INTERPRETING AUDIOGRAMS WITH MULTI-STAGE NEURAL NETWORKS

A PREPRINT

Shufan Li
Orka Labs Inc.
lishufan@hiorka.com

Congxi Lu
Orka Labs Inc.
chauncey@hiorka.com

Linkai Li
Orka Labs Inc.
linkai@hiorka.com

Jirong Duan
ENT&Audiology Center
Xinhua Hospital∗
&Punan Hospital
jirong_duan@163.com

Xinping Fu†
Ear Institute
University College London
xinping_fu@163.com

Haoshuai Zhou‡
Orka Labs Inc.
haoshuai@hiorka.com

December 20, 2021

ABSTRACT

Audiograms are a particular type of line charts representing individuals' hearing level at various frequencies. They are used by audiologists to diagnose hearing loss, and further select and tune appropriate hearing aids for customers. There have been several projects such as Autoaudio that aim to accelerate this process through means of machine learning. But all existing models at their best can only detect audiograms in images and classify them into general categories. They are unable to extract hearing level information from detected audiograms by interpreting the marks, axis, and lines. To address this issue, we propose a Multi-stage Audiogram Interpretation Network (MAIN) that directly reads hearing level data from photos of audiograms. We also established Open Audiogram, an open dataset of audiogram images with annotations of marks and axes on which we trained and evaluated our proposed model. Experiments show that our model is feasible and reliable.

Keywords Audiogram · Audiology · Chart Recognition · Computer Vision · Machine Learning · Image Rectification · Object Detection · Faster-RCNN · Mask-RCNN

1 Introduction

Hearing loss is becoming an increasingly widespread health problem [1][2][3]. WHO estimates that around 2.5 billion people, or 1/4 of the global population will experience hearing loss by 2050 [4]. Fortunately, hearing aids can compensate for hearing deficits and considerably improve individuals’ quality of life [5][6][7]. Before hearing aids can be delivered to a customer, appropriate adjustments to the device (known as the fitting process) must be performed according to the audiogram of the customer.

In response to the growing demand, the U.S Senate passed the Over-the-Counter Hearing Aid Act of 2017 which created a class of hearing aids directly available to consumers without the involvements of licensed professionals [8]. To make these products viable choices for consumers, automated fitting processes are of necessity. While some of these products can measure hearing loss and adjust the device accordingly (self-fitting), it is more desirable to use results from specialized hearing tests when a user is already in possession of an audiogram.

∗Xinhua Hospital is affiliated to School of Medicine, Shanghai Jiao Tong University
†Xinping Fu is studying at Ear Institute, University College London
‡Corresponding Author
Outside the U.S. many parts of the world are experiencing a shortage of audiologists which limited access to well-tuned hearing aids [9]. Moreover, even for professional audiologists, the task of interpreting audiograms can be laborious and prone to human error. Hence, an effective and reliable automatic audiogram interrupter can be a game-changer as it allows a limited supply of audiologists to serve a larger population and improves the experiences of OTC hearing aids’ customers.

While there have been attempts to directly classify the types of hearing loss (if any) from audiogram images [10, 11, 12], they are only able to provide class-based qualitative descriptions of audiograms. They cannot recover the full information, specifically the precise hearing level at different frequencies, which are crucial for tuning hearing aids. Hence, these models are not particularly helpful in the process of fine-tuning a hearing aid, and alternative methods which can extract detailed hearing level information would be more desirable.

To address this gap between growing demands and the limited capacity of existing models, in this paper we tackle the task of audiogram interpretation, which we define as recovering the \((frequency, hearing\_level)\) tuple from an arbitrary audiogram photo. Compared with audiogram classification, audiogram interpretation requires extracting the complete information necessary to reconstruct a digital copy of the original audiogram, and is therefore more challenging. We introduce a new dataset Open Audiogram consisting of 420 photos of 34 audiograms taken in real-world scenarios with varying lighting, angle, and backgrounds. In addition, we propose a baseline Multi-stage Audiogram Interpretation Network. In the first stage, we detect audiograms in an arbitrary image and crop each audiogram by its bounding box to remove potential interference from the background. In the second stage, we attempt to rectify the distortion caused by the camera angle by applying a perspective transform. In the third stage, we detect marks and axis labels from cropped, rectified images. We use axis labels to reconstruct a coordinate system, and then project mark coordinates onto the fitted axes to recover \((frequency, hearing\_level)\) tuples. RANSAC algorithm is adopted in this process to remove outliers caused by misclassification. An overview of this pipeline is shown in Figure 2. Experiments show that our proposed baseline is feasible with a test accuracy of 84%. Further, 94% of the detected tuples are within a margin of ±5 dB HL in hearing level, which is the smallest possible error in hearing level since all hearing level entries are usually multiples of 5 (Table 2).

2 Background

At a high level, audiograms are visual representations of an individual’s hearing level. The task of audiogram interpretation is the task of retrieving quantitative hearing level information from images of audiograms. In this section, we give formal definitions of this task and the relevant concepts. Additionally, we summarize prior related works.

2.1 Audiogram Interpretation

An audiogram is a line chart representing a person’s hearing level. As is shown in Figure 1, a typical audiogram consists of the following visual elements: X-axis representing hearing levels in dB HL ranging from -10 dB HL to 120 dB HL, and marks whose coordinates indicate an individual’s hearing level at different frequencies. The X-axis has log-scale tick marks at 125 Hz, 250 Hz, ..., 8000 Hz, 16000 Hz. The Y-axis has linear, uniform tick marks with a step size of 10 dB HL.

Compared with a general line chart whose marks have an infinite, continuous sample space contained in \(\mathbb{R}^2\), audiograms are easier to interpret because we can utilize this characteristic and map the fitted real-valued coordinates to its nearest neighbor in the sample space. Another important characteristic we make use of is the fact that X tick marks and Y tick marks have no overlapping values. So if we can recognize the number of a tick mark, we can automatically determine which axis it belongs to. This reduces the difficulty of inferring the direction of axes.

The digital representation of an audiogram is the set \(G = \{(x_i, y_i)\}_{i=1}^n\) containing the coordinates of all the marks observed in an audiogram, where \(x_i \in \{125, 250, ..., 8000, 16000\}\) are frequencies in Hz and \(y_i \in \{-10, 0, ..., 110, 120\}\) are hearing levels in dB HL. Let \(\Omega\) be the set of all audiogram images and \(\Gamma\) be the set of all possible digital representations of audiograms. There exists a natural mapping \(F : \Omega \rightarrow \Gamma\) which maps every audiogram image to its actual digital representation. We define the task of audiogram interpretation as the task of finding a computable, accurate approximation mapping \(f : \Omega \rightarrow \Gamma\) of \(F\) such that for an image \(A \in \Omega\) we can use \(f(A)\) to find an approximation of \(F(A)\). This paper is the first to study the automatic interpretation of audiograms.

2.2 Related Works

The related works of audiogram interpretation mainly fall into two categories: (1) audiogram classification and (2) chart data extraction.
Interpreting Audiograms with Multi-stage Neural Networks

2.2.1 Audiogram classification

In contrast to audiogram interpretation, the task of audiogram classification only attempts to retrieve a qualitative summary of audiograms based on classes. There have been a variety of works in this field, most of which rely on expert-tuned rules [13, 14, 15]. In recent years, some methods utilizing machine learning techniques to distinguish different audiograms have been proposed. Chariri et al. [10] make use of decision forest to classify audiograms based on configuration, symmetry, and severity. Lee et al. [11] leverage K-means clustering to categorize audiogram shapes into six basic types and five subtypes. Crowson et al. [12] employ ResNet-101 to differentiate audiograms of individuals with conductive, sensorineural, mixed, or no hearing loss. Nonetheless, all these methods only give a rough summary of certain properties of audiograms. They cannot provide enough details for a hearing aid prescription.

2.2.2 Chart Data Extraction

Chart Data Extraction refers to the general task of retrieving quantitative information for chart images. Existing Chart Data Extraction algorithms fall into two categories: automatic and semiautomatic. Semiautomatic algorithms typically involve extensive, time-consuming user interactions. For example, WebPlot-Digitizer [16], a famous semiautomatic digitizing tool, requires users to manually label and calibrate a lot of data points. ChartSense [17], another interactive data extraction system, also requires the user to manually identify chart areas and intervals of data points. In contrast, automatic algorithms require no manual work. One common approach is to combine image processing algorithms with OCR. Histogram-based techniques and spatial filtering are generally used to locate coordinate axes and data points, while OCR is employed to detect textblocks. Some recent works replace these traditional image processing techniques with deep learning models. Scatteract [18], for example, use ReInspect to detect data points, tick marks and axis labels. However, these methods are only tested on high-quality screenshots where there is little noise, occlusion, or rotation in the image, which are drastically different from photos in our dataset. Additionally, they did not make use of the aforementioned limited sample space of mark coordinates and tick marks. Hence they are at best sub-optimal solutions to our problem.

Our work lies at the intersection of Audiogram Classification and Chart Data Extraction. To the best of our knowledge, we are the first to propose a feasible model which automatically extracts the complete audiogram information from real-world photos under diverse lighting, occlusion, and rotation.

3 The Open Audiogram Dataset

In this paper, we present a challenging dataset for audiogram recognition. Most of the existing chart recognition datasets such as AutoChart consist of synthetic chart images with almost no variation in lighting conditions and perspective angles (Figure 3). These images also do not follow the aforementioned characteristics of audiograms. Hence, they are unfit for audiogram recognition. To address this apparent gap, we created Open Audiogram, a dataset consisting of 420 camera photos of 34 audiograms and additional 30 scanned images of different audiograms.
Interpreting Audiograms with Multi-stage Neural Networks

3.1 Image Preparation

We first collected raw audiogram images generated by Noah 4 System \cite{19} from our audiologists’ patient records and filtered out all sensitive data. Then these audiograms were printed on standard A4-sized paper. We took a total of 420 images of these printed audiograms under various angles between 0 and 45 degrees (measured by the angle of the camera direction and the normal vector of the paper). We also intentionally took the photos at different locations in an office building to adjust the balance between artificial indoor lighting and natural skylight coming from the window. In some photos, there are cast shadows of other objects on the paper with the edge of the shadow crossing the audiogram region. In some photos, obstructions such as pens are purposely placed on the audiogram region. In some photos, the paper is bent or folded (Figure 4). During this process, all sensitive information such as human faces and computer screens are masked out for privacy concerns. These intentional occlusions and distortions make the dataset particularly challenging. To evaluate the impact of these distortions, we generated an additional set of 30 audiograms, printed them, and scanned them using a laser scanner. These undistorted images look nearly identical to the raw image (Figure 5).

3.2 Annotation

There are four levels of annotations.

- Level 1: With the raw \((frequency, hearing_level)\) pairs used to generate the audiogram
- Level 2: With the bounding boxes of the gram region
- Level 3: With the bounding polygons (mostly quadrilateral with few exceptions) of the chart region
- Level 4: With the bounding boxes of all marks and axis present in the gram region

The level 1 annotations alone provide sufficient information to evaluate the performance of any audiogram interpretation algorithm because it allows the comparison between the predicted \((frequency, hearing_level)\) pairs and ground truths. Due to the multistage nature of our proposed baseline, it is not very helpful to annotate all images at all levels. For example, the marks and axis detection is performed on cropped images, and hence providing level 4 annotation...
Interpreting Audiograms with Multi-stage Neural Networks

Figure 4: Examples of photos in our Open Audiogram dataset taken under varying lighting conditions, occlusions and camera angles.

Figure 5: An Example of our scanned audiograms

on raw photos is meaningless. Therefore, we have varying numbers of images at each annotation level, which are summarized in Table 1.

To our best knowledge, our dataset is the first audiogram dataset focused on data extraction in real-world scenarios. It can serve as a benchmark for evaluating different existing approaches and facilitate future works on audiogram interpretation.

|       | Photo | Scanned |
|-------|-------|---------|
| Total | 410   | 420     | 30      |
| level1| 0     | 420     | 30      |
| level2| 0     | 420     | 0       |
| level3| 410   | 0       | 0       |
| level4| 223   | 0       | 0       |
Figure 6: An example of perspective distortion, in which the naive projection deviates from the ground truth because parallel lines are no longer parallel.

Figure 7: A comparison of line extraction methods. Our adaptive threshold method is able to better preserve gridline information.

4 Model Architecture

Our proposed Multi-stage Audiogram Interpretation Network (MAIN) consists of three stages: gram detection, perspective rectification, and axis and label detection (Figure 2). In the first stage, we use a Faster-RCNN to detect the gram region containing the axis and marks. In the second stage, we leverage a Mask-RCNN in combination with traditional image processing techniques such as hough line transform to get an estimate of vanishing points and apply a perspective transform to rectify the image. In the last stage, we employ another Faster-RCNN to detect axis labels and marks. We use RANSAC to fit two axes from the label coordinates, and then project mark coordinates onto the fitted axes to obtain their frequencies and hearing levels.

4.1 Gram Detection

The task of the gram detection network is to identify the bounding box of the gram areas containing axis labels and marks. We used a Faster-RCNN with ResNet and FPN backbone for this task. Existing benchmarks show that such architecture has a satisfying performance on the Common Objects in Context (COCO) Dataset and is therefore suitable for this task [20].

4.2 Perspective Rectification

Since the final \((frequency, hearing\_level)\) predictions are generated by projecting detected data points on the detected axes, it is crucial that the image does not contain a vanishing point. Parallel lines in the audiogram must stay parallel in the image. For example, in Figure 6, the projected coordinate of the mark deviates from the ground truth by a huge margin because of the perspective distortion. We propose two methods to rectify perspective distortions caused by camera angles. The first primarily relies on line detection to get an estimation of vanishing points, the second utilizes Mask-RCNN to predict the four corners of the gram region.

4.2.1 Line-Detection-Based Approach

Chaudhury et al. proposed a method to estimate vanishing points by applying RANSAC on extracted edgelets. They extract edgelets from images by applying a “Haris Corner Detector style” filter on the image and compute the eigenvalues
Figure 8: An illustration of perspective rectification method in 4.2.1. It has four stages. (1) Binarize the image by adaptive threshold. (2) Extract long lines using hough line transform. (3) Fit two vanishing points using RANSAC. (4) Apply an inverse perspective transform.

and eigenvectors of the Gaussian weighted covariance matrix in a square neighborhood [21]. A later work modified this approach by using canny edge detection in combination with probabilistic hough line transform to detect edgelets instead [22]. We replace the canny edge detector by running a gaussian blur operation on the image and then apply an adaptive threshold. The output is then passed to the probabilistic hough line transform to extract line segments. We made this choice because we intend to apply RANSAC on gridlines in the gram region which is structurally different from edges between different objects (Figure 7).

The output of probabilistic hough line transform is organized into a set of line segments \( \{ l_i \} \). We define the model \( M(l_1, l_2) \) be the representation of a vanishing point at the intersection of \( l_1, l_2 \). Given a model \( M \), the vote of a particular line segment \( l_i \) for \( M \) is

\[
\text{vote}(l_i, M) = \begin{cases} 
\|l_i\|_2 & \theta < 5^\circ \\
0 & \text{otherwise}
\end{cases}
\]

where \( \theta \) is the intersection angle between \( l_i \) and the line passing through the vanishing point represented by \( M \) and the center of the line segment \( l_i \). We apply a standard RANSAC using this metric by randomly generating model proposals and picking the one with the highest total votes. After one vanishing point \( M_1 \) is identified, all inliners \( \{ l_i | \text{vote}(l_i, M_1) > 0 \} \) are removed and we reapply RANSAC on the remaining line segments to get the second vanishing point. With two vanishing points, we compute an inverse perspective transform and finally rectify the image (Figure 8).

4.2.2 Mask-RCNN-Based Approach

Since the previous approach heavily relies on line grids, it is susceptible to interference from other patterns and shapes in the gram region. In some variations of audiograms (not included in our dataset), there are many congested shapes in the gram region which affects the reliability of hough line transform (Figure 9). To address this issue, we proposed an alternative border-based approach utilizing Mask-RCNN.

Mask-RCNN is a variant of RCNN modified for image segmentation tasks [23]. In addition to bounding boxes, Mask-RCNN is able to generate bounding polygons of detected objects from predicted masks. We define the chart region as the box containing line grids. It is different from the gram region in that it does not contain axis labels. Hence, it has separate annotations from gram regions in our dataset. We use a Mask-RCNN to extract the bounding polygon of the chart region, and approximate the bounding polygon with a quadrilateral. While similar polygon simplification tasks are often achieved by the Douglas-Peucker algorithm [24], it is not directly applicable as the mask polygons predicted by our network tend to have "rounded edges." Since the four corners are not vertices of the mask polygon, Douglas-Peucker will yield sub-optimal results as its output is a subset of the original polygon’s vertices. We propose an alternative greedy algorithm to address this issue. We intend to find four straight lines and make their intersections four corners of the quadrilateral. We first apply Douglas-Peucker with a very small threshold (0.005 \( \times \) perimeter) such that the long edges will be smoothed but the "rounded corners" will not be reduced. To find the first line, we find the longest edge in the smoothed polygon and extend it to a line. We remove all edges in close proximity to this line and repeat the same process to find the second, third and fourth line. With four lines we get six intersections (they can be at infinity when lines are strictly parallel). We remove the furthest 2 points and use the remaining four to construct the approximate quadrilateral of the original polygon. This method can find corners that are not a vertex of the original polygon and hence it is more desirable than applying Douglas-Peucker naively (Figure 10).

We use the four points of the fitted quadrilateral as estimations of four corners of the chart region. From these four points, we construct an inverse perspective transform that would rectify these four points to four vertices of a square.
4.3 Axis and Mark Detection

With the image rectified, we can proceed to detect marks and axis. This task is no more than a typical multi-class object detection task. We apply the same Faster-RCNN as in 4.1 with the only difference being that we have 24 classes (two types of marks for left ear and right ear, 14 hearing level ticks, and 8 frequency ticks) instead of one.

After we extract the bounding boxes of axis labels and marks, we calculate the centroid of each bounding box and consider them to be the coordinates of each mark/axis label. Since the two axes have no overlapping ticks, we can separate the axis labels into two groups purely based on their numerical value. We apply RANSAC linear regression for each group and fit two lines representing the direction of two axes. RANSAC is introduced to improve the resilience against some misclassified instances.

For each mark and axis, we calculate their projections onto the fitted axes and acquire \((p_{\text{frequency}}, p_{\text{hl}})\), where \(p_{\text{hl}}\) is the distance between the origin and the projected points on the hearing level axis and \(p_{\text{frequency}}\) is the distance between the origin and the projected points on the frequency axis (Figure 11).

We want to construct a a mapping \(g : \mathbb{R}^2 \rightarrow \Gamma\) that maps a \((p_{\text{frequency}}, p_{\text{hl}})\) tuple to an element in \(\Gamma\) (the finite set of all possible mark coordinates as described in 2.1). We fist construct two mappings \(g_f, g_l : \mathbb{R} \rightarrow \mathbb{R}\) such that \(g_f\) maps \(p_{\text{frequency}}\) to a real valued frequency and \(g_l\) maps \(p_{\text{hl}}\) to a real valued hearing level. These two mappings are individually fitted by RANSAC linear regression on the \(p_{\text{hl}}, p_{\text{frequency}}\) values of respective axis labels and their numerical values in dB HL and Hz. \(g_l\) is directly fitted since the hearing level ticks are linear. For \(g_f\), we take the log of frequency value before passing it to a RANSAC regressor as the frequency ticks are in log scale.
Interpreting Audiograms with Multi-stage Neural Networks

Figure 11: Projection of marks. $p_{\text{frequency}}$ and $p_{\text{hl}}$ are the magnitude of projected vectors on respective axis.

Figure 12: An example of MAIN's final outputs. The lines are two fitted axis. The rectangles are bounding boxes of axis labels and marks.

With these two mappings, we construct $g : \mathbb{R}^2 \rightarrow \Gamma$ such that $g(p_{\text{frequency}}, p_{\text{hl}})$ equals to the closet neighbor of $(g_f(p_{\text{frequency}}), g_l(p_{\text{hl}}))$ in $\Gamma$. For example, if $(g_f(p_{\text{frequency}}), g_l(p_{\text{hl}}))$ is $(130, 14)$, it will be mapped to $(125, 15)$. An example of model output is shown in Figure 12.

5 Experiments

5.1 Model Implementation and Data Preparation

We implemented Faster-RCNN and Mask-RCNN using PyTorch and detectron2 library [20]. We implemented our line-detection-based perspective rectification algorithms by modifying Chilamkurthy’s work [22]. We implemented our Mask-RCNN based algorithms from scratch utilizing various publicly available libraries such as OpenCV. All RANSAC processes in the paper are implemented by the scikit-learn library.

We randomly split the 420 raw images into a training set of 352 images and a test set of 68 images. All images have gram (level 2) annotations, and therefore these two sets are also the training and test set for the gram detector. We have 410 cropped images with annotations of bounding polygons of chart regions (level 3) and they are split into a training set of 369 images and a test set of 41 images. We have 223 cropped images with annotations of axis and marks (level 4) and they are split into a training set of 203 images and a test set of 20 images. It should be noted that while the split is not identical, no cropped images in any of the training set comes from raw images in the test set, which is used in our final evaluation.
Interpreting Audiograms with Multi-stage Neural Networks

Table 2: Test Set Performance

|                   | No Rectification | Line Detection | Mask RCNN | No Rectification |
|-------------------|------------------|----------------|-----------|------------------|
| Label Total Accuracy | 0.765, 0.727*    | 0.853, 0.834   | 0.843, 0.842 | 0.987, 0.991     |
| Frequency Accuracy  | 0.967, 0.918     | 0.977, 0.955   | 0.963, 0.962 | 0.987, 0.991     |
| HL Accuracy        | 0.809, 0.768     | 0.878, 0.858   | 0.868, 0.867 | 0.987, 0.991     |
| Label Total Accuracy ±5** | 0.937, 0.890    | 0.960, 0.938   | 0.950, 0.949 | 0.987.0.991      |

* The numbers are recall, precision respectively, where recall is defined as (number of correct predictions)/(number of ground truth marks) and precision is defined as (number of correct predictions)/(number of predicted marks)

** This metric assumes all predictions that are off by at most 5dB HL in hearing level as correct.

Table 3: Test Set Performance of Different Stages

|                        | AP    | AP50  | AP75  |
|------------------------|-------|-------|-------|
| Gram (Faster-RCNN)     | 0.907 | 0.987 | 0.987 |
| Axis and Marks (Faster-RCNN) | 0.621 | 0.850 | 0.822 |
| Chart Region (Mask-RCNN) | 0.965 | 0.965 | 0.965 |

5.2 Training

All the models are trained on one Nvidia GPU. The Faster-RCNNs were trained for 100 epochs and the Mask-RCNN was trained for 20 epochs. The learning rate of Faster-RCNN used for gram detection is $10^{-5}$ while the learning rate of Faster-RCNN used for axis and label detection is $10^{-4}$. The learning rate of Mask-RCNN is $10^{-4}$. We used the faster_rcnn_X_101_32x8d_FPN_3x pre-trained weights from detectron2 library to initialize our Faster-RCNN and mask_rcnn_R_50_FPN_3x to initialize Mask-RCNN [20]. Results show that all the networks achieved a satisfying performance on the test set on their respective tasks (Table 3).

5.3 Evaluation

We evaluate our multi-stage model on the test set of 68 camera photos. We compared the performance of two proposed perspective rectification methods. Results show that both methods are able to improve the model performance on camera photos. We also test our model on a set of 30 scanned images for comparison. Since there is no perspective distortion, we did not apply perspective rectification. The results show that our model can handle scanned images without ever being exposed to such images during the training. In fact, it performances significantly better on scanned images (Table 2). These results show that our proposed baseline is feasible. To the best of our knowledge, there are no comparable prior works. Hence we did not compare our baseline with other works.

6 Conclusion

There is a growing demand to automatize the process of audiogram interpretation, and our work shows that it is feasible to reconstruct audiogram information from camera photos and scanned images by leveraging state-of-art neural networks. Our baseline has an accuracy of 98% on scanned images which suggests that it can be an effective and robust replacement of manual interpretation for audiologists with access to a scanner. An 84% accuracy on camera photos suggests that such architecture can also be applied in scenarios where users need to take photos of their audiograms using a smartphone, which can be useful in a pandemic lockdown, the process of autonomous fitting of OTC hearing-aids, or in areas with limited access to audiologists.

Experimental results suggest that one major challenge in audiogram interpretation is the distortions caused by varying lighting conditions and camera angles. They also showed that our proposed rectification algorithms can mitigate the impact of perspective distortion.

To the best of our knowledge, we are the first to establish a benchmark dataset and propose a baseline model for audiogram interpretation. In future works, we will try to adjust our model so that it can be applied in circumstances where the hearing levels of the left and right ears are represented in a single audiogram by two lines with different
marks. We may also try to differentiate audiograms of air conduction and bone conduction audiometry which differ in their notations.

References

[1] World Health Organization. *Addressing the rising prevalence of hearing loss*. World Health Organization, 2018.

[2] GBD 2016 Lifetime Risk of Stroke Collaborators. Global, regional, and country-specific lifetime risks of stroke, 1990 and 2016. *New England Journal of Medicine*, 379(25):2429–2437, 2018.

[3] Aislyn Orji, Kaloyan Kamenov, Mae Dirac, Adrian Davis, Shelly Chadha, and Theo Vos. Global and regional needs, unmet needs and access to hearing aids. *International Journal of Audiology*, 59(3):166–172, 2020.

[4] Shelly Chadha, Kaloyan Kamenov, and Alarcos Cieza. The world report on hearing, 2021. *Bulletin of the World Health Organization*, 99(4):242, 2021.

[5] Theresa Hnath Chisolm, Carole E Johnson, Jeffrey L Danhauer, Laural JP Portz, Harvey B Abrams, Sharon Lesner, Patricia A McCarthy, and Craig W Newman. A systematic review of health-related quality of life and hearing aids: final report of the american academy of audiology task force on the health-related quality of life benefits of amplification in adults. *Journal of the American Academy of Audiology*, 18(2):151–183, 2007.

[6] Bridget Shield. Evaluation of the social and economic costs of hearing impairment. *Hear-it AISBL*, pages 1–202, 2006.

[7] Harvey Dillon. *Hearing aids*. Hodder Arnold, 2008.

[8] FDA Reauthorization Act of 2017 SEC. 709. Regulation of Over-the-Counter Hearing Aids. U.S. Public Law 115-52 Sec 709.

[9] Kaloyan Kamenov, Ricardo Martinez, Teena Kunjumen, and Shelly Chadha. Ear and hearing care workforce: current status and its implications. *Ear and Hearing*, 42(2):249–257, 2021.

[10] François Chariri, M. Bromwich, A. E. Mark, Renée Lefrançois, and J. Green. Data-driven audiogram classification for mobile audiometry. *Scientific Reports*, 10, 2020.

[11] Cheng-Yung Lee, Juen-Haur Hwang, Szu-Jen Hou, and Tien-Chen Liu. Using cluster analysis to classify audiogram shapes. *International journal of audiology*, 49(9):628–633, 2010.

[12] M. G. Crowson, J. W. Lee, Amr Hamour, Rafid Mahmood, A. Babier, V. Lin, D. Tucci, and T. Chan. Autoaudio: Deep learning for automatic audiogram interpretation. *Journal of Medical Systems*, 44:1–7, 2020.

[13] Robert H Margolis and George L Saly. Audiogram classification system, December 13 2011. US Patent 8,075,494.

[14] Raymond Carhart. An improved method for classifying audiograms. *The Laryngoscope*, 55(11):640–662, 1945.

[15] Stacy R Guild. A method of classifying audiograms. *The Laryngoscope*, 42(11):821–836, 1932.

[16] Ankit Rohatgi. Webplotdigitizer: Version 4.4, 2020.

[17] Daekyoung Jung, Wonjae Kim, Hyunjoo Song, Jeongin Hwang, B. Lee, B. H. Kim, and Jinwook Seo. Chartsense: Interactive data extraction from chart images. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017.

[18] M. Cliche, David S. Rosenberg, Dhruv Madeka, and Connie Yee. Scatteract: Automated extraction of data from scatter plots. In *ECML/PKDD*, 2017.

[19] HIMSA. Noah system 4. [https://www.himsa.com/products/all-about-noah-system-4/](https://www.himsa.com/products/all-about-noah-system-4/)

[20] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. [https://github.com/facebookresearch/detectron2](https://github.com/facebookresearch/detectron2), 2019.

[21] Krishnendu Chaudhury, Stephen DiVerdi, and Sergey Ioffe. Auto-rectification of user photos. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 3479–3483. IEEE, 2014.

[22] Sasank Chilamkurthy. Image-rectification. [https://github.com/chsasank/Image-Rectification](https://github.com/chsasank/Image-Rectification), 2020.
[23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.

[24] Mahes Visvalingam and J Duncan Whyatt. The douglas-peucker algorithm for line simplification: re-evaluation through visualization. In Computer Graphics Forum, volume 9(3), pages 213–225. Wiley Online Library, 1990.