Some Uses of AI in Amateur Radio

Charles Brain G4GUO

A personal journey into AI by the author

In the beginning

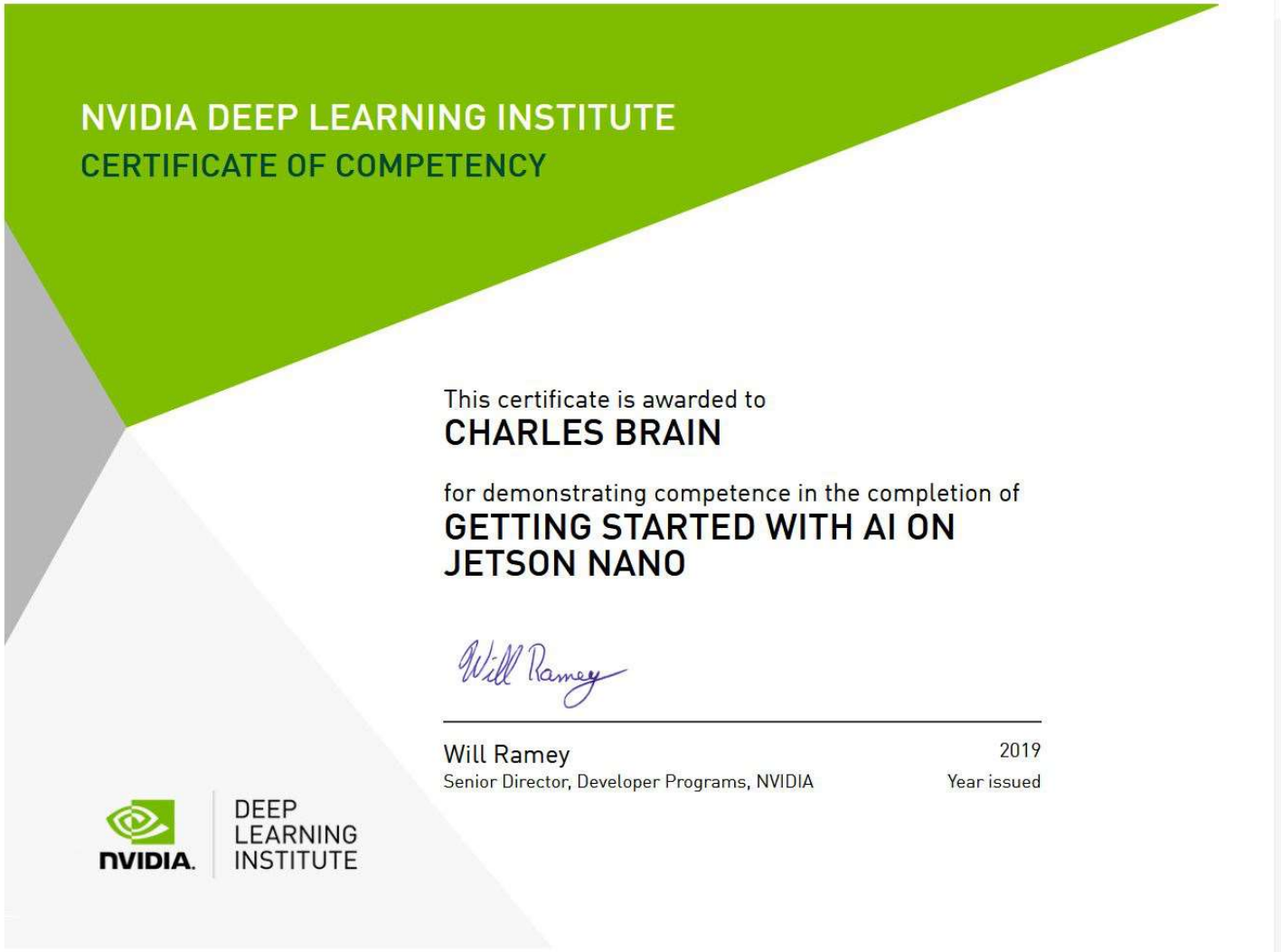
- I bought a book on CUDA
- Then started using GPUs for signal processing. I learnt about CUDA kernels and the different types of memory in a GPU.
- Then tried using them to do LDPC decoding
- Tried the cuFFT library

Then along came AI

- I started off trying to teach myself about perceptrons.
- That was followed by deep neural nets and back propagation.
- AI is an extremely large field and it encompasses everything from simple curve fitting through to subjects like transformers used in Large Language Models.
- Some of it is extremely boring like various clustering algorithms.

Free Education

- I even did an online course



NVIDIA DEEP LEARNING INSTITUTE
CERTIFICATE OF COMPETENCY

This certificate is awarded to
CHARLES BRAIN

for demonstrating competence in the completion of
GETTING STARTED WITH AI ON JETSON NANO

Will Ramey

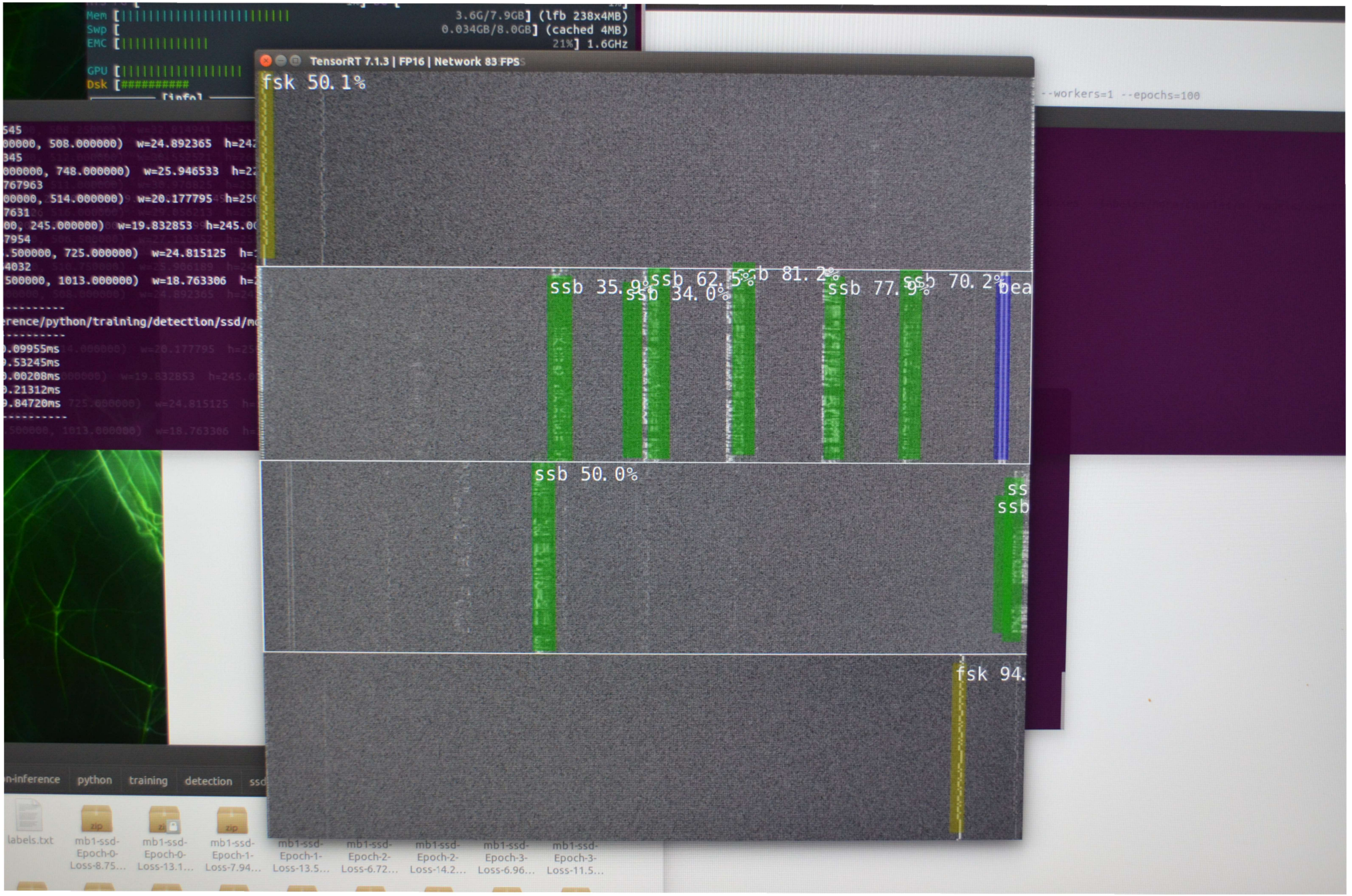
Will Ramey
Senior Director, Developer Programs, NVIDIA

2019
Year issued

NVIDIA DEEP LEARNING INSTITUTE

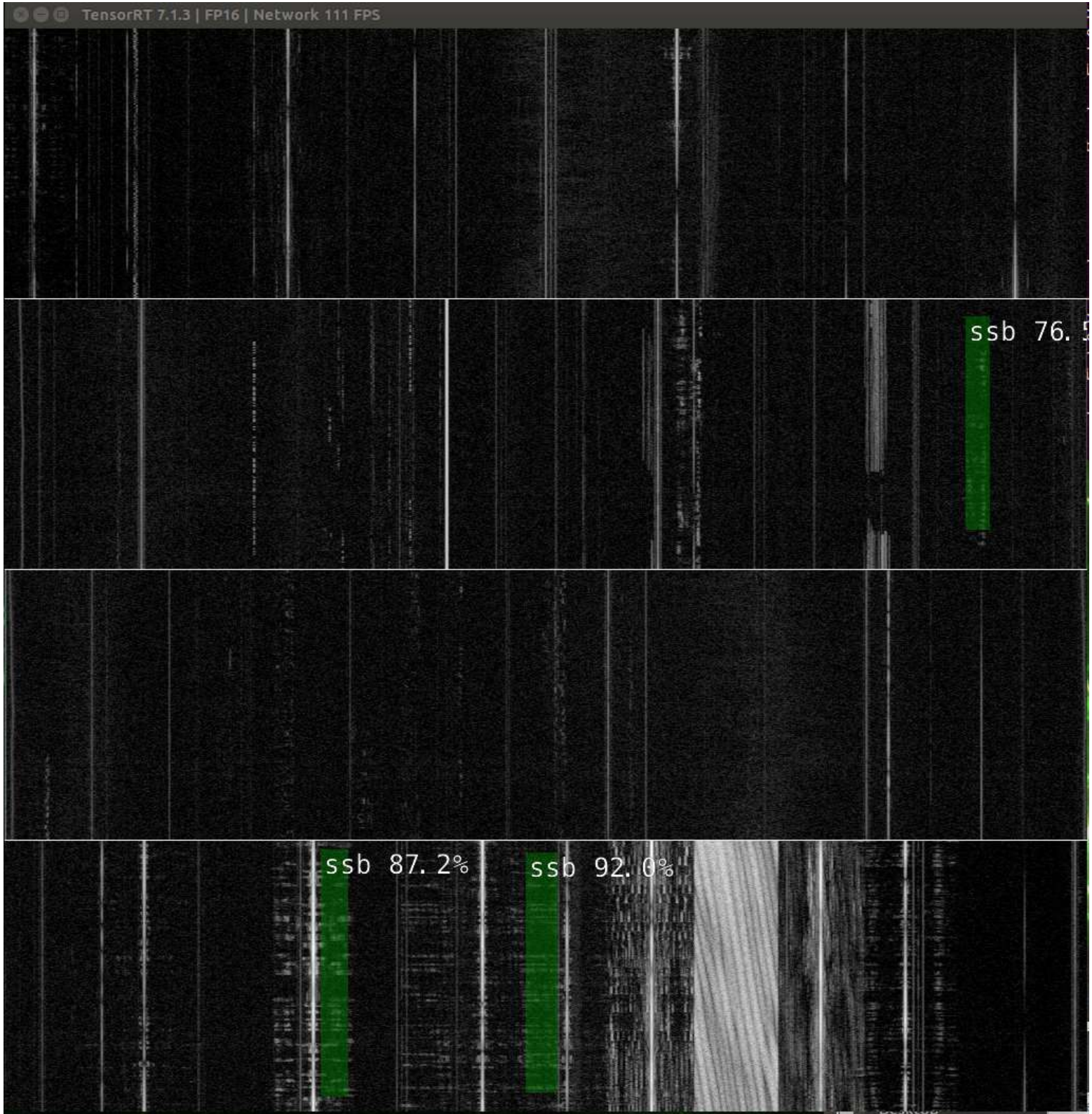
My First Project

- Using transfer learning to re-train mobilenetv2 to recognise radio signals on a waterfall.
- Originally this was a convolutional network designed to recognise everyday objects like people, animals, cars, cats etc.
- Transfer learning involves retraining the higher layers of a model to recognise other objects it was never trained on.


Mem [3.6G/7.9GB] (lfb 238x4MB)
Swp [0.034GB/8.0GB] (cached 4MB)
EMC [21%] 1.6GHz
GPU [
Dsk [
TensorRT 7.1.3 | FP16 | Network 83 FPS
fsk 50.1%
ssb 35.0%
ssb 34.0%
ssb 62.5%
ssb 81.2%
ssb 77.9%
ssb 70.2%
ssb 50.0%
fsk 94.
--workers=1 --epochs=100
python
training
detection
labels.txt
mb1-ssd-Epoch-0-Loss-8.75...
mb1-ssd-Epoch-0-Loss-13.1...
mb1-ssd-Epoch-1-Loss-7.94...
mb1-ssd-Epoch-1-Loss-13.5...
mb1-ssd-Epoch-2-Loss-6.72...
mb1-ssd-Epoch-2-Loss-14.2...
mb1-ssd-Epoch-3-Loss-6.96...
mb1-ssd-Epoch-3-Loss-11.5...

Training data

- This was manually trained by me and took a very long time to create the training files by selecting images by hand and individually labelling them.
- This worked well on QO-100 where all the signals were at similar signal strengths
- On HF this didn't work nearly as well.


TensorRT 7.1.3 | FP16 | Network 111 FPS
ssb 76.
ssb 87.2%
ssb 92.0%

How to fix this?

- Not enough training data was available.
- The solution would be to create synthetic training data using a channel simulator and synthesized modulations.
- AI may not be the best approach to solving this problem. It might be better to do some form of classification system looking at parameters such as signal bandwidth, amplitude and phase changes.

Moving on

- My next idea was to combine Radio and Speech to text.
- Models like Whisper (from OpenAI) are capable of turning human speech into text, they can also translate between languages.
- I tried 2 different ideas, the first was to simply pipe the audio from a receiver into a soundcard and let Whisper do it's thing. I tested it on a Russian Language AM broadcast on HF.

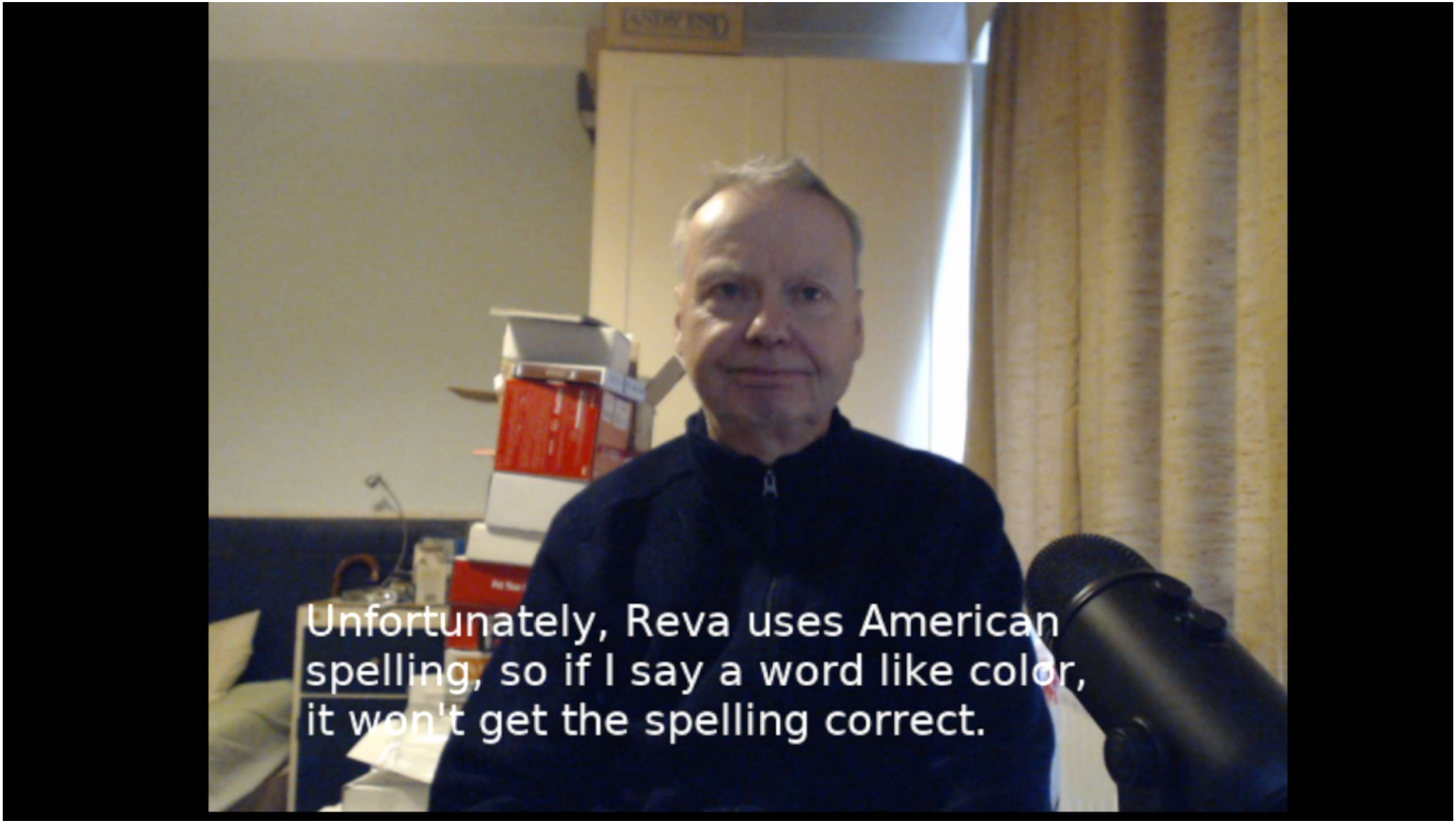
- Next was to connect the output of GNURadio to Whisper, this time listening to NBFM.
- This could be used to monitoring the calling channel for the rare CQ on 2M.
- This works but has trouble with short overs and Amateur Radio call-signs

Lessons learned

- For SSB some form of offset frequency correction is required.
- Pre-processing of speech signals to remove interference.
- Extra model training to recognise Amateur Radio style call-signs like post processing and decoder prompting (I asked Chat GPT how to fix this issue). There are also unofficial forks that can be partially fine tuned.

Adding subtitles

- A while ago I demonstrated adding subtitles to a DATV transmission. This time it was using NVIDIA's Riva running on a Jetson Xavier.
- The text was added to the picture using OBS.


Unfortunately, Reva uses American
spelling, so if I say a word like color,
it won't get the spelling correct.

Other ideas

- It is now possible to clone your own voice when doing text to speech. New open source models that you can run locally are appearing on an almost daily basis.
- When you work someone are you really talking to that person or even a human?

Now, what you really want to hear about

- Using an LLM (Large Language Model)

What have I used them for so far?

- Writing Verilog Code
- Making gnuradio-companion flow graphs
- Building a GnuRadio OOT module
- Telling me how to configure my hand held radio
- Configuring my pFSense router
- Discussing techniques for HF Channel Equalisation
- As a coding assistant helping me write Python.
- Summarising documents

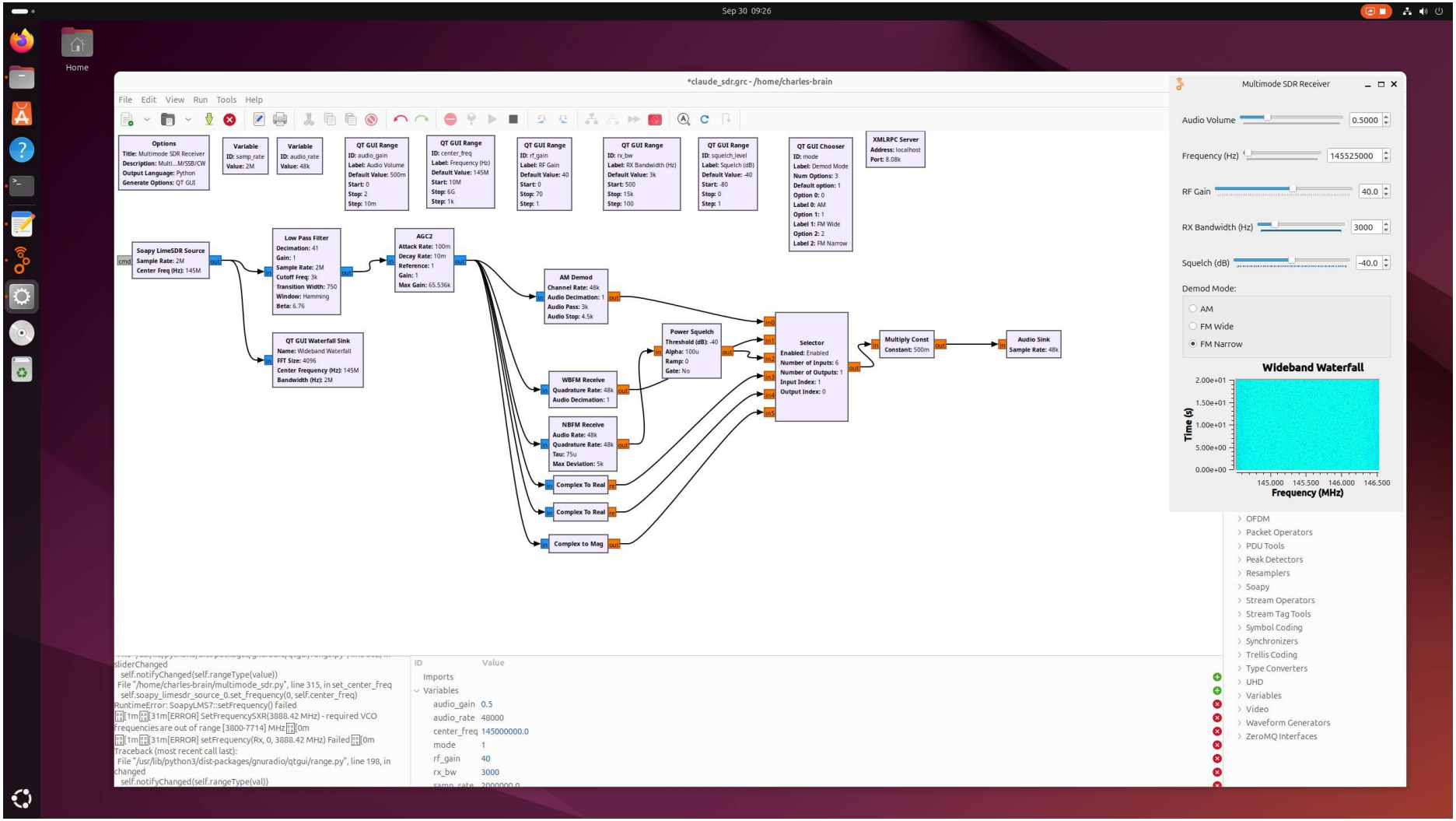
How successful were you?

- I have not found them much good at Vibe coding but rather as an assistant. Especially for languages I am not fluent in.
- Being able to specify a function and have it write it for you or highlight code and say what is wrong, is liberating.
- I have been using them for a couple of years and they are definitely improving.

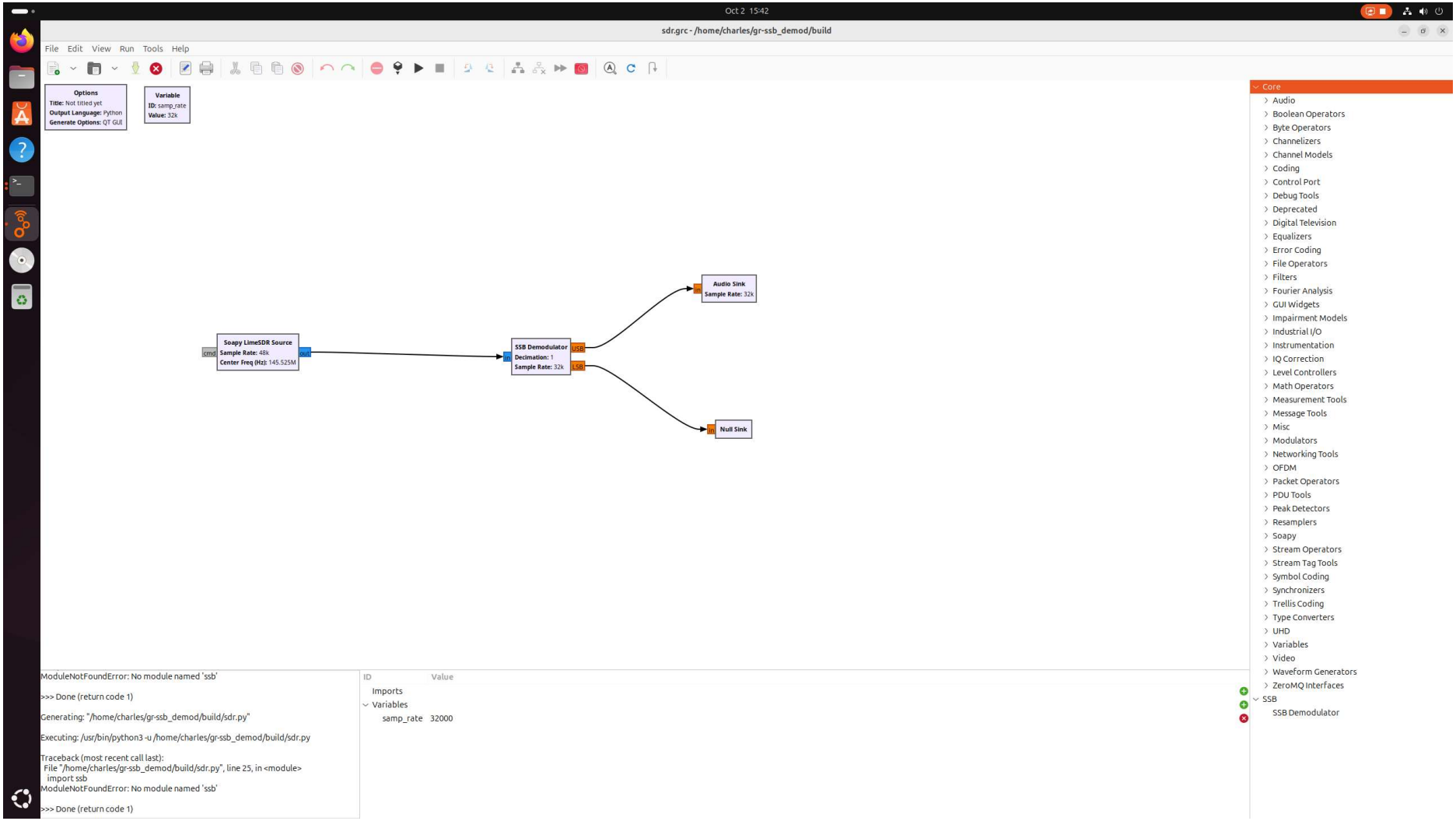
Techniques

- Prompt Engineering
- MCP Model Context Protocol. This is a way for the LLM to call tools that interact with the real world.
- RAG Retrieval Augment Generation. This allows a LLM to have access to locally stored information, usually via a vector database
- Fine tuning. This allows the model to be modified, usually a small number of parameters are modified using techniques like LoRa (Low Rank Adaptation).
- Agents

GRC Flowgraph



GRC OOT module



ChatGPT's ideas

- Exam question generation and tutoring
- Interference analysis, pattern recognition
- Intruder watch
- Propagation predictions
- Antenna design
- EMCOMM message handling
- Hands free operation
- Smart radio data networks

How to get started

- Free tools like ChatGPT, Claude 4.5
- Local models on your graphics card
- Local models on your CPU (slow)
- Ollama, vLLM, llama.cpp, LMStudio, Exo
- Github co-pilot when using VSCode
- Huggingface (repository of open models)