Effects of audio network architecture on network metamers

Katie Lu
9.58 Projects in the Science of Intelligence
MIT
kqlu@mit.edu

Abstract

Deep neural networks (DNNs) trained on visual tasks have not only achieved humanlike levels of performance, but also make accurate predictions of human error patterns and the activation patterns of human visual cortex. DNNs have also been investigated as models of other sensory systems, including audition. Work in audition often borrows from convolutional model architectures successfully used in vision, but it is unclear if these architectures are ideal as models of audition because of the distinct nature of audio spectrogram input. Due to the difficulty of accessing the human auditory cortex, more indirect methods are needed to assess DNN models for audition. We use one such method of assessing whether model metamers are recognizable to human observers as a metric for the similarity of model and human representations to investigate an alternate convolutional model architecture applied to a simple word classification task. Initial results are null, but more interesting differences may emerge by investigating more complicated models with similar methods.

1 Background

1.1 Deep neural networks (DNNs) as models of sensory systems

Neural network models were initially invented by taking inspiration from the human brain, and as the computing resources to train them have decreased in cost and increased in efficiency, they have begun to show their potential as models for various sensory systems.

Neural network models trained on visual tasks successfully modeled the receptive fields of neurons in visual cortex as early as 1988 (Lehky & Sejnowski, 1988). More recently, they have been shown to model higher cortical regions such as V4 and IT (Yamins & DiCarlo, 2016).

In recent years, successes in other sensory tasks have followed. DNNs trained on auditory tasks such as speech and musical genre recognition have been shown to achieve human-like task performance, and even exhibit human-like error patterns (Kell et al., 2018).

These successes are a promising indicator that using task-optimized deep neural networks to model the human cortex is a valid technique for a variety of tasks and cortical regions (Kell et al., 2018).

1.2 Auditory DNN architecture

Many auditory DNNs which achieve good performance on speech and music tasks use a spectrogram transform of the input audio as their input features (Hershey et al., 2016; Amodei et al., 2015). Because the frequency-based processing done in the human inner ear is well-understood, it is possible to use a special spectrogram transform with parameters chosen to match those of human hearing to

3rd Workshop on Shared Visual Representations in Human and Machine Intelligence (SVRHM 2021) of the Neural Information Processing Systems (NeurIPS) conference, Virtual.

generate a ’cochleagram’ input (Kell et al., 2018; Feather et al., 2019). Using a cochleagram input may assist the model in learning representations closer to those of the human auditory system.

A spectrogram or cochleagram input is convenient because many of the well-tested methods and architectures used in machine vision, such as 2D convolution and 2D convolutional models, can easily be applied to a 2D spectrogram by treating it like a 2D image. Using these methods, researchers have successfully applied model architectures which were first developed for image processing, such as VGG, AlexNet, Inception, and ResNet, to auditory tasks (Hershey et al., 2016).

However, it is unclear whether there is adequate scientific motivation to apply the same 2D convolution techniques that succeed in visual processing to auditory tasks. 2D convolution makes sense for vision because translation invariance intuitively holds in both spatial dimensions, but as mentioned in Kell et al. (2018), it is unclear whether translation invariance should hold along the entire frequency dimension of a spectrogram.

Kell et al. (2018) proceeded to use 2D convolution after observing that it gave better results in experiment, but in order to create DNN models that accurately model human audition, it is important to do a more thorough investigation of different connection architectures along the frequency dimension.

Temporal convolution for auditory tasks makes sense because temporal translation invariance intuitively holds for audio input, but several connection architectures are possible along the frequency dimension:

1. Convolutional, as in the commonly-used 2D convolution methods
2. Fully connected
3. An alternate type of connection architecture which applies different processing to each region along the frequency spectrum, such as a locally connected architecture with unshared weights

1.3 Network metamers for network evaluation

We do not have a detailed understanding of the human auditory cortex because it is located in a place that makes it difficult to access, so it is difficult to evaluate DNN models for audition by directly comparing model activations to neuron activations (Kell et al., 2018). Therefore, we must find creative ways to evaluate auditory models by comparing their behavior to human behavior.

In addition to more superficial approaches such as looking at task performance and whether they exhibit human-like patterns of behavioral errors (Kell et al., 2018), it is possible to probe the representations and invariances a model has learned by generating metamers from the model.

Model metamers, or pairs of physically distinct stimuli that a model perceives as identical, should also be metameric (perceived as identical) for a human observer if the model has learned human-like representations. Freeman & Simoncelli (2011) used this method to evaluate their model of visual processing. However, many of our current neural network models trained with standard methods are far from meeting this objective (Feather et al., 2019).

I trained 2 models, one which uses both temporal and frequency convolution and one which uses temporal convolution and applies different processing to different locations in the frequency spectrum, and generated metamers from both models to see if one of these model architectures promotes the learning of representations that are a closer match for human representations.

2 Previous work

Several other pieces of work have used metamers and/or features synthesized from models using similar gradient ascent methods to evaluate their learned representations and investigate what changes to the model architecture and training process can improve the human recognizability of these learned representations. Freeman & Simoncelli (2011) applied the metamer generation method to demonstrate model-human adherence up to the V2 cortical region for their visual model. Feather et al. (2019) applied the metamer test to look for divergence between auditory model and human representations and between different auditory models. They showed that applying anti-aliasing to counter distortions introduced by the downsampling performed in max pooling layers improved the human recognizability of model metamers. Santurkar et al. (2019) demonstrated that images generated from adversarially robust classifiers depend on more human-recognizable features.

Several pieces of work in audition have considered the question of whether or not to use frequency convolution, but mostly in terms of comparative model performance. Hershey et al. (2016) investigated the performance of speech recognition networks using 2D convolutional architecture versus 1D (temporally) convolutional architectures, but noted that the former gave slightly better task performance. Kell et al. (2018) made similar observations in their work.

3 Methods

The methods and code I used borrowed heavily from Feather et al. (2019)'s audio metamer generation demo, which is publicly available at `https://github.com/jenellefeather/model_metamers`.

3.1 Speech recognition models

I initially planned to search for publicly-available pretrained speech recognition models with the types of architectures I wanted to investigate. However, because most of these models either only use 2D convolution, were not available in TensorFlow 1 and would be difficult to adapt to work with Feather et al. (2019)'s code, or some combination of both, I decided to run my experiment with simpler models I would train myself.

My audio models were based on the TensorFlow 2 simple audio keyword recognition demo (Abadi et al., 2015), available at `https://www.tensorflow.org/tutorials/audio/simple_audio`. I modified the tutorial code to use a cochleagram to preprocess the input audio instead of TensorFlow's native STFT implementation, using the tfcochleagram module written by Feather et al. (2019), available at `https://github.com/jenellefeather/tfcochleagram`.

My 2D model otherwise used the same architecture as the demo model, with 2 convolutional layers with 3x3 kernels, 2x2 max pooling, two fully-connected layers, and 2 dropout layers for regularization. I found during my research that neither PyTorch nor TensorFlow has a built-in implementation of a layer that is convolutional in one direction but locally connected in the other. I wanted to use built-in layers only to prevent extra complications importing the model into the metamer generation code, so I decided to use 1D (temporal) convolution with grouping, which is a built-in in TensorFlow 2, to implement an alternate architecture that applies a different transform to each region of the frequency spectrum. These layers used a kernel size of 3 in the time dimension, and split the frequency dimension into 8 'groups' and was fully-connected within each group. The 1D with grouping model used the same architecture as the 2D model, but with 1D convolutional with grouping layers instead of the 2D convolutional layers and fully-connected layers of different sizes to ensure that the model has a similar number of parameters to the 2D model. A visualization of several different temporal connection architectures can be seen here. 1 A full summary of both model architectures can be found in supplemental material A. 2

I trained both models on the mini speech commands dataset used in `https://www.tensorflow.org/tutorials/audio/simple_audio`, which consists of 8000 WAV audio files of people saying 8 different words. I used the same train/validation/test split of 6400/800/800 as the tutorial. I trained the models for 10 epochs with early stopping. The 1D with grouping model achieved a test accuracy of 86%, and the 2D model achieved a test accuracy of 88%.

3.2 Metamer generation

Attempting to load my trained models in TensorFlow 1 to use Feather et al. (2019)'s TensorFlow 1 metamer generation code presented a number of compatibility issues because several layers, including the normalization, resizing, and grouped convolution layers, are not implemented as TensorFlow 1 built-ins. Saving and loading models between TensorFlow versions is much more difficult with custom layers and would have required a lot more debugging to make sure custom layers were functioning correctly, so I rewrote Feather et al. (2019)'s metamer generation code in TensorFlow 2.

I used the same hyperparameters as Feather et al. (2019)'s experiment and optimized a pink noise waveform input (instead of optimizing the cochleagram) to match the model activation from a target input at each metamer generation layer using the Adam optimizer with a learning rate of 0.001 and

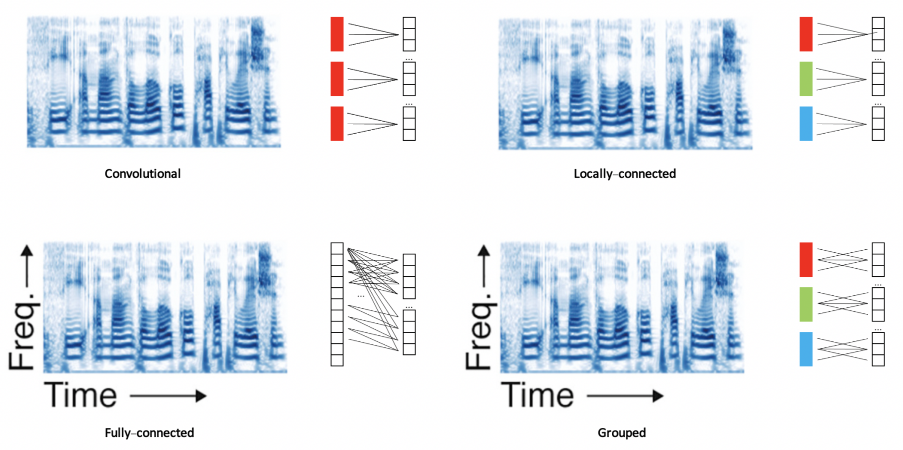

Figure 1: Visualizations of convolutional, fully-connected, locally-connected, and grouped connection along the frequency dimension.

exponential decay at a rate of 0.95. I used 1500 iterations of optimization instead of 15000 because the metamers converged faster for my smaller models. Metamers were generated for the cochleagram input, the ReLU after both convolutional layers, the first fully connected layer, and the final logits classification layer. To ensure that metamer generation succeeded, I verified that the model classifies every metamer the same way as the corresponding target input. Metamers that failed the classification verification test were discarded.

I generated metamers from every model and layer combination for at least 10 target sounds for each of the 8 word categories, for a total of at least 100 metamers per word category.

3.3 Human evaluation of network metamers

I shuffled the generated metamers into 10 conditions of 80 metamers each, equally balanced between the 10 different model-layer combinations. Each condition was presented to the participant in 8 batches of 10 metamers, with one metamer from each model-layer combination per batch. The condition audio was given to each participant as a WAV audio file with each metamer played once without repeats and 2 seconds of silence between metamers. An audio clip demarking each batch was played before each batch began to help participants keep their place.

Participants were fluent speakers of English, mostly native speakers, who were not hard of hearing. They were instructed that they could pause the audio, but not rewind it to hear a sound again. Participants were given a Google form with 80 multiple-choice questions and asked to select which word they thought each sound was supposed to be. I ran 4 conditions on myself, and recruited 4 participants to run another 4 conditions.

4 Results

The metamers generated from the cochleagram input were near-universally identifiable, so I used them as a criterion to discard any conditions in which the participant failed to identify 100% of the cochleagram metamers. This left me with data from 7 conditions, or 560 metamers total, 280 for each model.

My results showed that human recognizability of model metamers decreased at deeper layers for both the 2D model and my 1D with grouping model, dropping to approximately chance performance by the final classification layer for both. 2

The relative recognizability of metamers from the 2 models appears to reverse at the second convolutional layer, with metamers for the 2D model being more recognizable at the first convolutional layer while metamers for my 1D with grouping model are more recognizable at the fully-connected (dense) layer after the convolutional layers.

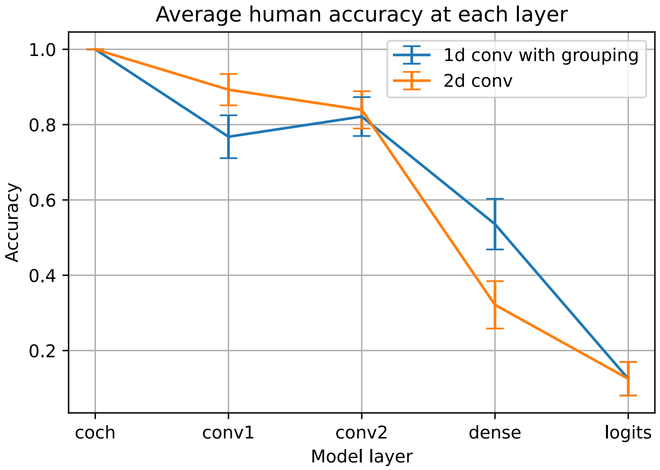

Figure 2: Aggregated human recognition performance at each layer of the 2 models.

Table 1: T-test results for model comparison

Layer	p-value
All	0.724
Cochleagram	n/a
Conv 1	0.079
Conv 2	0.803
Dense	0.022
Logits	1.0

The overall difference between the 2 models was not significant, but the difference between the models at the dense layer was significant, with p=0.022. The difference between the models at the first convolutional layer was close to significant, with p=0.079. 1

Separating the results by target word shows that the performance patterns vary by word, though I did not collect enough data per model, layer, word combination for very strong results. These figures are available in the supplemental materials. 5 Notably, the 1D with grouping model consistently performs better on the words ’stop’ and ’left’, while several other words show the same reversal pattern observed in the overall data.

The confusion matrices for each layer in the 2 models, also available in the supplemental materials, provide another view of the pattern of decreasing metamer recognizability and reversal between the relative performance of the two models. 6 The clarity of the diagonal (accurate classifications) decreases at deeper layers for both models. The 1D with grouping model is more confused at the first convolutional layer, but still has an observable diagonal at the dense layer while the 2D model is more scattered, and both have a very scattered confusion matrix by the last classification layer.

From the overall confusion matrices, it can be observed that the most common misclassifications for the 1D with grouping model were confusing ’yes’ for ’left’ and ’right’ for ’left’. Note that ’yes’ and ’left’ share a vowel, while ’left’ and ’right’ both end with the same consonant. The most common misclassifications for the 2D model were confusing ’go’ for ’right’ and ’down’ for ’right’, with some confusion of ’yes’ and ’stop’ for ’left’ and ’left’ for ’yes’. The shared features between the most confused pairs are less obvious for this model. 3

5 Discussion

Deep neural network models of audition which closely replicate human representations and human behavior on auditory tasks have the potential to guide our investigation of the human auditory cortex,

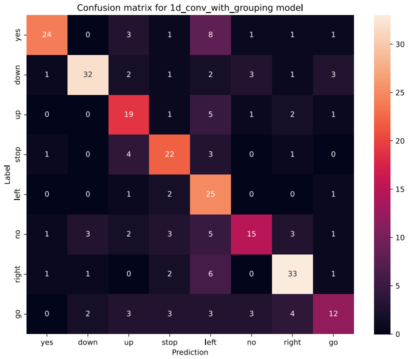

Figure 3: Confusion matrix for human judgments of metamers generated from the 1D with grouping model.

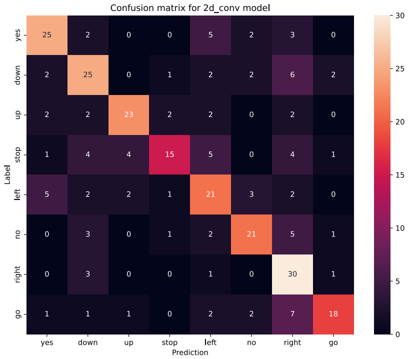

Figure 4: Confusion matrix for human judgments of metamers generated from the 2D model.

which is still understudied compared to other regions of the brain. In order to create good deep neural network models, we must answer certain open questions about what types of model architecture best represent how the human brain processes sound, one of which is whether to apply convolution in the frequency dimension. I used metamer generation, a technique successfully applied in vision to study the quality of model representations, to evaluate the effect of convolutional architecture on model representations.

My results reproduced the commonly observed pattern of representations learned by standard neural network models becoming less human-recognizable at increased depths, and entirely unrecognizable by the deepest layers. My results showed that an alteration of the convolutional architecture alone will not necessarily produce human-recognizable representations at the deepest layers. Creating a model that has very human-salient representations at the deepest layers may require not just an altered model architecture, but also adversarial training to discourage it from using non-robust features, as explored by Santurkar et al. (2019). However, it is still possible that certain architectural changes will still improve the representations learned to some extent, such as in Feather et al. (2019)'s experiment with antialiasing. if they are a better representation of the structure of human cortex.

Though my results failed to show an improvement in the human recognizability of representations from my 1D convolutional with grouping architecture as opposed the more commonly-used 2D convolutional architecture for these specific models, my results don't preclude the possibility of more significant results emerging with future work on more sophisticated models with more controls. The differing performance patterns show that the model hyperparameters do have some effect on the quality of the learned representations, even when both models achieve similar task performance on a test set.

The fact that metamers from my 1D with grouping architecture were most often misclassified for words that shared some feature for the target word is a promising sign that it may be better at preserving certain salient features. In addition, it is interesting that the most significant difference between models occurred at the fully-connected layer after both convolutional layers, indicating that the representations after convolutional processing may differ in some significant way.

I would be careful about assigning too much weight to my initial results because they should ideally be confirmed with more controls. I selected model hyperparameters to ensure both models had a similar amount of parameters, hoping that this would ensure they had about the same amount of predictive power. As Kell et al. (2018) noted, they were unsure if the improved performance they saw with 2D convolution was due to the architecture or if the reduced parameter count served to regularize their model, so I believed matching parameter count would be a good control to include. However, this meant that the layers of the two models were of different sizes. The fully connected layer after the convolutional layers, for example, was of size 512 in the 1D with grouping model but only 128 in the 2D model, which raises a valid question about whether the representation learned by the 1D with grouping model was significantly more human-recognizable because of the convolution architecture or the increased size. Ideally, future experiments would have more thorough controls for both number of parameters and representation size to get a clearer view of what effect is caused by architecture alone.

My current experiment reproduces work done by Feather et al. (2019) and applies their metamer generation method to evaluate a different question about model architecture, but much remains to be done. The first step would be to create more complex models that achieve closer to human performance on the speech recognition task, and train them on a larger dataset with more categories, such as the full-sized speech commands dataset (Warden, 2018). Any differences that convolution architecture can induce will be more likely to reveal themselves in a larger model that has more capability to mimic human auditory processing. In addition, more categories will allow us to do a more detailed evaluation of the features the models are learning and whether they resemble the phonetic features that humans use to process speech.

I would also want to spend time writing and debugging custom layers to perform a locally-convolved transform in frequency to test if there is a significant difference between that approach and the grouping workaround I used for this project. As another control, I would also want to train and evaluate a model which is fully connected in the frequency dimension. My metamer verification method also applied a weaker definition of metamerism, as it accepts all metamers that the model classifies the same way, and not only those which the model perceives as identical (has the exact same activation at the target layer). This criterion is more practical to apply in experiment, but for future work I could use more sophisticated metamer validation methods than prediction matching alone. Verifying that the Spearman R correlation between the metamer activation and original activation is outside the null distribution is one of these methods which (Feather et al., 2019) applied in their work.

A more distantly-related line of work that would be interesting to pursue in the future would be to train a speech recognition model with adversarial training, which is known to induce robust features in visual models, and investigate the representations the model learns to see if the same effect is induced in auditory models.

6 Conclusion

Our initial investigation of alternate convolutional architecture for auditory DNN models has not yet shown significant proof in favor of one architecture over another, but metamer evaluation reveals some trends that could be investigated further with more sophisticated models. Much work remains to be done before deep neural networks will be fully adequate models of human auditory cortex, but the method of evaluating metamer transfer can continue to be a helpful method for evaluating their learned representations as their development advances.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse H. Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Y. Hannun, Billy Jun, Patrick LeGresley, Libby Lin, Sharan Narang, Andrew Y. Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang, Bo Xiao, Dani Yogatama, Jun Zhan, and Zhenyao Zhu. Deep speech 2: End-to-end speech recognition in english and mandarin. *CoRR*, abs/1512.02595, 2015. URL `http://arxiv.org/abs/1512.02595`.

Jenelle Feather, Alex Durango, Ray Gonzalez, and Josh McDermott. Metamers of neural networks reveal divergence from human perceptual systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/ac27b77292582bc293a51055bfc994ee-Paper.pdf`.

Jeremy Freeman and Eero P. Simoncelli. Metamers of the ventral stream. *Nat Neurosci*, 14(9): 1195–1201, 2011. doi: https://dx.doi.org/10.1038\%2Fnn.2889.

Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin W. Wilson. CNN architectures for large-scale audio classification. *CoRR*, abs/1609.09430, 2016. URL `http://arxiv.org/abs/1609.09430`.

Alexander J.E. Kell, Daniel L.K. Yamins, Erica N. Shook, Sam V. Norman-Haignere, and Josh H. McDermott. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644.e16, 2018. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2018.03.044. URL `https://www.sciencedirect.com/science/article/pii/S0896627318302502`.

S R Lehky and T J Sejnowski. Network model of shape-from-shading: neural function arises from both receptive and projective fields. *Nature*, 333(6172):452–4, 1988. doi: https://doi.org/10.1038/333452a0.

Shibani Santurkar, Dimitris Tsipras, Brandon Tran, Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Computer vision with a single (robust) classifier. *CoRR*, abs/1906.09453, 2019. URL `http://arxiv.org/abs/1906.09453`.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *CoRR*, abs/1804.03209, 2018. URL `http://arxiv.org/abs/1804.03209`.

Daniel L K Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19:356–365, 2016. doi: https://doi.org/10.1038/nn.4244.

A Full model architectures

B Additional results

Table 2: 2D Convolutional Model

Layer type	Output Shape	Param #
Resizing	(None, 32, 32, 1)	0
Normalization	(None, 32, 32, 1)	4
Conv2D	(None, 30, 30, 32)	320
ReLU	(None, 30, 30, 32)	0
Conv2D	(None, 28, 28, 64)	18469
ReLU	(None, 28, 28, 64)	0
MaxPooling2D	(None, 14, 14, 64)	0
Dropout	(None, 14, 14, 64)	0
Flatten	(None, 12544)	0
Dense	(None, 128)	1605760
ReLU	(None, 128)	0
Dropout	(None, 128)	0
Dense	(None, 8) 0	
Total params: 1,625,612		
Trainable params: 1,625,610		
Non-trainable params: 2		

Table 3: 1D Convolutional With Grouping Model

Layer type	Output Shape	Param #
Resizing	(None, 32, 32, 1)	0
Normalization	(None, 32, 32, 1)	3
Reshape	(None, 32, 32)	0
Conv1D	(None, 30, 128)	1664
ReLU	(None, 30, 128)	0
Conv1D	(None, 28, 512)	25088
ReLU	(None, 28, 512)	0
Reshape	(None, 28, 512, 1)	0
MaxPooling2D	(None, 14, 256, 1)	0
Dropout	(None, 14, 256, 1)	0
Flatten	(None, 3584)	0
Dense	(None, 512)	1835520
ReLU	(None, 512)	0
Dropout	(None, 512)	0
Dense	(None, 8) 4104	
Total params: 1,866,379		
Trainable params: 1,866,376		
Non-trainable params: 3		

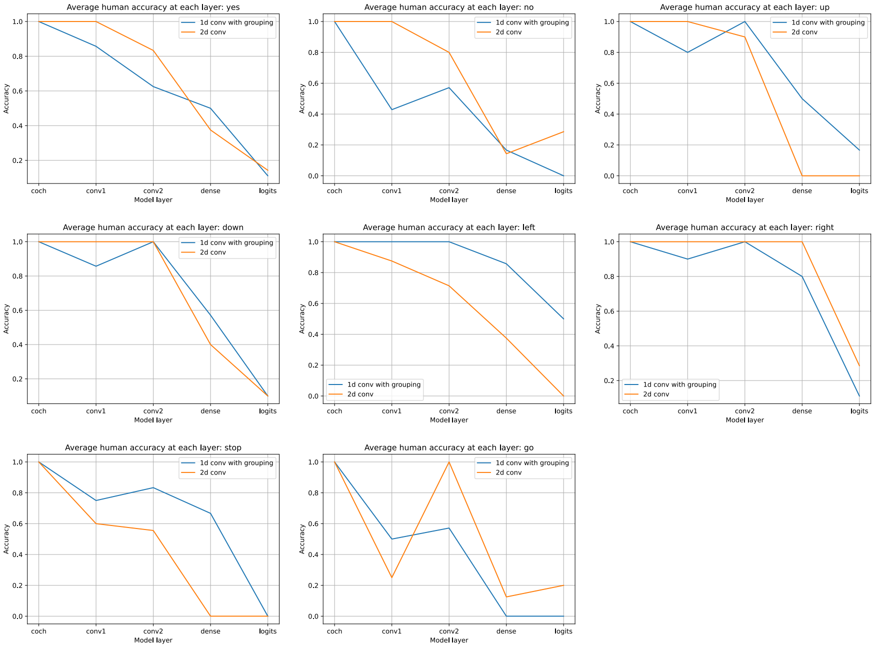

Figure 5: Human recognition performance at each layer of the 2 models, broken down by target word.

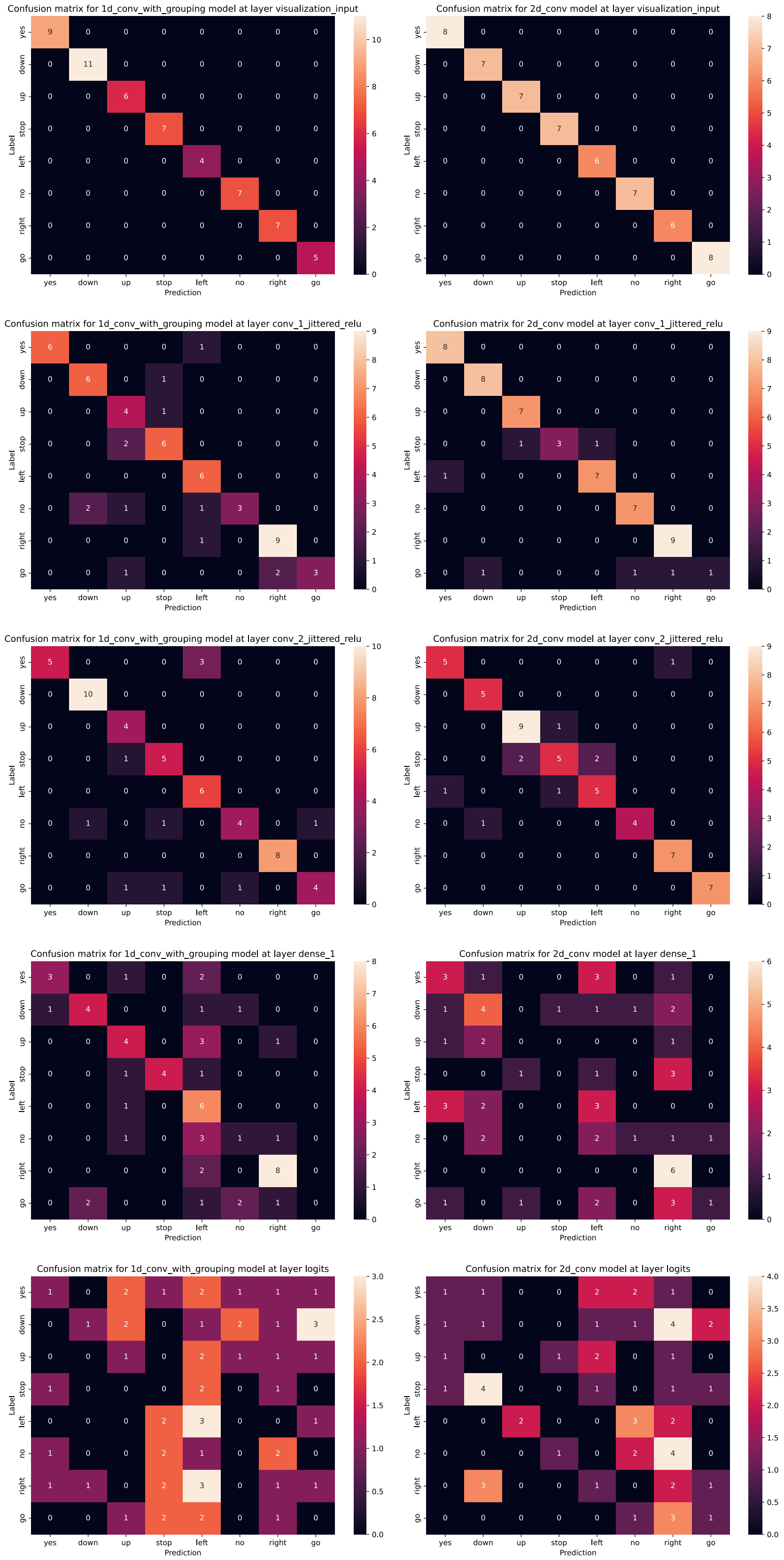

Figure 6: Confusion matrices for human judgments of model metamers at each layer of the 2 models.