Applying Machine Learning to Otitis Media Diagnosis & Image Capture

Remmie Health

MIE429 Final Report

December 5th, 2022

Zahra Nadine Kandola (1004913873)

Markus Kunej (1005351897)

Jay Mohile (1005220547)

Aidan Lawford-Wickham (1005089226)

Tian Lan (1006544615)

I. BACKGROUND

Ear infections, clinically known as Otitis Media (OM), are globally one of the most common childhood ailments [1], [2]. OM often presents an array of painful symptoms, including fever, ear pain, hearing loss, and increased chances of reinfection [1], [2]. Early diagnosis and appropriate management can reduce symptoms and their progression [2]. Unfortunately, socioeconomic factors, geographical location, and practitioner availability limit access to treatment for this common illness. Since OM has well-defined symptoms and requires a visual diagnosis, this lack of access can be addressed through telehealth services. Remmie, our corporate partner, offers a platform to facilitate virtual care for OM.

II. PROBLEM

One product Remmie currently offers is their home otoscope [3]. This device is a U.S. Food and Drug Administration (FDA) registered otoscope fitted with a 1080P Wireless WiFi ear camera [3]. With this product, users can capture an image of their potentially infected ear, nose, or throat, and receive 24/7 evaluation from a doctor through the telemedicine service offered through their app [3].

Currently, many photos taken by users are not immediately suitable for diagnosis. This requires multiple rounds of communication to resolve, and threatens Remmie's ultimate goal of seamless, highly-automated diagnosis through their platform.

Our goal was to aid Remmie in addressing these obstacles by building two models. Our solution can be divided into three objectives:

1. Process, label, and clean the images of eardrums. **("Data Processing")**
2. Develop a machine learning model to identify whether a taken image is suitable for diagnosis, by locating certain visual landmarks (prescribed by Remmie). **("Object Detection Model")**
3. Develop a binary machine learning classifier that can evaluate the probability of a person having OM based on an image of their eardrum. **("Classification Model")**

Initially, our primary objective was to develop the Classification Model. However, after significant experimentation and realizing (alongside Remmie's medical advisors) that our dataset had little diagnostic power, we turned our focus to Object Detection. This still aligns with Remmie’s need to capture high-quality images of their patients’ eardrums for diagnosis (which is their major current obstacle). Thus, Object Detection (and its prerequisite, Data Processing) were our primary objectives. Here, our success would be evaluated based on precisely detecting key features of the middle ear. This pivot occurred very late into the project, after a literature review, and as such, only a proof of concept was targeted

The Classification Model was now considered a stretch goal, and while preferred, not necessary to project success. Model accuracy, precision, and recall were used to evaluate project success, emphasizing the latter two due to our client’s priorities and desire for future FDA approval. This requires a highly consistent classifier with a precision of 85% and a recall of 90%.

III. DATA

Datasets

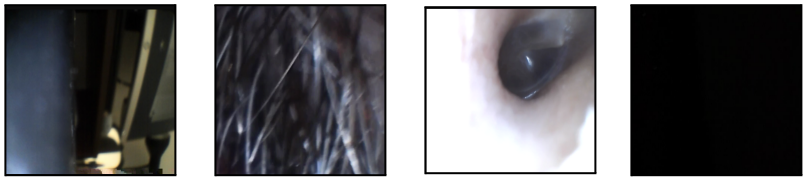


Figure 1: Sample images from users captured using Remmie’s otoscope.

Our largest dataset consisted of 7GB of unlabelled and unclassified images captured by Remmie's clients (shown in *Figure 1*). Although the images contained some eardrums, the majority were of other objects. Additionally, these images were not consistently sized, and many were too blurry for identification. Through discussions with our client, and their consultants, we determined that this dataset would be unsuitable for our use cases as the noise of the dataset requires significant amounts of manual processing.

A second dataset, 3008 anonymized images collected via Remmie's partner institution, served as the basis for our models (**"Dataset A"**). Each data point consists of an image of the patient's eardrum and their corresponding diagnosis. A summary of **Dataset A's** features, and its binarized version used for classification, **Dataset B**, are presented in *Appendix A.a*. **Dataset B** has a 38.43%/61.57% split for OM/Not OM.

These images were taken by the treating doctor using standard otoscopes retrofitted with a camera under various conditions, such as during surgical procedures or diagnosis. Datasets used for similar tasks were not publicly available due to the sensitivity of medical data. Therefore, the dataset could not be supplemented with externally sourced images.

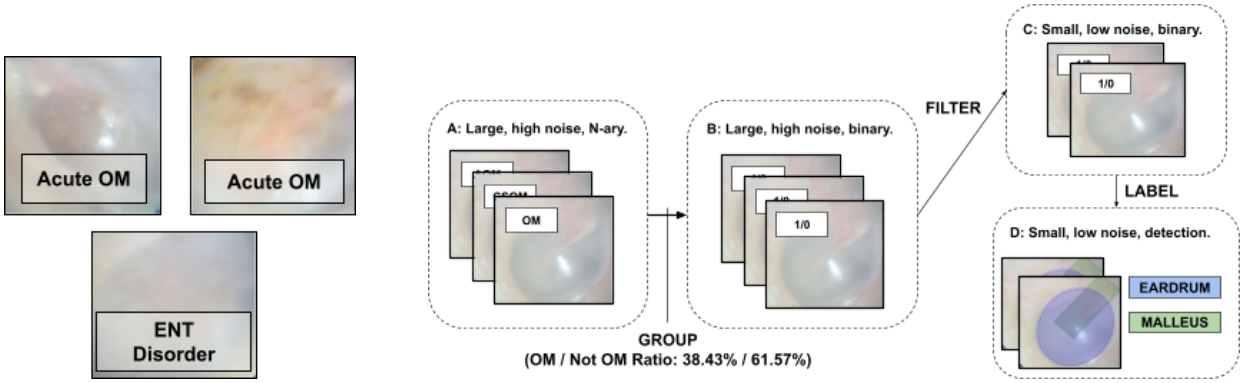


*Figure 2: **Left**: Sample images from Dataset A. **Right**: Datasets A, B, C, and D.*

Data Processing

Data Filtering

Due to the varying quality of images in our dataset, significant effort went into processing/filtering to produce an appropriate training set. Based on our discussions with Remmie's consulting Ear Nose, and Throat (ENT) physician, the following high-level criteria were established to deem an image relevant for diagnosis: (1) the malleus handle clearly visible, (2) the eardrum clearly visible, and (3) no obvious artifacts such as glare, blur, or occlusion.

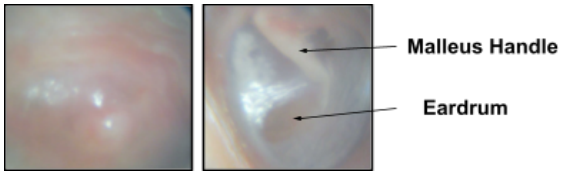


*Figure 3: Unsuitable (**Left**) vs suitable (**Right**) images.*

Given the size of the dataset and the specialized knowledge required, fully filtering this dataset was not feasible. We instead prepared a subset of obvious examples, intending to demonstrate a proof-of-concept improvement over the original one. This subset will subsequently be referred to as **Dataset C.** The features and characteristics of this dataset are summarized in *Appendix A.b*. This dataset has a 40%/60% split for OM/Not OM.

Labelling Infrastructure

Unlike classification, Object Detection required first generating additional labels for the dataset. This was done by marking the location and shape of the eardrum and malleus handle using bounding shapes. Given the complexity of these labels, time was invested into designing an efficient labelling workflow. Ultimately, the open-source Label Studio tool was selected. While an ideal system would run centrally on a Remmie server, infrastructure limitations required us to use several self-hosted instances. Using this infrastructure, bounding boxes were added to **Dataset C** to create **Dataset D.**

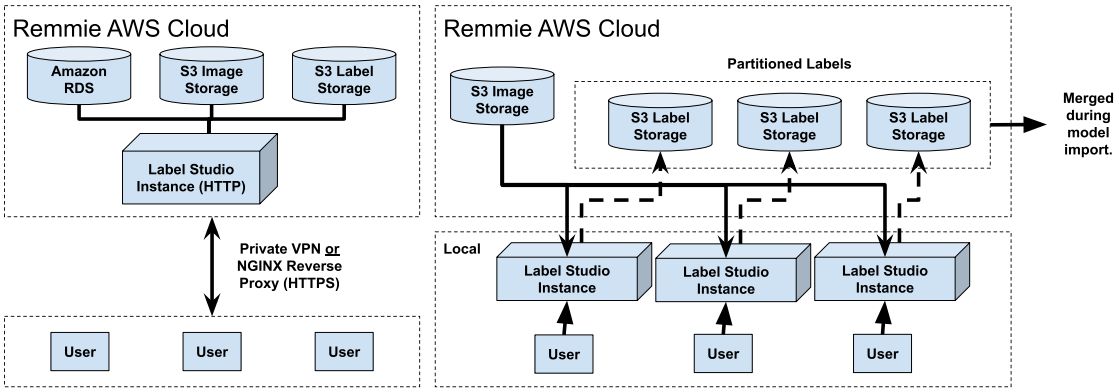


*Figure 4: Ideal architecture (**Left**) vs our workaround (**Right**).*

Online Image Processing

During training, we use standard pre-processing and augmentation techniques. Pre-processing is applied to all images at the network-input level, and includes contrast correction and channel-wise (RGB) colour normalization, as shown in *Figure 5*. Contrast correction is non-trivial, since naively boosting it could cause "clipping" of data near white/black. As such, a context-aware correction is applied, clamping/interpolating between the darkest and lightest points. Augmentation is further applied to just the training set, and is done online to ensure (a) greater data diversity, and (b) simplified train/test isolation. Both datasets undergo standard colour-based augmentation (e.g. chromatic distortions and blur).

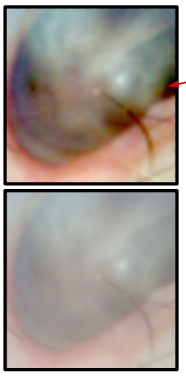


*Figure 5: Example of an image after (**Up**) colour correction and before (**Down**).*

Furthermore, the classification datasets (**Datasets B** and **C**) underwent a full suite of spatial distortions, including randomized crop, scale, and rotation. These datasets were also normalized via RGB mean subtraction and divided by the standard deviation values of the images, which were calculated to be [0.6243, 0.6003, 0.5717] and [0.2209, 0.2121, 0.2039], for each colour channel. These augmentations were chosen for **Dataset B** and **C** because they were previously used, with much success, by researchers in 2020 attempting a similar problem to ours, classifying OM from otoscopic images [4]. The Object Detection dataset, **Dataset D**, had a similar augmentation process. However, since **D** features spatial coordinates (bounding boxes), it is run through a much more limited spatial augmentation, only randomized X/Y/XY flip.

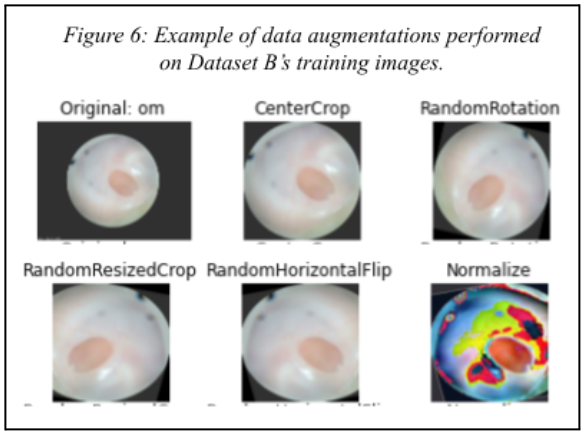


Figure 6: Example of data augmentations performed on Dataset B's training images.

IV. METHODS

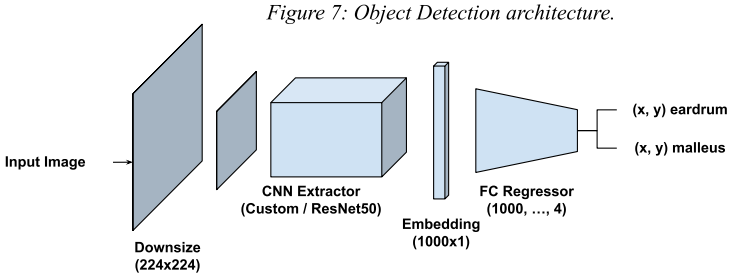
Object Detection Model

Guided by the above criteria, a model was built to identify features of **Dataset D**: the malleus handle and the eardrum. While many identification methods are possible, our first attempt involved detecting the spatial centroids of each feature with the possibility of adding more predictions (rotation, extent) later on.

Specifically, the model consumes a 224x224 image and outputs a 4x1 vector, interpreted as the X/Y coordinates of the malleus and eardrum respectively. These are outputted in size-normalized form (0, 1), to maintain invariance against the image size used. Building on previous similar work in image-based regression [5], our model uses a convolutional neural network (CNN) based feature extractor, followed by a fully connected regressor. Much of the model was configurable through hyper-parameters, allowing a high degree of experimentation.

One such parameter controlled the network's CNN backbone. Both custom and ResNet50 extractors were explored, owing to previous work showing that classification backbones can have predictive power in low-dimensional regression settings [5]. In both configurations, a 1000 feature output was used, with ReLU activation through the regressor. Another parameter involved the network's output activation. Given the prediction generally falls within a 0-1 normalized output range, inductive bias suggests that a sigmoidal activation may better represent the problem. However, counter arguments exist due to the possibility that (a) centroids occasionally fall outside the viewport and (b) the vanishing gradient problem. As such, empirical evidence was desired.

Figure 7: Object Detection architecture.



Finally, the network required standard tuning of hyperparameters such as learning rate (LR), batch size, and fully connected (FC) layers. Stochastic Gradient Descent (SGD) optimization was used, with momentum also subject to tuning, and schedule-based LR tied to the validation loss plateau. For all, a standard 60/30/10 split was used.

Classification Model

Per the request of our client, our goal was to implement a binary classification model to predict whether an eardrum has OM. In their *Client Statement of Needs*, our client indicated a preference for a CNN. In addition to this, much of the prior work done in this field successfully implemented CNN models [4], [6]. A CNN model is a strong choice for this task, as it's the gold standard for image-related tasks, and it can automatically detect important image features [7]. As a result, we focused on developing 2 potential CNNs that could be used for identifying the presence of OM.

DenseNet Transfer Learning Model

In recent years, studies such as that of Zeng et al. have had success with classifying similar ear diseases using endoscopic images with a CNN approach [6]. In particular, two variations of the DenseNet architecture used in a transfer learning context showed the best trade-off between accuracy and training time (reaching an average accuracy of 95.59% for their 9-class dataset of 20,542 images) [6]. The images used in the study by Zeng et al. share many characteristics with those available for our own classification task, although it is worth noting that the images in their dataset are generally more consistent in terms of lighting and camera angle. This is likely because of their more consistent use of a singular camera system

in generating their dataset. However, their success with the DenseNet architecture suggests that a similar approach could yield useful results for our task.

For this reason, we decided to make use of the DenseNet architecture (in *Figure 8*) for our binary classification task. We followed a similar transfer learning approach to that of Zeng et al., in which we opted to use a version of DenseNet pre-trained on the 1000 class ImageNet dataset. We altered this pre-trained model to use two output classes in the final linear layer, and downscaled our images to a resolution of 224x224 pixels, as required by DenseNet. The weights of early layers in the model were frozen while gradient descent was performed to find optimal weights in the adjusted linear layer.

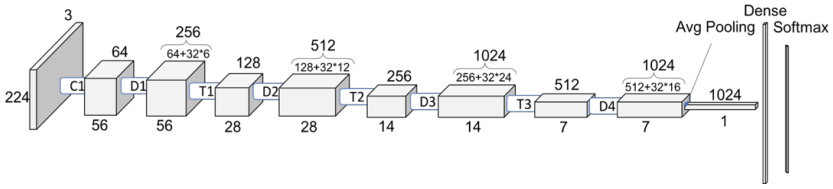


Figure 8: DenseNet architecture [8].

This model was trained for 50 epochs using a train/validation/test split of 60/30/10 with various combinations of the following hyperparameters: LR, momentum, and batch size. The results of hyperparameter tuning are described in *Appendix B*, alongside graphs for the best-performing combination (labelled *honest-grass-3*). This hyperparameter combination was implemented after it exhibited the best overall performance across both training and validation data in terms of accuracy and converging loss.

AlexNet Model

A mini literature review was conducted to identify prior work done in a similar domain, namely use CNNs to classify medical images. One relevant paper outlined a 2018 study that developed an image-processing model to aid in the identification and programming of cochlear implants [9]. As part of this model, Zhang et. al proposed the use of an AlexNet model on 2000 head Computed Tomography (CT) images to classify them based on which ears were present [9]. This model achieved a classification accuracy of 95.97% [9].

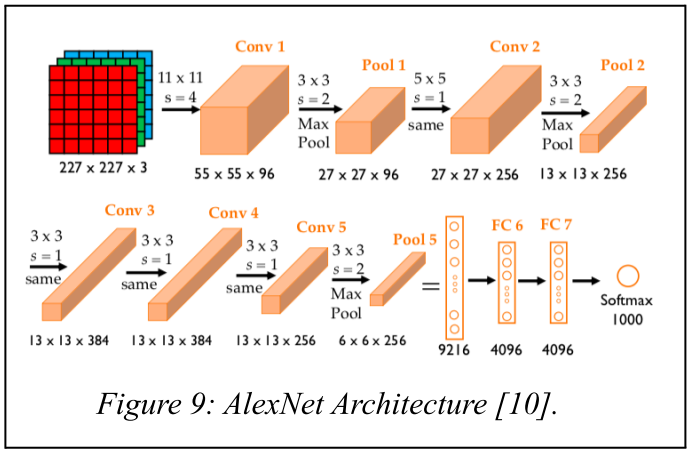
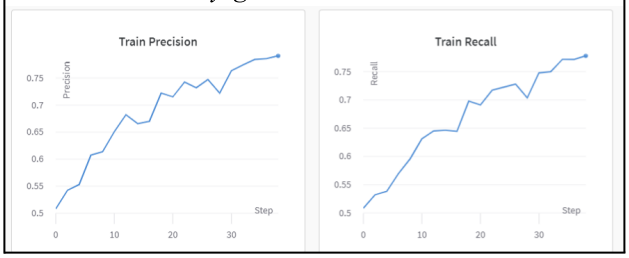


Figure 9: AlexNet Architecture [10].

In their dataset, many images were taken from different angles/positions and had different levels of granularity and contrast, much like ours [9]. Given the domain and dataset similarities, we felt it was appropriate to test an AlexNet-based implementation.

Figure 10: Train Precision and Recall for dainty-glade-11 on AlexNet.



Our AlexNet-based model followed the architecture depicted in *Figure 9,* except for modifying the size of the first FC layer from 9216 neurons to 2304 neurons to accommodate the size of our images. A cross-entropy loss function with SGD optimization was used.

A standard train/validation/test split of 60/30/10 was used. This model was trained using 20 epochs and a batch size of 200. The LR, momentum and weight decay were treated as hyperparameters. Before

training, the images were resized to 150x150 pixels due to RAM constraints on *Google Colab*. Hyperparameter tuning and training results are summarized in Table 1. The corresponding graphs for the highest-performing model, *dainty-glade-11*, are presented in *Figure 10.* Although it appears that training run *youthful-breeze-9* had the highest performance, its loss curve showed no evidence of convergence over time (see *Appendix C.a.2.iv*). See *Appendix C.a.2* for the remaining results and graphs.

Table 1: Subset of Hyperparameter Tuning and Validation (Val.) Results on AlexNet using Dataset B

Training Run	LR	Momentum	Weight Decay	Val. Loss	Val. Precision	Val. Recall	Val. Accuracy
astral-aardvark-7	0.02	0.01	0.06	1.187	0.5207	0.5134	0.5519
youthful-breeze-9	0.001	0.1	0.006	1.099	0.5796	0.5426	0.5861
dainty-glade-11	0.01	0.1	0.07	1.196	0.5715	0.5725	0.5751

V. RESULTS

Object Detection Model

While further work is necessary to achieve a deployable implementation that can improve the quality of images captured by Remmie's users, the model showed strong evidence of learning and an ability to locate landmarks much more reliably than random chance. Specific architectural tuning examples will be presented, followed by tangible results from a model shortlist.

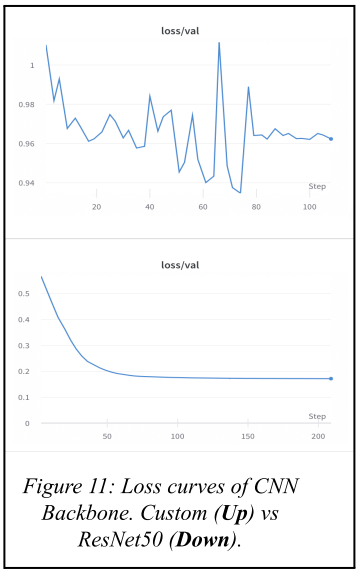


*Figure 11: Loss curves of CNN Backbone. Custom (**Up**) vs ResNet50 (**Down**).*

In *Figure 11*, we see an otherwise identical network built on a custom CNN extractor compared to ResNet50. At a glance, it appears that the latter is significantly more effective. While we did not have time to develop a full rotated-bounding-box/segmentation model as hoped, our result was able to very accurately predict malleus/eardrum centre coordinates in a large number of cases. Five runs were chosen for deeper analysis, and a large sample of test predictions is provided in *Appendix D*. Note that when presenting the average prediction accuracy, mean-percent-distance (MPD) is calculated as the Euclidean distance between two points (prediction vs ground truth) on a 0-1 normalized square. It can be shown that two randomly chosen points exhibit an MPD of ~0.5 [11]. Based on quantitative and qualitative results, *summer-snowball-25* is our best-performing run.

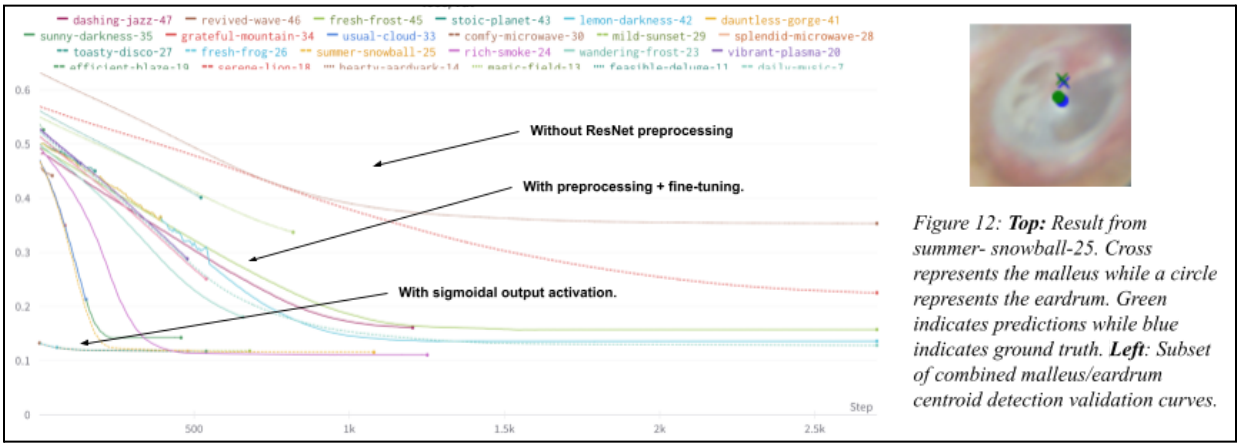


*Figure 12: **Top:** Result from summer- snowball-25. Cross represents the malleus while a circle represents the eardrum. Green indicates predictions while blue indicates ground truth. **Left**: Subset of combined malleus/eardrum centroid detection validation curves.*

Table 2: Summary of Test Results for Object Detection Model

Model	rich-smoke-24	summer-snowball-25	lemon-darkness-42	toasty-disco-27	fresh-frost-45
Malleus MPD	0.247	0.246	0.290	0.240	0.247
Eardrum MPD	0.159	0.161	0.218	0.160	0.159

Classification Model

Table 3: Summary of Test Results for Selected DenseNet and AlexNet Trained on Dataset B

Architecture	Loss	Precision	Recall	Accuracy
DenseNet	0.6877	0.4598	0.0413	0.6104
AlexNet	0.3617	0.5145	0.5158	0.5205

Despite hyperparameter tuning, both preliminary classification models did not achieve our third objective. Several trained models demonstrated a lack of convergence in the loss value (see *Appendix B* and *Appendix C.a*). Accuracy metrics for the pre-trained DenseNet model were optimistic in training and validation, though accuracy dropped to ~61% when run on the test set (see Table 3). The loss curves for DenseNet suggest that the model was able to learn and was not overfitting, though the model was not able to fully converge to provide any dependable results. In the AlexNet model, there was evidence of learning in the loss curve (*Appendix C.a.2.v*). Despite appearing underfit, the precision, recall, and accuracy graphs presented in *Figure 10* and *Appendix C.a.2.v* have defined inflection points, indicating the model was appropriately trained. For both models, recall and precision were consistently low, indicating "guessing" of some labels, which is likely partially caused by **Dataset B's** class imbalance.

Although AlexNet showed poorer accuracy across training, validation, and test data, DenseNet showed very low recall and precision, suggesting that the model is unable to make reliable predictions for the positive (OM) class. This is a key metric for Remmie's application based on the need for FDA approval, and hence AlexNet is the better-performing model overall.

Based on the success of prior work, we hypothesized that running the best-performing architecture on the more diagnosable images would yield higher results. We thus moved forward by testing the AlexNet model against **Dataset C**. This model used the same batch size, epochs, and hyperparameters as the first AlexNet model. Images were resized for the same reason as well. Hyperparameter tuning and training results and graphs are presented in *Appendix C.b*. On all permutations of the model, training run *fluent-firefly-8* had the best overall performance. Therefore, this model was used on the test dataset. It had a test loss of **0.3138**, and had accuracy, precision and recall values of **0.7143, 0.6929** and **0.5744** respectively.

VI. DISCUSSION

Data Processing

While we saw some degree of success with the current state of our data processing, we believe this is a significant area for improvement that could enhance the quality of modelling. Specifically, there are three factors that we believe are worth considering for future work. Firstly, a more robust and centralized labelling environment would be beneficial. Given the overhead our partitioned workaround created, it was difficult to divide work, review results, and experiment with new approaches. Investment in strong infrastructure here could lead to a richer process later on.

Secondly, a significant level of noise was observed in the original dataset. This was difficult to remove due to (a) a lack of domain expertise, and (b) the fact that, unlike purely statistical data, outlier removal is much more complicated for images (barring manual labelling). Additionally, as this forced our filtering towards obviously identifiable cases, it likely introduced a bias into the dataset.

Finally, while we performed a subset of labelling and filtering, our ability to do so was limited by both time and domain knowledge. Ultimately, the models that consumed this data relied on a much smaller subset of the original dataset; improvements in quality/volume could lead to higher performance.

Object Detection Model

Several families of models emerge from the training runs in *Figure 12*. These correspond to the tuning of structural hyperparameters such as the preprocessing used, layers present, and training method. Within each family, variation appears due to tuning routine parameters such as LR, batch size, and momentum.

Although the loss function used was kept consistent, solely loss-based comparisons between families were not particularly useful. For example, models that used sigmoidal activation naturally had a lower initial loss, but actually appeared to perform worse than other alternatives when accounting for rotated/far-from-centre images (see *Appendix D*).

In general, there is a qualitative aspect to evaluating these models: bias-oriented metrics such as recall and precision are well-formed in classification, but their analogues here are difficult to characterize. For example, the performance of the model based on attributes such as left/right ear, artifacts such as glare, centre distance of the features, etc., is difficult to quantify without more (time-consuming) labelling.

One interesting result was the apparent convergence of all models to a loss of ~0.10-0.15. While modifying factors such as network size and other hyperparameters had an impact on the overall training-curve shape, this appeared to be a hard floor on model performance. Further analysis is necessary, but one possible explanation is that this is a generalization limit given the small dataset used.

Classification Model

Despite improvement in classification results, both models are indicative of training on a dataset that is insufficient for diagnostic classification. As documented in prior sections, we were aware of this potential problem, and performed heavy data processing to mitigate its effect, which evidently was insufficient. Additionally, limitations brought on by local machines may have impacted model performance. Due to RAM constraints, images had to be made smaller, which may have exacerbated image quality. This also constrained the batch size and the number of epochs used. Ideally, with more RAM available, a larger image size, larger batch size and more epochs would have been tested.

Model performance was also limited by the lack of symptom labels, which were initially going to be provided with each data sample. As OM has very clearly defined symptoms [1], [2], we believe that having access to these labels would greatly boost model performance. This was confirmed by Remmie's data expert, who noted that symptom labels are a significant part of their current diagnostic process.

DenseNet

As previously mentioned, the pre-trained DenseNet model performed well in certain metrics, like accuracy, but fell short compared to the requirements for precision and recall. This indicates that the model is not able to make predictions reliably for positive cases of OM. One possibility for the poor performance of this architecture is that eardrum images vary too drastically from the images used in the ImageNet dataset, preventing the model from extracting valuable features. Loss metrics taken during the training process suggest that the model may have reached better performance after training for more

epochs, but as previously mentioned, our training process was resource-constrained. At the same time, this would not necessarily improve the poor precision and recall that the DenseNet model was achieving.

AlexNet

The model trained on **Dataset C** improved significantly, as evident by the convergence and improvement in the training and validation curves presented in *Appendix C.b.* Furthermore, test accuracies, precision, and recall were significantly higher than our highest performing **Dataset B** model (71% vs. 52%, 69% vs. 51%, and 57% vs. 52%, respectively). The precision and recall values indicated that this model was not simply "guessing" a label, and was learning from the training samples, unlike the first AlexNet model. This is an especially important result given the class imbalance present in both datasets, which suggests performance may be improved with suitable data.

Proposed Solutions and Comparison to Similar Products

Based on these results, we believe our Object Detection model has met the success criteria defined early in this document. While further development (additional features, data) are required for production, we have demonstrated beyond random chance that learning is feasible with our architecture. During our literature review, we encountered no prior work in this field similar enough for comparison. Unfortunately, the Classification Model was not able to achieve the third objective. However, we would propose the usage of the AlexNet model as it was more successful. Current work suggests that this is a novel approach to our task. Much of the prior work described in **Section IV** (Methods) was conducted with datasets more suited for diagnostic classification (with diagnosable images, symptom labels, etc.). Therefore, these models produced results with higher accuracy, precision, and recall values. However, as demonstrated by the development of an AlexNet model on **Dataset C**, we anticipate that as more suitable data with symptom labels are collected, model performance will improve significantly. Additionally, as much of the prior work has implemented a DenseNet model [6], we also recommend re-testing the DenseNet model with sufficient data.

VII. IMPLEMENTATION

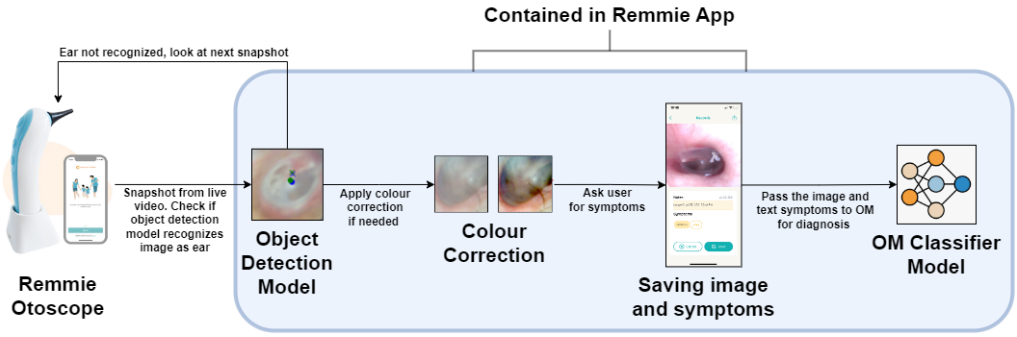


Figure 13: Suggested use-case of our models in Remmie's ecosystem.

Data Processing

To utilize the model proposed in this report, Remmie should set up a data processing procedure to produce training data *en masse*. We recommend Remmie set up a Label Studio instance on their server, as doing so would make their labelling task easy to manage for large datasets. The instance would pull in unlabelled data from Amazon S3 storage and store the labels in S3, separately from the original image, as illustrated in **Section III** (Data) of this report. The standard of selecting and labelling usable data is also defined in **Section III**. Considering the size of their dataset, we recommend Remmie assign domain experts to the data labelling task or use third-party services such as Amazon Mechanical Turk or Scale AI for efficiency.

After selecting usable data and labelling them, Remmie could use the image pre-processing and data augmentation techniques we used, also introduced in **Section III**. These image pre-processing steps introduced more separation between the different classes of data and facilitated the training of our model. With more data labelled and processed, Remmie can train our model further and increase its precision and recall performance. If they have access to platforms with more memory than ours, they can also adjust our model to make it take in larger-sized and higher-quality images as inputs.

Object Detection Model

The purpose of our Object Detection Model is to assist Remmie's users with capturing clear images of their eardrums at home. Thus, it needs to provide near-real-time responses to the users. Based on ResNet50's model size (<200 MB), we believe our Object Detection Model could be deployed alongside Remmie's mobile app. This way, Remmie's app can pull images from the otoscope, with a frequency of one image per second or higher, and use our Object Detection Model to process these images locally. When the model detects an eardrum and malleus handle in the centre of the field of view, the app can prompt the user to take an image for diagnosis.

Alternatively, Remmie can host an instance of our Object Detection Model on their server to process the images remotely for users, increasing accessibility for users on devices with less processing power. Either way, with the Object Detection Model providing a steady stream of high-quality images, Remmie can also train the model further. See *Appendix E* for a graphical illustration of this process.

Classification Model

Given its unsatisfactory precision and recall performance, we do not recommend that Remmie implement our AlexNet-based model in its current state. Instead, we recommend Remmie to train it further with a larger dataset containing clearer photos and accompanying symptoms in text format. Once (and if) our Classification Model achieves the precision and recall values Remmie requires, they can deploy it as a component of their app. It will then process locally the images taken by the user (with the help of the Object Detection Model) and produce a classification of whether the user has OM. However, the model is not intended to replace human doctors, and a doctor's review is required for a conclusive diagnosis.

VIII. CONCLUSION

Overall, our results serve as proof of concept for Remmie's objectives. We consider our Object Detection Model successfully able to locate features of the ear, with an accuracy exceeding random chance. While our Classification Model (AlexNet) demonstrates some ability to classify Otitis Media, we consider this an area that requires significant future work to be usable.

However, this work is far from ready to deploy to production. For the reasons discussed above, we believe that a limiting factor is the requirement for a large set of cleaned data, and with this, much better results are possible. For this reason, we strongly recommend implementing a data pipeline, as described.

For the Classification Model, future steps involve the application of the existing models to this larger set of data. A key limitation without our approach is noted, however: our models were trained on a simplified OM/Not OM problem. The direct applicability of this problem to Remmie's app is unknown, and success on more general applications is unknown. For the Object Detection Model, centroid detection was explored as a proof of concept. With this showing promise, a production application should include detection of the size/rotation of objects, to allow richer feedback to users.

We will be providing Remmie with our models via well-documented *Colab* notebooks, as well as our online environment detailing all training runs and experiments. Furthermore, we will be compiling a document summarizing our recommendations on usage, deployment, and overall system architecture.

IX. ATTRIBUTION

Table 1: Attribution Table and Project Responsibility

Team Member	Primary Project Responsibility	Primary Final Report Responsibility
Zahra Nadine Kandola	1. Project Management 2. Communication with Teaching Team 3. Data Labelling 4. Classification Model (AlexNet and early multi-class version)	1. Background and Problem 2. Datasets 3. Sections Relating to: Classification Model 4. Appendix 5. Editor
Aidan Lawford-Wickham	1. Data Pipeline 2. Data Labelling 3. Classification Model (DenseNet and early multi-class CNN)	1. Sections Relating to: Classification Model 2. Editor
Jay Mohile	1. Project Management 2. Communication with Teaching Team 3. Object Detection Model 4. Data Processing and Infrastructure 5. Data Labelling	1. Data Processing and Data Diagrams 2. Sections Relating to: Object Detection 3. Conclusion 4. Editor
Tian Lan	1. Communication with Teaching Team 2. Data Labelling	1. Implementation 2. References 3. Editor
Markus Kunej	1. Project Management 2. User Guide 3. Data Processing and Infrastructure 4. Data Labelling	1. Implementation Diagrams 2. Editor

X. REFERENCES

[1] "Ear Infections in Children," *National Institute of Deafness and Other Communication Disorders*, 16-Mar-2022. [Online]. Available: https://www.nidcd.nih.gov/health/ear-infections-children. [Accessed: 4-Dec-2022].

[2] "Ear infection (otitis media): Symptoms, causes, prevention & treatment," *Cleveland Clinic*, 16-Apr-2020. [Online]. Available: https://my.clevelandclinic.org/health/diseases/8613-ear-infection-otitis-media. [Accessed: 4-Dec-2022].

[3] "Remmie Home Ear-Nose-Throat Monitor Camera Scope / Otoscope," *Remmie Health*, n.d. [Online]. Available: https://www.remmiehealth.com/products/remmie-dolphin-otoscope. [Accessed: 4-Dec-2022].

[4] M. A. Khan, S. Kwon, J. Choo, S. M. Hong, S. H. Kang, I.-H. Park, S. K. Kim, and S. J. Hong, "Automatic detection of tympanic membrane and middle ear infection from OTO-endoscopic images via Convolutional Neural Networks," *Neural Networks*, vol. 126, pp. 384–394, Apr. 2020.

[5] P. Fischer, A. Dosovitskiy and T. Brox, "Image Orientation Estimation with Convolutional Networks," *University of Freiburg*, 2015. [Online]. Available: https://lmb.informatik.uni-freiburg.de/Publications/2015/FDB15/image_orientation.pdf. [Accessed: 4-Dec-2022].

[6] X. Zeng, Z. Jiang, W. Luo, H. Li, H. Li, G. Li, J. Shi, K. Wu, T. Liu, X. Lin, F. Wang and Z. Li, "Efficient and accurate identification of ear diseases using an ensemble deep learning model," *Scientific Reports*, 25-May-2021. [Online]. Available: https://www.nature.com/articles/s41598-021-90345-w. [Accessed: 4-Dec-2022].

[7] A. Dertat, "Applied Deep Learning - Part 4: Convolutional Neural Networks," *Towards Data Science*, 8-Nov-2017. [Online]. Available: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2. [Accessed: 4-Dec-2022].

[8] P. Ruiz, "Understanding and visualizing DenseNets," *Towards Data Science*, 10-Oct-2018. [Online]. Available: https://towardsdatascience.com/understanding-and-visualizing-densenets-7f688092391a. [Accessed: 4-Dec-2022].

[9] D. Zhang, J. H. Noble and B. M. Dawant, "Automatic Detection of the Inner Ears in Head CT Images Using Deep Convolutional Neural Networks," *National Library of Medicine*, 19-Apr-2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6474381/. [Accessed: 4-Dec-2022].

[10] A. H. Reynolds, "Large-scale image recognition: AlexNet," *Anh H. Reynolds*, n.d. [Online]. Available: https://anhreynolds.com/blogs/alexnet.html. [Accessed: 4-Dec-2022].

[11] *Random Point Distance Simulation*. (2022), Jay Mohile. [Online]. Available: https://colab.research.google.com/drive/1O7YAm-Bsfbu3wKlglElKYbkeKsKDoBk-?usp=sharing. [Accessed: 4-Dec-2022].

XI. APPENDIX

A. Additional Tables for Dataset Features and Characteristics

a. Dataset A and B

Table 1: Dataset A and B Features and Characteristics

Binary Condition	% of Binary Classes	Condition	Number of Samples	% of Dataset
Otitis Media	38.43	Acute Otitis Media (AOM)	247	8.21
		Acute Suppurative Otitis Media (ASOM)	248	8.24
		Chronic Suppurative Otitis Media (CSOM)	246	8.18
		Serous Otitis Media (OME)	250	8.31
Not Otitis Media	61.57	Otitis Externa	249	8.28
		Abnormal Pinna	59	1.96
		Foreign Body	249	8.28
		Fungal Infection	250	8.31
		Impacted Wax	250	8.31
		Inflammation of Pinna	223	7.41
		Normal Ears	250	8.31
		Nose Throat Disorders	58	1.93
No Visible Characteristics		Diminished Hearing	180	5.98
		Tinnitus	249	8.28
Total (Binary)		**Otitis Media and Not Otitis Media**	**2579**	**85.7**
Total			**3008**	**100**

b. Dataset C

Table 2: Dataset C Features and Characteristics

Binary Condition	% of Binary Classes	Condition	Number of Photos	% of Total
Otitis Media	40	Acute Otitis Media (AOM)	15	7.14
		Acute Suppurative Otitis Media (ASOM)	15	7.14
		Chronic Suppurative Otitis Media (CSOM)	15	7.14
		Serous Otitis Media (OME)	15	7.14
Not Otitis Media	60	Otitis Externa	15	7.14
		Abnormal Pinna	15	7.14
		Foreign Body	15	7.14
		Fungal Infection	15	7.14
		Impacted Wax	15	7.14
		Inflammation of Pinna	15	7.14
		Normal Ears	15	7.14
		Nose Throat Disorders	15	7.14
No Visible Characteristics		Diminished Hearing	15	7.14
		Tinnitus	15	7.14
Total (Binary)		**Otitis Media and Not Otitis Media**	**180**	**85.7**
Total			**210**	**100**

B. DenseNet Train and Validation Results

a. Tabular Results

Table 3: Hyperparameter Tuning and Validation (Val.) Results on DenseNet using Dataset B

Training Run	LR	Momentum	Batch Size	Val. Loss	Val. Precision	Val. Recall	Val. Accuracy
decent-paper-1	0.001	0.01	50	0.6065	0.2500	0.0521	0.6874
hardy-feather-2	0.001	0.05	50	0.5921	0.3864	0.4435	0.6833
honest-grass-3	0.001	0.01	20	0.6462	0.4523	0.6847	0.6593
northern-lake-6	0.01	0.01	10	0.6413	0.4412	0.1556	0.6362
fresh-bee-7	0.01	0.1	20	0.6523	0.2937	0.4231	0.6884

b. Graphical Results

i. Training and Validation Results for decent-paper-1

ii. Training and Validation Results for hardy-feather-2

iii. Training and Validation Results for honest-grass-3

iv. Training and Validation Results for northern-lake-6

v. Training and Validation Results for fresh-bee-7

C. AlexNet Train and Validation Results

a. Dataset B Models

a.1. Tabular Results

Table 4: Hyperparameter Tuning and Validation (Val.) Results on AlexNet using Dataset B

Training Run	LR	Momentum	Weight Decay	Val. Loss	Val. Precision	Val. Recall	Val. Accuracy
dashing-dew-3	0.01	0.01	0.05	1.668	0.5476	0.5347	0.5683
bright-field-4	0.001	0.09	0.005	1.286	0.5194	0.5053	0.5601
astral-aardvark-7	0.02	0.01	0.06	1.187	0.5207	0.5134	0.5519
youthful-breeze-9	0.001	0.1	0.006	1.099	0.5796	0.5426	0.5861
dainty-glade-11	0.01	0.1	0.07	1.196	0.5715	0.5725	0.5751

a.2 Graphical Results

i. Training and Validation Results for dashing-dew-3

ii. Training and Validation Results for bright-field-4

iii. Training and Validation Results for astral-aardvark-7

iv. Training and Validation Results for youthful-breeze-9

v. Training and Validation Results for dainty-glade-11

b. Dataset C Models

b.1. Tabular Results

Table 5: Hyperparameter Tuning and Test Results on AlexNet using Dataset C

Training Run	LR	Momentum	Weight Decay	Val. Loss	Val. Precision	Val. Recall	Val. Accuracy
quiet-jazz-2	0.01	0.1	0.07	0.8353	0.521	0.521	0.524
dutiful-water-3	0.01	0.01	0.05	0.4712	0.5523	0.534	0.5582
efficient-durian-4	0.015	0.01	0.06	0.4666	0.577	0.5592	0.5788
avid-grass-5	0.01	0.02	0.045	0.6131	0.546	0.5331	0.5548
trim-energy-6	0.01	0.02	0.045	0.1065	0.2705	0.5	0.5411
fluent-firefly-8	0.01	0.03	0.045	0.3804	0.6521	0.5444	0.5788

b.2 Graphical Results

i. Training and Validation Results for quiet-jazz-2

ii. Training and Validation Results for dutiful-water-3

iii. Training and Validation Results for efficient-durian-4

iv. Training and Validation Results for avid-grass-5

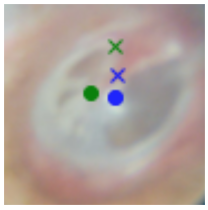
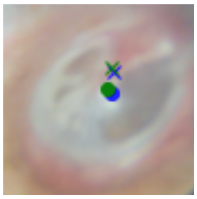
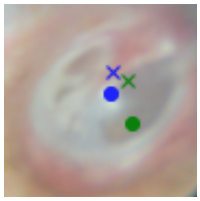
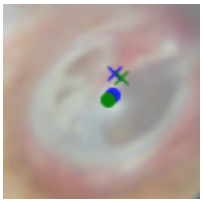
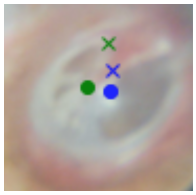
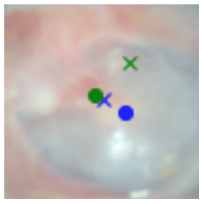
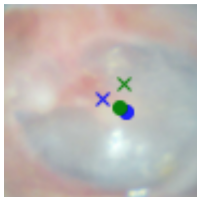
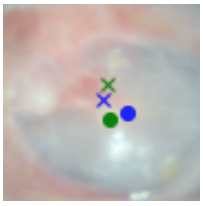
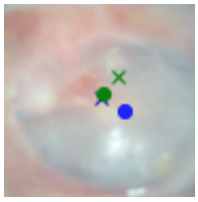
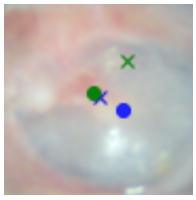
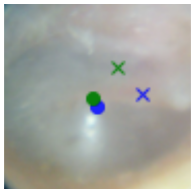
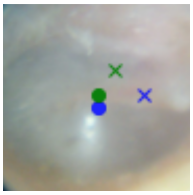
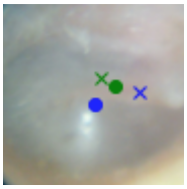
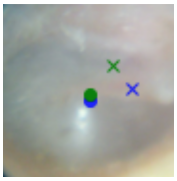
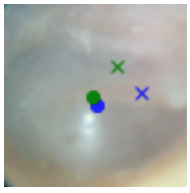
v. Training and Validation Results for trim-energy-6

vi. Training and Validation Results for fluent-firefly-8

D. Object Detection Test Results

In the following sample of results, green indicates predictions while blue indicates ground truth. Additionally, a cross represents the malleus handle while a circle represents the eardrum.

Table 6: Subset of Test Results for Object Detection Model

	rich-smoke-24	summer-snowball-25	lemon-darkness-42	toasty-disco-27	fresh-frost-45
Image 1					
Image 2					
Image 3					

E. Improving Dataset Quality via the Object Detection Model