Podcast Transcript

EPISODE #120

Silencing the Siren: Real-Time AI Noise Reduction

Published December 29, 2025 • Runtime: 22:04

https://myweirdprompts.com/episode/real-time-audio-ai-edge/

EPISODE SYNOPSIS

In this episode, Herman and Corn dive into the fascinating world of deep neural networks and their role in cleaning up messy audio on mobile devices. From the challenges of "non-stationary" noises like sirens to the engineering trade-offs of running AI on mobile NPUs, they explore how 2025's hardware is changing the way we communicate. They discuss the shift from cloud-based processing to edge computing, the importance of quantization, and why the future of audio intelligence is being built directly on your device.

DANIEL'S PROMPT

Daniel

How do deep neural networks for noise reduction work, and what is the current feasibility of implementing them for real-time or near-real-time use on mobile devices? I'm specifically interested in the trade-offs between on-device (edge) processing versus server-side processing, especially when dealing with challenging background noises like sirens or traffic.

TRANSCRIPT

Corn

Hey everyone, welcome back to My Weird Prompts! I am Corn, and I am sitting here in a somewhat chilly Jerusalem with my brother.

Herman

Herman Poppleberry, reporting for duty. It is definitely a bit blustery outside today, which is actually quite fitting considering the audio our housemate Daniel sent over to us.

Corn

It really is. Daniel was out for a walk in the wind and it sounded like he was fighting a gale just to get his thoughts recorded. But it led him to a really fascinating question about how we actually clean up that kind of mess. He was asking about deep neural networks for noise reduction, specifically for mobile devices.

Herman

Right, and he brought up some really specific challenges like sirens, traffic, and even the sound of a crying baby. He noticed that some of these modern tools can surgically remove those sounds without leaving those weird watery artifacts we used to get in the old days of digital noise reduction.

Corn

It is amazing how far it has come. I remember when noise reduction just meant everything sounded like it was underwater. But Daniel wants to know the nuts and bolts. How do these neural networks actually work, and more importantly, can we actually run them in real time on a phone in twenty twenty-five?

Herman

That is the million dollar question. Or maybe the billion dollar question given how much companies are investing in specialized hardware. To understand how we got here, we have to look at what changed. Traditional noise reduction, what we call DSP or digital signal processing, mostly relied on mathematical filters. They would look for a steady hum, like a fan or white noise, and try to subtract that frequency from the audio.

Corn

But that does not work for a siren or a car horn, right? Because those sounds are always changing pitch and intensity. They are not a steady hum.

Herman

Exactly. Those are what we call non-stationary noises. Traditional math struggles with those because by the time the filter adapts to the siren, the siren has changed its frequency. Deep neural networks, however, do not just look at the math of the wave. They have been trained on millions of examples of what a human voice sounds like versus what a siren sounds like.

Corn

So it is more like pattern recognition than a simple subtraction.

Herman

Precisely. Most of these models today use something called a mask. Imagine the noisy audio as a messy painting. The neural network creates a digital stencil, or a mask, that perfectly fits over the human voice parts of the painting. It then applies that mask to the audio, letting the voice through while blocking everything else.

Corn

I have heard about things like U-Nets and Recurrent Neural Networks being used for this. Is that still the standard as we head into twenty twenty-six?

Herman

U-Nets are still very popular because they are great at capturing both high-level context and fine details. But the real breakthrough lately has been in Transformers and more efficient Recurrent Neural Networks. The challenge with a Transformer is that it can be very heavy on memory, which is a nightmare for a mobile device. But when you are dealing with something like a siren, you need that temporal context. You need the model to remember what happened twenty milliseconds ago to predict what the voice should sound like now.

Corn

That makes sense. If the model knows how a human sentence usually flows, it can almost fill in the blanks if a loud honk briefly covers a syllable. But let us talk about the hardware. Daniel mentioned a paramedic app. If a paramedic is in the back of an ambulance and they need to communicate with a doctor, they cannot have a three-second delay while a server in the cloud processes the audio.

Herman

That is the core of the edge versus cloud debate. In twenty twenty-five, the hardware on our phones has become incredibly specialized. We have these things called NPUs, or Neural Processing Units. If you look at the latest chips from Qualcomm or Google, they have dedicated silicon just for running these matrix multiplications.

Corn

But even with an NPU, running a deep neural network at forty-eight kilohertz in real time is a massive power draw, is it not? I mean, would the phone not just get incredibly hot?

Herman

It definitely can. This is where the engineering trade-offs come in. To make it feasible for on-device use, developers have to use a process called quantization. Basically, instead of using high-precision thirty-two bit floating point numbers for the weights of the neural network, they crush them down to eight-bit integers.

Corn

Does that not make the noise reduction less effective?

Herman

Surprisingly, not as much as you would think. A well-trained model can lose a lot of its precision but still keep its intelligence. It is like the difference between seeing a high-definition photo and a slightly compressed version. You can still tell exactly what is in the picture. For a paramedic app, that eight-bit quantization is the difference between the app running for five hours or the phone dying in forty minutes.

Corn

So, if we are looking at real-time feasibility on an Android device today, is it actually happening? Can a developer just drop in a model and have it work?

Herman

We are right on the edge of it being a standard feature. There are frameworks like TensorFlow Lite and ONNX Runtime that are specifically designed to tap into those NPUs. But there is a huge gap between a high-end flagship phone and a budget device. If you are building an app for paramedics who might be using older hardware, you cannot rely on the NPU being there.

Corn

Which brings us back to the server-side option. But then you have the privacy issue. If it is a medical app, you are talking about sensitive patient data. Sending that to a server for cleaning introduces a whole world of regulatory headaches.

Herman

Not to mention the latency. Even with five-G, you are looking at a round-trip time that might make a conversation feel disjointed. If I say something and it takes five hundred milliseconds to reach you because it had to go to a server in Virginia to have a siren removed, we are going to be talking over each other the whole time.

Corn

It seems like a classic engineering puzzle. You want the quality of the cloud but the speed and privacy of the edge. Before we dive deeper into the specific models that are winning this race, let us take a quick break for our sponsors. Larry: Are you tired of the world being too loud? Do you wish you could just turn off the sound of your neighbors, your city, or your own thoughts? Introducing the Silence-Sphere. The Silence-Sphere is a revolutionary personal isolation chamber made of high-density, bio-acoustic semi-permeable membranes. It looks like a large, opaque plastic bag, but it is actually a masterpiece of engineering. Simply step inside the Silence-Sphere, zip it up, and enjoy the absolute void. Perfect for meditation, high-stakes phone calls, or just hiding from the reality of twenty twenty-five. Warning: The Silence-Sphere is not oxygen-permeable. Use only in short bursts of thirty seconds or less. The Silence-Sphere - find your inner void before your outer void finds you. BUY NOW!

Herman

Thanks, Larry. I think I will stick to my noise-canceling headphones for now. I am not sure I am ready for the inner void in thirty-second increments.

Corn

Yeah, Larry always has a way of making peace and quiet sound like a safety hazard. Anyway, back to the topic. Herman, you mentioned that some models are more efficient than others. Daniel was talking about how a crying baby was surgically removed from his audio. That sounds like a very high-complexity task.

Herman

It is. Crying babies and sirens are particularly difficult because they share a lot of frequency space with the human voice. A low-frequency hum from an air conditioner is easy to separate. But a siren is literally designed to cut through and be heard by humans. It is aggressive.

Corn

So how does a model like RNNoise or DeepFilterNet handle that?

Herman

RNNoise is a classic at this point. It is very small, very fast, and it uses a hybrid approach. It uses some traditional signal processing combined with a small recurrent neural network. It is great for low-end devices, but it can struggle with those complex sounds like Daniel's baby Ezra crying. It might leave some artifacts or make the voice sound a bit robotic.

Corn

And what about the more modern stuff?

Herman

The cutting edge right now is something like DeepFilterNet. It works in the frequency domain but uses a very clever way of predicting both the magnitude and the phase of the audio. In the past, we mostly ignored the phase, which is why things sounded watery. By predicting the phase, these models can reconstruct the voice much more naturally.

Corn

Daniel also asked about the architecture for an app. He suggested two scenarios. One was a real-time call for paramedics, and the other was a walkie-talkie app where you could clean the audio after the user hits send.

Herman

The walkie-talkie scenario is much easier to implement today. If you have even three or four seconds of buffer, you can run a much heavier, more accurate model. You can look ahead at the audio that is coming, which gives the neural network more context. In real-time audio, the model is essentially blind to the future. It only knows what just happened.

Corn

So for the walkie-talkie app, you could use a high-fidelity model on the device, and even if it takes one second to process a two-second clip, the user does not really care. It just feels like a slight delay in the message being sent.

Herman

Exactly. That is a very safe architectural choice for twenty twenty-five. You get the privacy of on-device processing and the quality of a deeper network. But for that real-time paramedic app, you are forced into what we call low-latency mode. Usually, that means the model has to process chunks of audio that are twenty milliseconds or smaller.

Corn

Twenty milliseconds is not a lot of time for a computer to think.

Herman

It is incredibly fast. Most of that time is taken up just moving the data from the microphone to the memory and then to the NPU. The actual inference, the thinking part of the neural network, might only have five or ten milliseconds to finish. If it misses that window, the audio stutters.

Corn

So, if you were building that paramedic app today, would you go on-device or server-side?

Herman

If it is for emergency services, I would argue for a hybrid approach, but with a heavy lean toward on-device. You cannot guarantee a high-speed internet connection in a moving ambulance or a disaster zone. If the connection drops, your noise reduction should not just stop working. I would implement a highly optimized, quantized model on the device.

Corn

And what about the cost? Running these models on a server for thousands of users gets expensive very quickly.

Herman

That is another huge point. If you have a million users and you are processing all their audio on Nvidia H-one-hundreds in the cloud, your burn rate is going to be astronomical. Moving that computation to the user's phone is basically offloading your electricity and hardware costs to them. It is the only way to scale a free or low-cost app.

Corn

It is interesting how the economics of AI are driving everything back to the edge. We spent a decade moving everything to the cloud, and now we are realizing that for things like audio and video, the cloud is just too slow and too expensive.

Herman

It is a full circle. But the software side is catching up. We are seeing things like weight pruning, where developers literally cut out the parts of the neural network that do not contribute much to the output. It is like a digital lobotomy that actually makes the brain more efficient. You can reduce the size of a model by sixty or seventy percent with almost no loss in quality.

Corn

That is fascinating. So, for Daniel's question about sirens and traffic, how specifically do they train for that? Do they just record a bunch of people talking in front of fire trucks?

Herman

Pretty much! It is all about the dataset. You take thousands of hours of clean speech, recorded in a studio, and then you take thousands of hours of pure noise, sirens, honking, wind, rain. Then you use a script to mix them together at different volume levels. The neural network is shown the noisy version and the clean version, and its only job is to learn the transformation from one to the other.

Corn

It is like a student being given the messy rough draft and the final polished essay and being told to figure out the rules of grammar in between.

Herman

That is a great analogy. And because we have so much data now, the models are getting very good at understanding the grammar of sound. They know that a siren has a specific harmonic structure that is different from a human vowel. They can see the siren as a separate object in the soundscape.

Corn

I wonder about the psychological effect of this. If we start cleaning up all our audio, do we lose some of the context? If a paramedic is calling a doctor and the doctor cannot hear the siren in the background, they might not realize how urgent the situation is.

Herman

That is a really profound point, Corn. We often talk about noise as a bug, but sometimes noise is a feature. It provides situational awareness. Some of the more advanced systems are now looking at what they call transparent noise reduction. Instead of deleting the siren, they just turn it down by twenty decibels and move it to the background, so the voice stays clear but the context remains.

Corn

That sounds much more useful for a professional setting. You want the clarity, but you do not want to be in a sensory deprivation tank.

Herman

Exactly. It is about control. In twenty twenty-five, we are moving away from simple on-off switches for noise reduction and toward more intelligent scene management. The phone can identify that you are in a car, or at a construction site, and adjust the filtering accordingly.

Corn

So, to summarize for Daniel and anyone else thinking about building in this space. If you are doing real-time, you need to go on-device with heavy quantization and optimization for NPUs. If you are doing asynchronous communication, like a walkie-talkie app, you have much more breathing room to use high-fidelity models on the device.

Herman

And if you are worried about privacy and cost, the edge is your best friend. The tools are there. Between TensorFlow Lite, Core ML on iPhones, and the various Android neural network APIs, the plumbing is mostly finished. The real work now is in the fine-tuning of the models to handle those specific, aggressive noises like sirens.

Corn

It is a brave new world of silence. Or at least, artificial silence. I think Daniel's experiment with his son Ezra's crying being surgically removed is a perfect example of where this is going. It is not just about making things better; it is about making things possible that were not possible before.

Herman

Absolutely. Imagine a world where a person with a speech impediment can have their voice clarified in real time, or where a journalist can record an interview in the middle of a protest and have it come out sounding like a studio recording. That is the power of these deep neural networks.

Corn

Well, I think we have covered a lot of ground here. From the math of U-Nets to the sketchy isolation bags of Larry. Daniel, thanks for the prompt. It was a great excuse to dive into the current state of audio tech.

Herman

Yes, thanks Daniel. It is always fun to see what you send our way from your windy walks.

Corn

If you want to hear more episodes or send us your own weird prompts, you can find us at myweirdprompts.com. We have a contact form there, and you can subscribe to our RSS feed or find us on Spotify.

Herman

We love hearing from you, even if you are not our housemate. Though being our housemate does get you a faster response time.

Corn

Usually. Unless Herman is deep in a research paper. Anyway, thanks for listening to My Weird Prompts. We will be back next time with more deep dives into the strange and wonderful world of technology and beyond.

Herman

Stay curious, and maybe keep a little bit of that background noise. It keeps life interesting.

Corn

This has been My Weird Prompts. See you next time!

Herman

Goodbye everyone!