (Table 2).

**TABLE 2.** Performance of the individual object detection models in 3-fold cross-validation.

| | Model | | |
|---|---|---|---|
| | Audiogram | Axis Label | Symbol |
| Recall | $1.00 \pm 0.00$ | $0.70 \pm 0.12$ | $0.83 \pm 0.04$ |
| Precision | $1.00 \pm 0.00$ | $0.79 \pm 0.14$ | $0.86 \pm 0.04$ |
| mAP@.5 | $0.84 \pm 0.01$ | $0.34 \pm 0.06$ | $0.39 \pm 0.01$ |

Unsurprisingly, the easiest task is that of detecting audiograms within an audiology report. The model trained to do so achieved a very high performance, with perfect recall and precision.

Symbol detection was more difficult. Our symbol detection model achieved an mAP@0.5 of $0.39 \pm 0.01$. The difficulty of this task was not unforeseen, as the symbols are often hand drawn and overlapping, which leads to high variability within a single report (by the same hearing clinician) and from one report to another (between hearing clinicians).

It is the axis label detection model that achieves the lowest performance. While it is true that the font used on the grids varies from one report to another and that this may adversely affect performance, it seems more likely that the scarcity of

reports containing label annotations used in training (463) is responsible for this.

### B. END-TO-END PERFORMANCE OF THE DIGITIZATION ALGORITHM

To test the end-to-end performance of the digitization algorithm, we ran the complete digitization algorithm on the 206 reports that were not used for training. We computed the precision and recall of the digitization algorithm for every report that could be successfully digitized as follows:

$$Pr = \frac{|\mathcal{T}_{detected} \cap \mathcal{T}_{actual}|}{|\mathcal{T}_{detected}|} \tag{7}$$

$$Sn = \frac{|\mathcal{T}_{detected} \cap \mathcal{T}_{actual}|}{|\mathcal{T}_{actual}|} \tag{8}$$

where $\mathcal{T}$ is a set of thresholds, each having a symbol (indicating the measurement type, see Table 1), a frequency, and a threshold value in decibels.

The digitization algorithm had a mean precision of $0.66 \pm 0.35$ and a mean recall of $0.64 \pm 0.35$ over the remaining reports. The distribution of precision and recall is shown in Figure 6A. In total, only 16 reports (7.8%) were digitized perfectly. However, these metrics only capture a small part of the story, as a threshold off by 5 dB is classified incorrect while in many cases, it may not be clinically meaningful. One may wonder why the recall and precision follow each other so closely. This is because the algorithm rarely fails to detect a symbol. It is far more common for it to assign an incorrect threshold value to a symbol that actually exists. As a result of the way the precision and recall are computed, if a symbol is given the wrong threshold value, it will be counted both as a false positive (a threshold that does not exist is predicted) *and* a false negative (actual threshold is missed).

Given that these metrics are strict and do not distinguish between digitized values that are off from the actual value by 5 dB from those that are off by 30 dB, we computed the distribution of distances in dB between the actual and digitized values (Figure 6B). The majority of the incorrectly extracted thresholds (52%) are off by only 5 dB.

Factors contributing to the difficulty of the task include the low resolution of the images in our dataset. The fact that the digitization algorithm follows largely a succeed-or-fail trend is mostly caused by a lack of robustness of the grid extraction step which relies on the Hough transform and axis label detection to determine the pixel-to-threshold and pixel-to-frequency transforms. As mentioned previously, failure to properly derive the transforms leads to inaccurate frequencies and/or threshold values for all the detected symbols within a report. Moreover, hearing clinicians are inconsistent in how they report bone conduction thresholds. The widely respected convention for air conduction thresholds is to put the symbol directly on top of the vertical line indicating a frequency. However, for bone conduction measurements some write down the symbols for the left ear (> and ]) to the right of the frequency line and the symbols for the right ear (< and [) to the left of the frequency line,

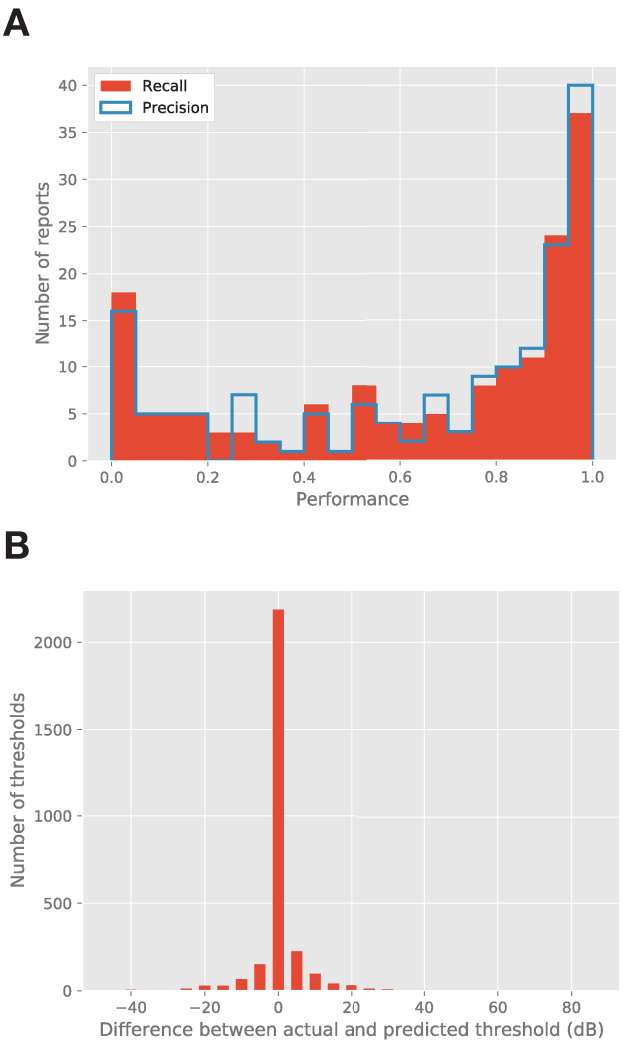


**FIGURE 6. Performance of the algorithm in end-to-end digitization. (A) The recall and precision distribution of the algorithm over 163 successfully digitized reports. (B) The distribution of distances between the actual threshold values and those obtained via digitization.**

while others follow the same convention as for air conduction thresholds. Errors in snapping these symbols to the correct frequency adversely affects performance.

### C. ANALYSIS OF DIGITIZATION FAILURES

Errors in one of the modules composing a multi-stage model may lead to inaccurate results or complete failure. The identification of bottlenecks or points of failure is key to understanding how to improve the overall model.

The most critical step in the algorithm is arguably the derivation of accurate pixel-to-frequency and pixel-to-threshold transforms. These transforms rely on the accurate detection of at least 2 grid lines per axis and their associated labels. This step is critical, because failure at this step will affect the conversion of *all* symbol coordinates to frequency-threshold pairs downstream. Poor image resolution of the image and the variety of font types used to label the axes make this step the most challenging. In fact, of

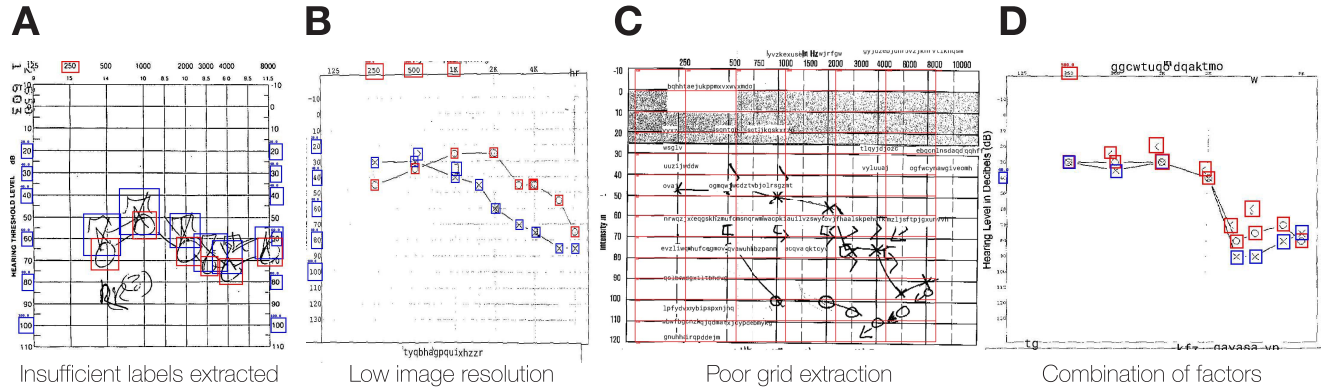


**FIGURE 7.** Typical sources of audiogram digitization failure. (A) An audiogram where only one axis label could be detected, leading to the inability to derive a pixel-to-frequency map. (B) An audiogram where the low scanned image resolution prevented the detection of grid lines. (C) Audiogram where the grid could not be accurately extracted (along the frequency axis), likely because of the fuzzy region denoting normal hearing between 0 and 20 dB. (D) An audiogram where multiple factors led to digitization failure, *i.e.*, low resolution, absence of grid lines, and an insufficient number of axis label detections along the frequency axis.

the 206 audiology reports used for testing, there were 43 (21%) for which the grid could not be extracted. This occurred when the transforms could not be derived because too few axis labels (Figure 7A), or too few grid lines were detected in images of poor resolution (Figure 7B). These two scenarios typically led to complete failure, *i.e.*, prevented *any* threshold from being extracted, even though the symbols are correctly detected. Occasionally, grid lines were incorrectly extracted (Figure 7C), leading to inaccurate digitization. Finally, the algorithm failed on reports where multiple of these sources of failure occurred; for example, when the resolution of the report scan was so low that neither the axis labels nor the grid lines could be extracted correctly (Figure 7D).

### D. COMPARISON WITH MAIN

We ran the pre-trained MAIN digitization model developed by Li et al. [12] on the same 206 audiology reports. It was unable to extract *any* thresholds for 116 (56%) of these reports. This occurred more than twice as frequently as for our own method (21%). We also observed that, in contrast to our own method, theirs does not include logic to deal with duplicate detections, *i.e.*, two or more thresholds detected for the same ear and frequency. For this reason, many of the reports that were in fact digitized contained up to three or four hearing threshold values detected for the same ear and frequency. This makes a fair assessment of precision and recall virtually impossible.

These observations indicate that MAIN did not fully generalize to our dataset. This was reasonably foreseeable, given that it was trained on a relatively small dataset of audiograms generated using a single audiology software system.

Upon closer inspection of the results, we observed that their symbol detection network failed to detect several symbols that it was trained to detect (Figure 8), *i.e.*, the circle (○) and the cross (×). The symbol detector appears to lack the ability to correctly detect symbols when they are overlapping. Their axis label detection network appeared to

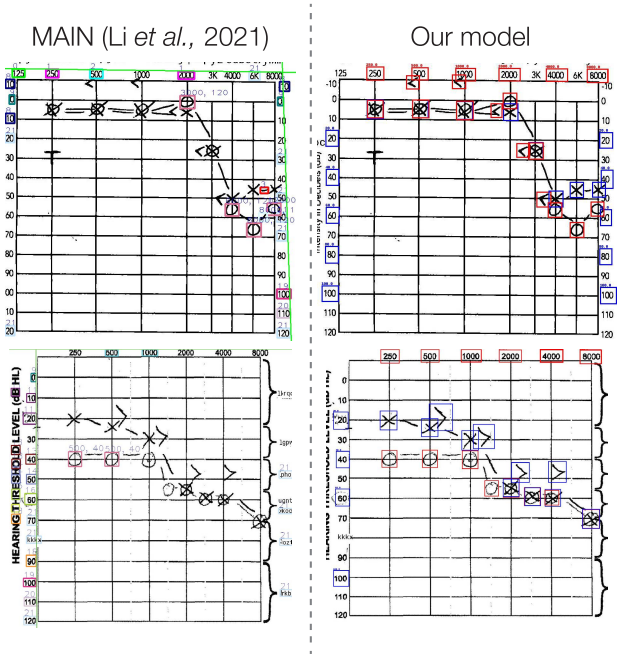


**FIGURE 8.** Comparison of the components detected within representative audiograms. The axis labels and audiological symbols detected by Li et al.'s model (left) and ours (right) within two representative audiograms used for testing. The colors of the boxes on the left represent the different classes of objects. For our model, blue boxes represent label detections on the hearing loss axis or left ear symbols, while red boxes represent label detections on the frequency axis or right ear symbols.

offer marginally better detection, but also failed to detect multiple axis labels. Clearly, the overall multi-stage model's performance is impaired by the low recall of its individual components.

Given the similarity between the methods, it is likely that MAIN could perform significantly better if it were trained on a larger and more diverse set of audiology reports.

## VII. CONCLUSION

In this work, we presented a novel audiogram digitization algorithm capable of extracting hearing thresholds from a

variety of handwritten and computer-generated audiology reports as a proof-of-concept. We have shown that even if the task is seemingly simple, the variability in font, resolution, handwriting, and conventions followed by hearing clinicians make the task quite challenging. We believe that the work that is most needed to improve the performance of the algorithm involves refining the grid extraction step to improve its robustness. Collecting additional annotations, especially for axis labels, to retrain the axis label detector could also significantly improve the performance of the algorithm. At this time, the algorithm lacks the accuracy to be deployed in an unsupervised fashion, but combined with human supervision and corrections, it can still drastically speed up the digitization process by requiring only some manual adjustments to the extracted thresholds. The NIHL portal bootstraps the audiogram digitization algorithm to produce an initial annotation that can rapidly be adjusted by the annotator. This allows for the expansion of the dataset to further train the audiogram, axis labels, and symbol detection deep neural networks to enhance their accuracy.

Taken together, our work is novel as our model is the first of its kind to be trained on a large dataset of faxed or scanned audiology reports and capable of extracting hearing thresholds for all 8 commonly encountered audiological measurements. In addition, we developed software that allows for the rapid collection of annotations that can be used to further improve the recall and precision of our individual component detectors.

We anticipate that this work will be of great interest not only to insurance companies, who must process scanned or faxed audiological reports, but will also be useful to researchers in the field of audiology interested in using archived audiological records in paper format and to hospitals and clinics in migrating records in their archives to a digital format.

## ACKNOWLEDGMENT

The authors would like to thank Sherine Stephens, Valerie Hilton, Eric Gordon, and Ian Cross from the WSIB Innovation Laboratories for providing the data that was used in this work and also would like to thank Jaser El-Habrouk, Abhinav Yalamanchili, Ahmed Abdelrazik, Jason Fernandes, Tanzin Norjin, Syed Ahmed, Kenneth Calangan, Brandon Ca, and Vivian Han who each contributed numerous hours to annotate hundreds of audiological reports.

## REFERENCES

[1] K. Feder, D. Michaud, J. McNamee, E. Fitzpatrick, H. Davies, and T. Leroux, "Prevalence of hazardous occupational noise exposure, hearing loss, and hearing protection usage among a representative sample of working Canadians," *J. Occupational Environ. Med.*, vol. 59, no. 1, pp. 92–113, Jan. 2017.

[2] P. M. Rabinowitz, "Noise-induced hearing loss," *Amer. Family Physician*, vol. 61, no. 9, pp. 2749–2756, May 2000.

[3] R. H. Margolis and D. E. Morgan, "Automated pure-tone audiometry: An analysis of capacity, need, and benefit," *Amer. J. Audiol.*, vol. 17, no. 2, pp. 109–113, Dec. 2008.

[4] D. W. Swanepoel, F. Maclennan-Smith, and J. W. Hall, "Diagnostic pure-tone audiometry in schools: Mobile testing without a sound-treated environment," *J. Amer. Acad. Audiol.*, vol. 24, no. 10, pp. 992–1000, Nov. 2013.

[5] R. H. Margolis, R. Frisina, and J. P. Walton, "AMTAS: Automated method for testing auditory sensitivity: II. Air conduction audiograms in children and adults," *Int. J. Audiol.*, vol. 50, no. 7, pp. 434–439, Jul. 2011.

[6] G. P. Thompson, D. P. Sladen, B. J. H. Borst, and O. L. Still, "Accuracy of a tablet audiometer for measuring behavioral hearing thresholds in a clinical population," *Otolaryngol.-Head Neck Surgery*, vol. 153, no. 5, pp. 838–842, Nov. 2015.

[7] R. H. Margolis and G. L. Saly, "Toward a standard description of hearing loss," *Int. J. Audiol.*, vol. 46, no. 12, pp. 746–758, Jan. 2007.

[8] N. Bisgaard, M. S. M. G. Vlaming, and M. Dahlquist, "Standard audiograms for the IEC 60118-15 measurement procedure," *Trends Amplification*, vol. 14, no. 2, pp. 113–120, Jun. 2010.

[9] C.-Y. Lee, J.-H. Hwang, S.-J. Hou, and T.-C. Liu, "Using cluster analysis to classify audiogram shapes," *Int. J. Audiol.*, vol. 49, no. 9, pp. 628–633, Sep. 2010.

[10] F. Charih, M. Bromwich, A. E. Mark, R. Lefrançois, and J. R. Green, "Data-driven audiogram classification for mobile audiometry," *Sci. Rep.*, vol. 10, no. 1, p. 3962, Dec. 2020.

[11] M. G. Crowson, J. W. Lee, A. Hamour, R. Mahmood, A. Babier, V. Lin, D. L. Tucci, and T. C. Y. Chan, "AutoAudio: Deep learning for automatic audiogram interpretation," *J. Med. Syst.*, vol. 44, no. 9, p. 163, Sep. 2020.

[12] S. Li, C. Lu, L. Li, J. Duan, X. Fu, and H. Zhou, "Interpreting audiograms with multi-stage neural networks," Dec. 2021, *arXiv:2112.09357*.

[13] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, Y. Kwon, K. Michael, J. Fang, C. Wong, Z. Yifu, A. V, D. Montes, Z. Wang, C. Fati, J. Nadar, P. Skalski, A. Hogan, M. Strobel, M. Jain, and L. Mammana. (Aug. 2022). *Ultralytics/YOLOv5: V6.2—YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.AI Integrations*. [Online]. Available: https://doi.org/10.5281/zenodo.7002879

[14] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context," 2014, *arXiv:1405.0312*.

[15] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Los Angeles, CA, USA, Jun. 2019, pp. 658–666.

[16] R. O. Duda and R. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[17] Google. (Nov. 2020). *Tesseract-OCR/Tesseract*. [Online]. Available: https://github.com/tesseract-ocr/tesseract

**FRANÇOIS CHARIH** received the B.Sc. degree in biochemistry and the B.A.Sc. degree in chemical engineering from the University of Ottawa, Ottawa, ON, Canada, in 2016, and the M.A.Sc. degree in electrical and computer engineering from Carleton University, in 2019, where he is currently pursuing the Ph.D. degree and researches the applications of artificial intelligence in peptide drug design.

**JAMES R. GREEN** (Senior Member, IEEE) received the B.A.Sc. degree in systems design engineering from the University of Waterloo, Waterloo, ON, Canada, in 1998, and the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from Queen's University, Kingston, ON, Canada, in 2000 and 2005, respectively. He is currently a Professor with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada. His current research interests include machine learning challenges within biomedical informatics, patient monitoring, image analysis, computational acceleration of scientific computing, and design of novel assistive devices.