Partial Logbook

of steps and time being wasted on waiting, valuable time that could be used to save lives. ONCO AI+ eliminates these periods of wait, allowing concerned patients to get answers quicker. CT scans, X-rays and other can STILL be used, as ONCO TREAT uses these scans to determine a treatment plan, minimizing the wait and the use of an invasive biopsy procedure, as time is an essential factor in the survival of people that suffer from cancer. Lung cancer can be very aggressive, so it is crucial to identify, diagnose, and prescribe treatment to cure it as soon as possible.

Microwave Interaction w/ Lung Tissues

Lung Tissue Type	Characteristics	Microwave Behavior
Normal lung parenchyma	Low water content (air-filled alveoli)	Low permittivity + conductivity → weak microwave absorption
Tumor tissue (malignant)	High water + cell density	High permittivity + conductivity → strong microwave absorption
Blood vessels	Very high water content	Strong absorption, useful as landmarks
Fibrotic/scarred tissues	Intermediate water	Moderate response
Pleura (lining)	Thin layered	Minimal effect unless thickened (eg. by disease)
Airways (bronchi/bronchioles)	Mostly air; cartilage rings	Very low interaction due to air content

high-intensity/reflective regions

Cancerous lesions vs Normal lung tissue
↳ denser + wetter, altering dielectric constant, absorbs + scatters microwave energy
↳ microwaves reflect less off air-filled areas + more off solid masses like tumors
↳ contrast enables algorithms (CWN) to spot anomalous regions in microwave images

Tissue Permittivity (Approx. @ 2.45 GHz)

Tissue	Relative Permittivity (ε_r)	Conductivity (σ in S/m)
Air	~ 1	~ 0
Normal Lung	5 - 20	~ 0.1 - 0.3

...(lung) 50-70+ ~1.0+

...ric Properties
- How materials interact with an electric field, in response to electromagnetic ...
...e microwaves":
...ative permittivity (ε_r):
- dielectric constant
- measures how much electric energy the material can store
- ↑ ε_r → ↑ energy stored

Each tissue has unique dielectric properties =
(i) Water content (ii) Ionic concentration
(iii) Cell density

Tumors have = ↑ water content, ↑ permittivity + conductivity

...Conductivity (σ):
- measures how well a material conducts electricity
- ↑ σ → ↑ microwave energy converted to heat (loss)

...Loss tangent ($\tan\delta$):
- ratio of energy lost to energy stored
- indicates how dissipative the material is when exposed to microwaves

...Relies on difference in dielectric properties to detect tumors
↳ non-ionizing, functional imaging

...tromagnetic Wave Propagation
...ignals are: Emitted from antennas, Penetrate tissue, Reflected/scattered at ...
...undaries between tissues w/ contrasting dielectric properties

Components of MWI

Component	Role
...rowave source	Generates EM waves @ frequencies like 1 GHz / 2.45 GHz
...enna array	Arranged in a circular/linear/planar pattern around the object (e.g. thorax); used to transmit + received signals
...upling medium	often water/gel is used between the skin + antenna to reduce signal reflection at the skin surface

- Ideal

((Research is for using AI to detect cancer via various methods instead of one approach

- external cancers like skin = physical device NOMA AI
- internal cancers like lung = hardware + imaging tech PULMO AI
- brain cancers (instead of breast) = fMRI + multi-omics for precision therapies

Project forms a vision for how AI can transform cancer detection + treatment across domains

Brain → Multi-Modal AI for Precision Glioblastoma Therapy

↳ Identify subtypes of glioblastoma w/ fMRI + multi-omic data, then suggest precision treatment options using drug repurposing

- Glioblastoma is incredibly heterogenous, meaning treatment is very different
- it is the most common & aggressive brain tumour in adults
- Genetic factors (drivers) contribute to the heterogeneity:
  - EGFR amplification
  - MGMT promoter methylation
  - IDH1 mutation
- Biomarkers (imaging) that are used include:
  - MRI/fMRI: tumor location, edema, perfusion, functional connectivity disruption
  - Diffusion MRI: maps infiltration

My libcamera picamera2 wasn't working bc I had 2 libcamera installs (local build improperly) and system → locally compiled version: /usr/local/lib/aarch64-linux-gnu/libcamera.so.0.5.2 + solution: removed the locally compiled version that caused symbol conflicts

→ now uses properly packaged version from Pi repo, matches Python bindings

→ Instead of rebooting Pi after updating script → restart service file (stop then start)

→ Model kept changing its mind on what dimension size it wanted

scp "file path" havil@ipadress:/home/havil/noma

path to service file = sudo nano /etc/systemd/system/noma_ai.service

Redid my training bc it wasn't working out (Kaggle skill)

Converted to tflite (quantized) on Colab

I transfer my quantized tflite from scp (not ssh) → from computer (local) terminal then Pi password will be prompted → it worked! Solving Problem!

Now working on app (Android Studio) (SDK Platforms)

After downloads, I go to flutter doctor to see if any issues → I have Android 16 + 15 installed, and the SDK Build, Command-line, Platform tools + Emulator & Emulator hypervisor driver

Back to device → deleted previous model, noticed I didn't have proper PyQt5 packages, causing xcb cv2 error → configured venv to directly see via implementation of pyenv.cfg → installed tflite runtime to handle model

launch_eglfs.sh, noma_ai.service, run_noma.sh ✗

↳ launch + run_noma do the same thing but run_noma doesn't have hide cursor or remove conflicting Qt paths, so I am deleting it + updating service bc this could be the cause of errors!

When transferring files from computer → use scp (first exit the Pi) → then put actual path of file ("C:\Users\havil\Downloads\noma_cancer...") → then put host name + ip : then the directory (location → /home/havil/noma_ai

Try to find file type of model, if only data, it didn't load properly or got corrupted (what happened to me)

Having errors w/ model on Pi bc I used a converter too advanced for Pi (select_tf_ops)

→ My EfficientNetV2 model has tf.ops (tf.Mul) that need select_tf_ops to convert to tflite → However I can try to reconvert my model to use a backbone that doesn't deal w/ issue → next time, use EfficientNetB0 → not working so converting w/ float 16 → does not work so retraining model w/ EfficientNetB0

↳ EfficientNet (all variants) has built in mul requiring select_tf_ops, so mobilenet is no instead → Others = ResNet, DenseNet, Xception, VGG16 → MobileNetV3 small bc all other ones need Flex Ops due to their preprocessing layer (when I remove it the whole model dies)

A - 12 hr dry time

Phantom Materials for Lung Mimic

Most phantom designs use gelatin and/or agar as a soft-tissue analog → have a short ~~self-life~~ shelf life → silicone ✓ (more stable, still mimics soft-tissue), (better relaxation time)

Tissue Layer	Mmm	Additive	Mmm Product Name	Mmm Brand
Skin	Silicone	None	Ecoflex 00-20	Smooth-On
Muscle/Fat	Silicone	Graphite Powder	Ecoflex 00-10	↓
Bone	PVC	Gorilla Glue Clear Epoxy*	Life Size Human Skeleton Model	ZENY
Lung	Silicone	None	Sonal Foam 25	Smooth-On

Used Source - Technical Note: The Design And Validation Of A Multi-Modality Lung Phantom

↳ US images were constructed w/ L14-5/38 Linear Array Transducer (Ultrasonix®) on a Sonix Touch system (Ultrasonix®) in a sagittal image

	MR relaxation values						
	T1--3T (ms)	T2--3T (ms)	Adjusted T1 -4.7T (t) (ms)	Material	Additive	T1-t (ms)	T2-t (ms)
Skin	N/A	N/A	N/A	Silicone	None	995.3±12	55.7±0.6
Muscle/fat	965±127	34	1121±148	Silicone	Graphite Powder	1100±29	49.2±0.6
Bone	N/A →			PVC	Epoxy	N/A →	
Lung	→			Silicone Foam	None	821±21	35.1±0.4

Support was added in place of spine via PVC pipings, Regular Clear PVC Cement (Oatey Co.®) and Ebod™ plastic adhesives (Eclectic Products®); Epoxy Gorilla Glue was evenly distributed across all sides of each bone to increase the apparent density for US to more closely mimic in vivo bone

Controlling Relative Permittivity to Mimic Lungs

GOAL: phantom should mimic real lung tissue; dielectric constant (ε_r) need to be as close as possible to that of lung

How Sugar controls ε_r

Sugar (sucrose) increases polarizability of the solution

↳ ↑ polarizable molecules = ↑ ability to store electric energy when exposed to an electric field → higher ε_r

Paper: Human Brain Phantom Modeling: Concentration and Temperature Effects on Relative Permittivity

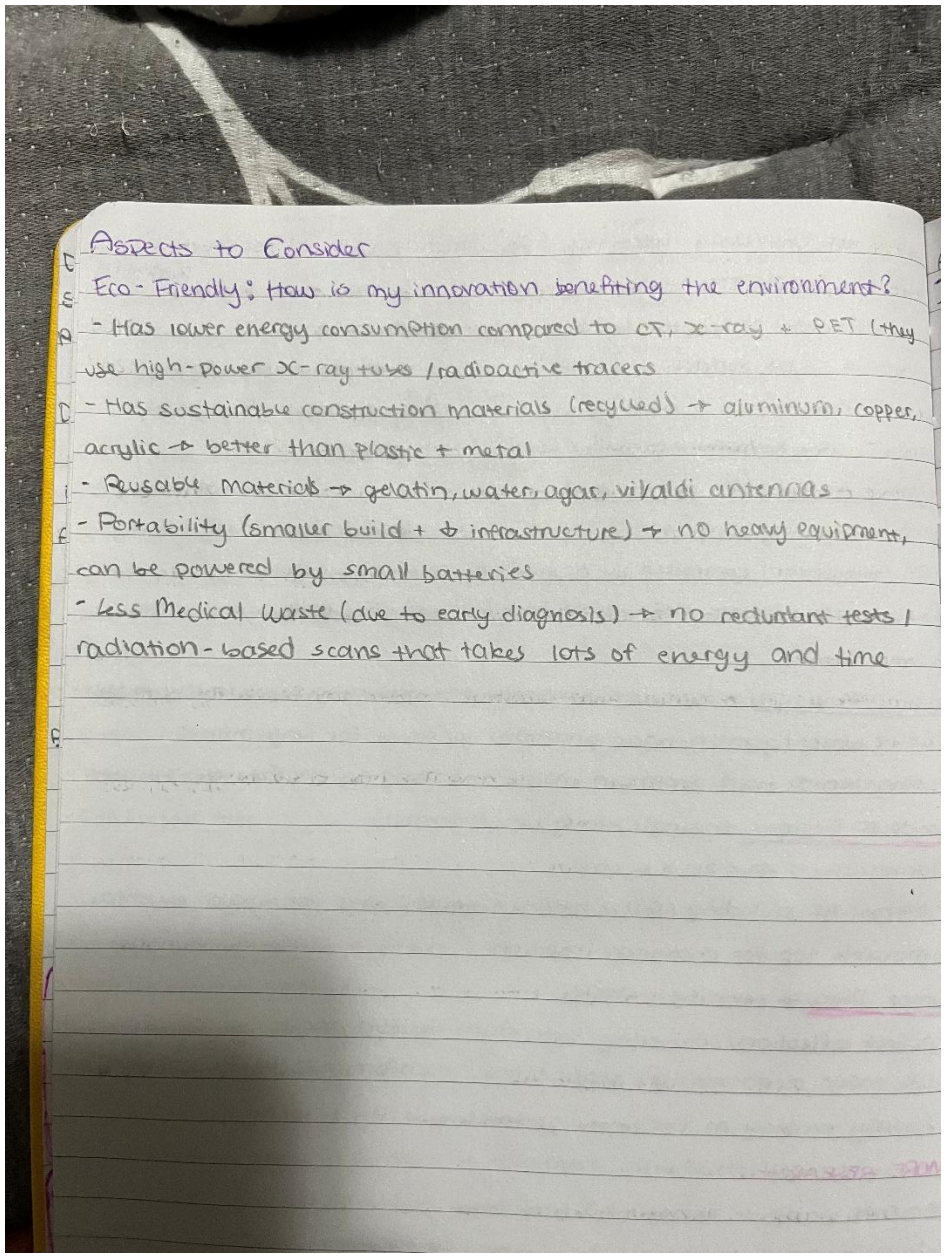
↳ Experiment's results (used five samples - increases sugar concentration)

4	25 g	54,43
5	26 g	53,97
6	27 g	53,78
7	28 g	53,49
8	29 g	53,47

Sample | Sugar (g) | ε_r @ 1 GHz

- keep phantom at stable room temp (20-22°C) during experiments to ensure consistent ε_r → simulate fever/inflammation (higher body temp) by heating slightly and seeing how system behaves → use VNA to measure ε_r

- Antennas should be a range (2-6 GHz) because it lets my system extract both depth and detail → 2 GHz penetrates deeper (blurry), 6 GHz gives sharp details (↓ penetration)



Aspects to Consider

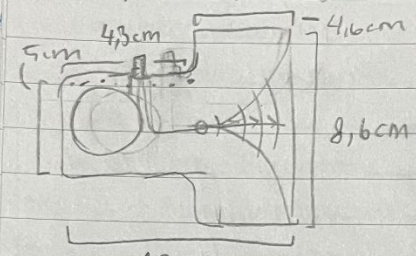
Eco-Friendly: How is my innovation benefiting the environment?

- Has lower energy consumption compared to CT, x-ray & PET (they use high-power x-ray tubes / radioactive tracers
- Has sustainable construction materials (recycled) → aluminum, copper, acrylic → better than plastic & metal
- Reusable materials → gelatin, water, agar, vivaldi antennas
- Portability (smaller build & ↓ infrastructure) → no heavy equipment, can be powered by small batteries
- Less Medical Waste (due to early diagnosis) → no redundant tests / radiation-based scans that takes lots of energy and time

Antenna Assessment

Labelled all antennas — 1 to 6

Condition = Good (7/10)



- 0,1 cm thick
- The central connector (+ lines) covers 2,3 to 3,1 cm of the line when cropped vertically

4, 5, 6 have the circle as a whole, while 1, 2, 3 have circle covered in

Recap

- 6 antennas
  - ↳ No need for substrate, the antennas already finished + packaged
- 7 SMA cables → found the perfect SMA cables after hours of searching!
  - ↳ 1 for each antenna, 1 for VNA
  - ↳ Bingfu SMA Female Bulkhead Mount to SMA male RG316 Antenna Extension cable, 2 feet 60 cm
- 2 RF Switches
  - ↳ Tx / Rx

Updated PULMO AI audio

The acoustic subsystem of PULMO AI achieves 86.4% accuracy (95% CI: 81.5–90.3) in classifying five pulmonary conditions from breath sounds, w/ particularly strong performance in COPD detection (94% F1-score). Statistical analysis confirms significance over baseline methods ($p < 0.000001$). This provides a robust foundation for integration w/ microwave imaging in a multi-modal diagnostic device

→ Dshnal will no longer work bc piezo pics up on heart sounds, not directly lungs. Using stethoscope w/ BOYA BY-M1S

- I had some error w/ PyQt5 in my code → self.image_label.setAlignment((Qt.AlignCenter) instead of Qt.Gui.Qt.AlignCenter
- All of a sudden, my Pi IP changed
- I'm having issues booting the app because the app starts but crashes at boot, but I try to run it again and it causes error
- The reason is because I have a cron file (the crash) that was set well (doesn't incorporate venv and system) while my systemd file does → causes conflict
- I fixed it all but its still not working (maybe I should just use streamlit) X → can't do it bc it uses chrome; not an app
- Error is bc of the use of old cv2.VideoCapture bc I keep getting → pipeline hasn't been created, failed to allocate memory, can't open camera by index

LAST STEP AND STILL ERROR WHY WHY WHY WHY WHY

- Changed IP= 192.168.1.248 → 105
- ! Camera isn't working because I'm using picamera / libcamera → which apparently isn't compatible with OpenCV or smth AAAAAAAAAGHHHH!
- I noticed the app runs but crashes saying: "the qt.qpa.xcb: could not load qt platform even though it is found. This application failed to start because no qt platform plugin could be initialized" - it then gives a list of platforms including 'eglfs'
  - ↳ OpenCV's Qt plugins are still interfering so I am trying to set my env variable before activating the virtual env → export QT_QPA_PLATFORM=eglfs
- It loaded but it said QWidget::paintEngine: Should not be called on unknown
- Tried to test my model separately, but millions of crash logs spammed bc of a mismatch (my env is Tensorflow 2.15.1) → its too new → I am changing to 2.12.0 → now I want to convert my file (model) to tf lite bc it told me but it gave the Functional error, so now I'm researching the right → to convert model, I need to redo my whole model and instead of Colab, I'm using Kaggle Notebooks because → longer runtime, better GPU, no sudden disconnects ↳ took forever to load dataset on Colab + train only for internet to undo progress...
  - ↳ still failing: could not deserialize class 'Functional' because its parent module keras.src.models.functional cannot be imported → now I'm trying

↳ On Kaggle, I've uploaded my dataset and created my model called best_model.keras using EfficientNetV2B0 → initially I trained it for 60 epochs (25 unfrozen layers, remaining slowly unfreezing layers based off of model performance that got average to 60%).

↳ Database = 12,888 images of 24 classes → class distribution was quite imbalanced (benign = 1093, squamous = 181) → 6:1

↳ So to mitigate, I did an automatic class weight calculation of: class_weights = compute_class_weight('balanced', classes, y_train) → so that the model doesn't over/underfit

↳ Next up, Stage 3 of continued training (epochs 61-120) because large datasets need several epochs (≥100)

From Epochs 1-60:

Final Training Acc = 60.91% Final Val Acc = 52.74%

Total Classes = 24 Best Val Acc = 52.93%

So the classes with the least amt had more weights

→ squamous cell = 3.0 (highest) → benign = 0.5 (smallest) and the classes follow that order

I did the Step 3 but training didn't go past 65% so I'm redoing it w/ a higher LR (Adam) and other data augs (RandomTranslation → randomly shifts images V/H = regularization | GaussianNoise → adds random value to input to help model learn features, less sensitive to small variations/details = counter overfitting)

(*If I want to write a quick script to run, add '.py' to the end of the sc name → nano led_test.py

→ from gpiozero import LED
from time import sleep
# Just GPIO 17 for now
led_pins = [17, 27, 22]
GPIO.setmode(GPIO.BCM) To run,
for pin in led_pins: ← python3 led_test.py
GPIO.setup(pin, GPIO.OUT)
# number of blinks
num_blinks = 10
for i in range(num_blinks):
for pin in led_pins:
GPIO.output(pin, GPIO.HIGH) # turn on
time.sleep(0.5)
for pin in led_pins:
GPIO.output(pin, GPIO.LOW) # turn off
time.sleep(0.5)
GPIO.cleanup()

* To check pi name, type whoami in console (SSH)
* Can't find the location of my app, run: find /home/havil -name "nom -app.py" → Learned my path is /home/havil/noma_ai/noma_app.py
* #Everyloves2ilerchic123
* On Pi (via Wareshare), create venv after heading to directory (but download the option to create env) then download my packages excep
* PyQt5 (have to install its tools independantly) → sudo apt install
python3-pyqt5 (system-wide)

To make it run automatically, I run
→ crontab -l (that's an 'l' lowercase) → should say none
→ crontab -e (opens editor) → choose '1' for nano
→ @reboot /usr/bin/python3 /home/havil/noma_ai/noma_app.py & (might sing it)
→ crontab -l (now it should say that line) DISPLAY=:0 (this represents the wave-share screen)

How to edit my app code via Wareshare! → use nano
nano /home/havil/noma_ai/noma_app.py
Since it is in the venv (only) and PyQt5 is in system-wide, how can I allow them to both be used → Google search (how to run packages both system-wide + inside a venv on a Pi 4?)
→ cd /home/havil/noma_ai
→ export PYTHONPATH="/usr/lib/python3/dist-packages/PyQt5:$PYTHONPATH" → I don't want all of Python bc system PyQt5 and the other packages in venv might have a version of numpy that isn't compatible X
→ python3 noma_app.py
Didn't work →
→ cd /home/havil/noma_ai
→ rm -rf venv (delete it)
→ python3 -m venv venv --system-site-packages
→ ls
→ source venv/bin/activate
→ ask it to show PyQt5 + tensorflow (python3 -m pip show —)
→ python3 -m PIP install tensorflow (bc it didn't exist)
→ pip install --upgrade --force-reinstall "tensorflow<2.16" "numpy<1.24"
← Why can't I install PyQt5 in my env? Everybody else is in the venv except for this one and it is having trouble downloading... ** → I don't need PyQt5 in an env

(5)

source ~/libcam-env/bin/activate
• /libcamera-hello

☆ It wasn't working because I was using the wrong cable (DSI vs CSI)
With the camera plugged in, run: vcgencmd get_camera
I still am having errors, so:
- Install dependencies
- Clone the libcamera source
- Configure the build (w/ meson) ← Lets me build raw from scratch I think, so there isn't any errors but there still error :—
- Compile the code (ninja -C build)
- Install apps (sudo ninja -C build install)
- Test (libcamera-hello -t 5000)

Still not working w/ libcamera, so I'm adjusting to Picamera2
- sudo apt install -y python3-picamera2 python3-tk sftt libcamera-apps
- my config file must have = start_x=1 and gpu_mem=128
- sudo apt update, sudo apt install -y python3-picamera2
- sudo apt remove python3-libcamera ← They are old
- sudo rm -rf /usr/lib/python3/dist-packages/libcamera
- sudo apt install --reinstall python3-picamera2

Reason libcamera didn't work is bc it says it's a transitional package (binary does not last or smth) ← dummy package
-sudo apt remove --purge -y (libcamera, python3-picamera2)
- sudo apt autoremove -y
- sudo apt update
-sudo apt install -y libcamera-apps rpicam-apps python3-picamera2

Now picam works! Yahoooooo!!!
- sudo nano /boot/firmware/config.txt → for camera
- add (dtoverlay=ov5647) at [all] section
-sudo reboot
-rpicam-hello --list-cameras
- for Pi 4, blue tab should face Ethernet/USB ports ☆ This may be done right
- silver contacts face camera PCB (back of camera)
To test camera (on Pi)
- rpicam-vid -t 0 -n -l | vlc --demux h264 -
- I updated my script but after doing it, you have to update the file on desktop (on normal powershell, no havil@skin:~$)
↳ scp "path/to/script" havil@IP Address:~/noma_ai/
-to check if file is there
↳ ls -l ~/noma_ai

LED SETUP!!!

Red LED → GPIO 17
Yellow LED → GPIO 27
Green LED → GPIO 22
Breadboard
Jumper wires

3 resistors (~~330~~ 150 Ω) → bc of 5V/3.3V
↳ I changed because I want light to be bright and its for all of them (1 R/LED)
☆ I only have 100 Ω and 200 Ω, so I'll go for 220 Ω for safety so my Pi does not explode

LED has anode (positive long leg) → GPIO pin
LED has cathode (negative) → ground (GND)
GPIO pin - Resistor - LED anode
LED cathode - GND
GPIO pin = positive
GND = negative
☆ Instead of connecting to power supply, connect to GPIO
- Worried by the wires because it's transparent

Diagram

At least for one for example

-reconnect to session via SSH:
- ssh (username)@(device address)
- ssh havil@192.168.1.177

How to get to folder of work (NOMA-AI) to put to Pi
- Pi via SSH runs on Linux, so cmds change
↳ scp -r "C:\Users\havil\Downloads\NOMA\NOMA-AI" havil@192.168.1.177:~/noma_ai → -r copies the folder; ~/noma_ai is the destination of the folder on the Pi ✱ Do this code outside of Pi SSH terminal
After run on Pi SSH
- cd ~/noma-ai, source venv/bin/activate
When installing PyQt5, install it from the system package manager instead of pip
↳ sudo apt install qtbase5-dev qtchooser qt5-qmake qtbase5-dev-tools ✱ this one configures for pip → pip install PyQt5

When done, transfer files to Pi
- scp files (seperated by space bar) havil@192.168.1.127:~/noma_ai/
- or do the entire folder

Waveshare Screen Setup
- git clone https://github.com/waveshare/LCD-show.git
- cd LCD-show
- run the drivers script for model ~~replacing~~ adding your screen's script → LCD~~###~~ LCD35-show
- chmod +x LCD35B-show
- sudo ./LCD35B-show

Now you need to let the app run on screen but make it automatic so it doesn't just time out
- cd to my project (noma_ai)

mkdir -p ~/.config/lxsession/LXDE-pi ← this create path when it don't exist
nano ~/.config/lxsession/LXDE-pi/autostart ← blank file in nano editor
@python3 /home/pi/noma-ai/noma-app.py ← in the editor
↳ runs automatically

save n' exit then reboot pi (sudo reboot)

if touch creen:
-sudo apt update → sudo apt install xinput-calibrator
-xinput-calibrator (if errors happen concerning install): → sudo apt --fix-broken install
↳ run this in desktop environment (LXDE)

How to get in LXDE
-startx ← but first do go to driver folder → cd ~/LCD-show
if the boot/config.txt doesn't work → sudo ./LCD35B-show
like after checking: lsmod | grep fbtft)
and that cmd says nothing, make sure this line is at the very end: dtoverlay=waveshare35b:rotate=0 (but if there is alr a cmd that has: dtoverlay=waveshare35b) ← remove that
✱ The result will only show on the Pi → correction: the sudo nano /boot/config.txt command will not show anything bc I have Waveshare DSI display and it doesn't use fbtft

→ update my config.txt
→ ls /dev/input/ to see if Pi recognizes the display (event0, event1)
→ xinput list | so xinput recognize too
→ sudo i2cdetect -y 1 | if touch doesn't appear
→ echo "i2c-dev" | sudo tee /etc/modules-load.d/i2c.conf
↳ creates config file to tell Pi to always load the i2c-dev module at boot

→ sudo modprobe i2c-dev | loads i2c-dev kernel in
→ ls /dev/i2c* | sees if the I²C is alive
→ for bus in 0 10 20 21 22; do
  echo "Scanning i2c-$bus"; ← to see which bus my touchscreen controller is on
  sudo i2cdetect -y $bus";
done

if the address 0x38 shows up, thats the capacitive touchscreen controller Waveshare uses

→ sudo apt update ↔ install packages
→ sudo apt install xserver-xorg-input-evdev xinput-calibrator ↔ touchscreen config
→ sudo nano /usr/share/x11/xorg.conf.d/40-touchscreen.conf
↳ inside write: Section "InputClass"
  Identifier "Waveshare Touchscreen"
  MatchProduct "FT6236"
  MatchDevicePath "/dev/input/event*"
  Driver "evdev"
EndSection

Now touchscreen works

Camera

sudo raspi-config X doesn't work because of my config setup (doesn't give option of camera

use → ls /dev/video* → instead so Pi can detect camera

Camera is connected if /dev/video0 or /dev/video1 shows

sudo apt update ↔ install camera tools
sudo apt install libcamera-apps -y
libcamera-hello
↳ if no work

dpkg -L libcamera-apps | grep bin (if nothing shows up)
sudo apt update
sudo apt install -y git cmake build-essential libboost-dev libdrm-dev libexpat1-dev libjpeg-dev libtiff-dev libpng-dev python3-dev python3-pip
git clone https://git.linuxtv.org/libcamera.git
cd camera
meson build ← builds libcamera from DA SOURCE for Pi
ninja -C build
sudo ninja -C build install
why do I need Lamire in my code config? (Lamire kamal?)
there were errors in install bc python3 doesn't have yaml, ply
libcamera needs Lamire Yamal and ply for its code
sudo apt update
sudo apt install -y python3-yaml python3-ply ← install
cd ~/libcamera
rm -rf build } Build
All these errors because of wrong cable!!!

I was having errors bc I was initializing the cam startup
bc I didn't do it in an env
sudo apt update (1)
sudo apt install python3-full python3-venv python3-pip
sudo apt install python3-yaml python3-ply python3-jinja2
(2)
cd ~
python3 -m venv libcam-env
source libcam-env/bin/activate
(3)
pip install PyYAML ply jinja2
(4) Build

Materials for NOMA AI

Raspberry Pi 4 (4 GB)
- Main computer/processor
- operating system, controls all → essentially the brain

Camera Module
- Real-time images
- Processed by Pi 4

Waveshare 5" Touchscreen
- shows live camera feed, results + UI

64GB micro SD Cards
- storage for Pi OS, ML model, application files, etc

INIU Powerbank (20 000 mAh)
- Portable power source for Pi

Green Acrylic Sheets
- External enclosure
- Gives a clean look

~$250

Gorilla Adhesive
- secures components
- non conductive when dried

Geeek Pi case
- Protects Raspberry Pi board
- Has built in fan for cooling

Micro HDMI to HDMI cable
- used for initial setup by connecting Raspberry to a monitor
- Lets me see Pi desktop before touchscreen is configured

Building NOMA AI

I'm using Google colab to train this time of a dataset of 7616 images in training

Since my life is rigged, I've had to repeatedly train new model from 22 classes to 24 because it hasn't been saving my model and I lose my data after 90 minutes → I implemented a checkpoint system that saves my model to my DRIVE after every 2 epochs

Since I'm using DenseNet121, I'm using a Functional API ~~over~~ instead of the Sequential API bc:
- Sequential API is best for simple models where layers are stacked in order (layers are one after another)
- Functional API is flexible and allows for complex architectures, like those w/ multiple inputs/outputs

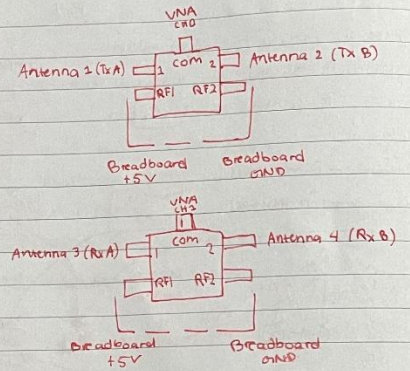
- I switched to MobileNetV2 bc the Raspberry Pi is 4GB, so I need a lightweight model

- Raspberry user + pswrd: hawil, 510270516
- hostname: ofsin.local
- ssh = pswrd authentification
- router authorization (user/pswrd) ≠ wifi details

Update Pi
Install Python + Libraries
Test Python
Transfer code
Run app!

* I'm using heat sinks instead of the case

RF Switch #1 (Tx)



Issue: I HAVE FEMALE to MALE connectors instead of MALE to MALE!!!
Need to research ways to allow for connection
↳ Bought SMA male-male adapters (10) from Amazon

My Explanations:
My RF switch online manual specifies the "RF" ports as DC / voltage control, so thats why I decide to connect them to 5V and GND, to alternate to be specific to the different antenna pairs →

CONTROL LOGIC		RF outputs	
Control Ports			
1	2	1	2
-V	0	On	Off
0	-V	Off	On

Octave

Other specifications highlighted by the ZFSWA-2-46 manual: (1) Max input RF power

+30 dBm except 100-500 MHz, +27 dBm, and DC-100 MHz +24 dBm, (2) control voltage (-10V) maximum (3) Video leakage / break through is defined as leakage of switching signal to RF output ports (4) OFF state at RF output is low impedance (5) All RF connections must be DC blocked / held at 0V DC → Btw, my 2 Mini-Circuits ZFSWA2-46 4.6 GHz SMA RF Switches are Single-Pole, Double-Throw (SPDT) switch = train track switch
- Common Port (Port 1) = main incoming track → to VNA (com)
- Port 2 & 3 = outgoing tracks → to antennas
- RF1 & 2 = lever that changes tracks → 5V / GND

★ RN switching is manual

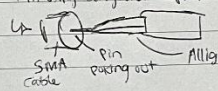
Paths
Tx1 → Rx2
Tx1 → Rx4
Tx1 → Rx1
Tx2 → Rx1

ET CETERA!
↳ switching will be controlled by GPIO

★ when implementing for Pi, I decided to:
- Connect Pi 5V pin to breadboard positive rail
- Connect Pi GND pin to negative rail
- Connect RF1 (V1/CTRL) to transistor's collector
- Connect RF2 (GND) to breadboard negative rail

Transistor Workings = GPIO LOW (0V) + Transistor OFF; No current flows so the RF switch CTRL pin sees 0V via internal pull-down, selecting Port 2 !
GPIO HIGH (3.3V) → Transistor turns ON; connects the CTRL pin to 5V, selecting Port 3 →

I'm using alligator clips to connect to breadboard
↳



★ 2N2222 - NPN Bipolar junction transistor → switching applications
1 = Collector 2 = Base 3 = Emitter

Important Files = quick_test.py ★ Removed all has built-in OSL calibration (no script needed)
Scripts = switch_controller.py + vna_interface.py + array_scan.py + calibration.py

★ Eventually, this project will use electronic RF switching → allows static scanning across imaging w/o mechanical rotation = ↑ temporal resolution + ↓ mechanical error compared to motorized scanning systems of the Princeton students

Patient Journal PULMO AI

Front cover:

PULMO AI LUNG HEALTH PASSPORT

Name: _____ ID: [QR CODE to digital record]

Page 1: Understanding your screening (what PULMO AI is)

Page 2: Your scan results (fill by hand)

Page 3: Lung Health basics (educational)

Page 4: Personal Health ~~Log~~ Tracker

Page 5: Prevention + Lifestyle

Back cover: Resources + Emergency (Region specific)

Both Journals have been completed on Canva! ✓

PULMO AI RF Switch Integration Setup Second Internship! ✓

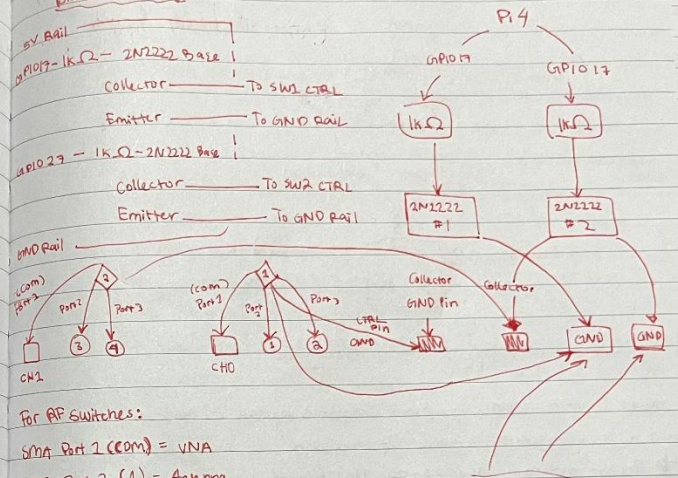
What I need:

- Breadboard (no soldering side so I can change + refine)
- 2x 2N2222 NPN Transistors → from my kit
- 2x 1kΩ Resistors → from my kit
- Jumper wires (to attach to Pi)
- Pi 4
- SMA Pingfu cables, antennas, VNA
- RF Switches

* Each switch governs over 2 Antennas (Tx, Rx) and attaches to one port
→ RF switch #1 goes to CH0 (Tx); RF switch #2 goes to CH1 (Rx)

◇ = RF ○ = Antenna □ = VNA

Breadboard Layout



For RF Switches:

SMA Port 1 (com) = VNA

SMA Port 2 (1) = Antenna

SMA Port 3 (2) = Antenna

SMA Port 4 (RF1) = DC control

SMA Port 5 (RF2) = DC control

↳ Apply voltage to switch between antenna 1 and 2

Breadboard 5V Rail → RPi 5V Pin

Breadboard GND Rail → RPi GND Pin

To Antenna:	Connect RF1:	Connect RF2:
Port 1 (Antenna A)	+5V	GND
Port 2 (Antenna B)	GND	+5V

Voltage = 1NLV Powerbank or 5V pin from Pi

Next page: more compatibility →

Calibrating Procedure 😅 – YouTube Video

Follows – OSLT (Open, Short, Load, Through) + my VNA gives detailed steps

1. Determine range: 2–3 GHz → 2 (start) 3 (stop) GHz
2. Begin Calibration (Two Port) → Cal
   a) OPEN on CH0 (open standard)
   b) SHORT on CH0 (short standard) in this order
   c) LOAD on CH0 (50 Ω terminator)
   d) OPEN on CH1 (Both 2 of S21 params)
   e) SHORT on CH1
   f) LOAD on CH1
   g) THROUGH (connect ports w/ cable)
   – Done, then SAVE

When I was done, I noticed the result was interesting. →  what?

But I think its bc VNA is doing continuous correction, so after CAL is hit, any change/updated is tracked. I have found 2 tests to see if CAL was good:

50 Ω Load Test: Get 50-Ω terminator and put on CH0 (look for S11) (near technically) (ideal)
→ Result = line @ bottom of screen @ below −40 dB across 2–3 GHz, and flat
→ I got swiggly lines hovering @ −6.52 dB.. so yeah

Through Connection: CH0 to CH1 via cable (look for S21)
→ Result = smooth + flat near 0 dB; ensure transmission is perfect
→ I got exactly that (0 to −0.1 dB)

So currently, my S21 calibration is good but not my S11

Issues could maybe be my load itself?
Wrong standard? Mistakenly using a SHORT?
I'm using multimeter to verify → I got 50–60 Ω range (most 51–54)
→ This means it is a proper 50 Ω terminator so maybe I miscalibrated?
→ I am calibrating again, middle to click LOAD only once transition from SHORT is smooth → still the same result = perfect S21, bad S11

I have to change the direction of microwave – primarily dependant on S21 params – which are perfect, so its OK
So I must scan S21 data and train my XGBoost model on S21 params

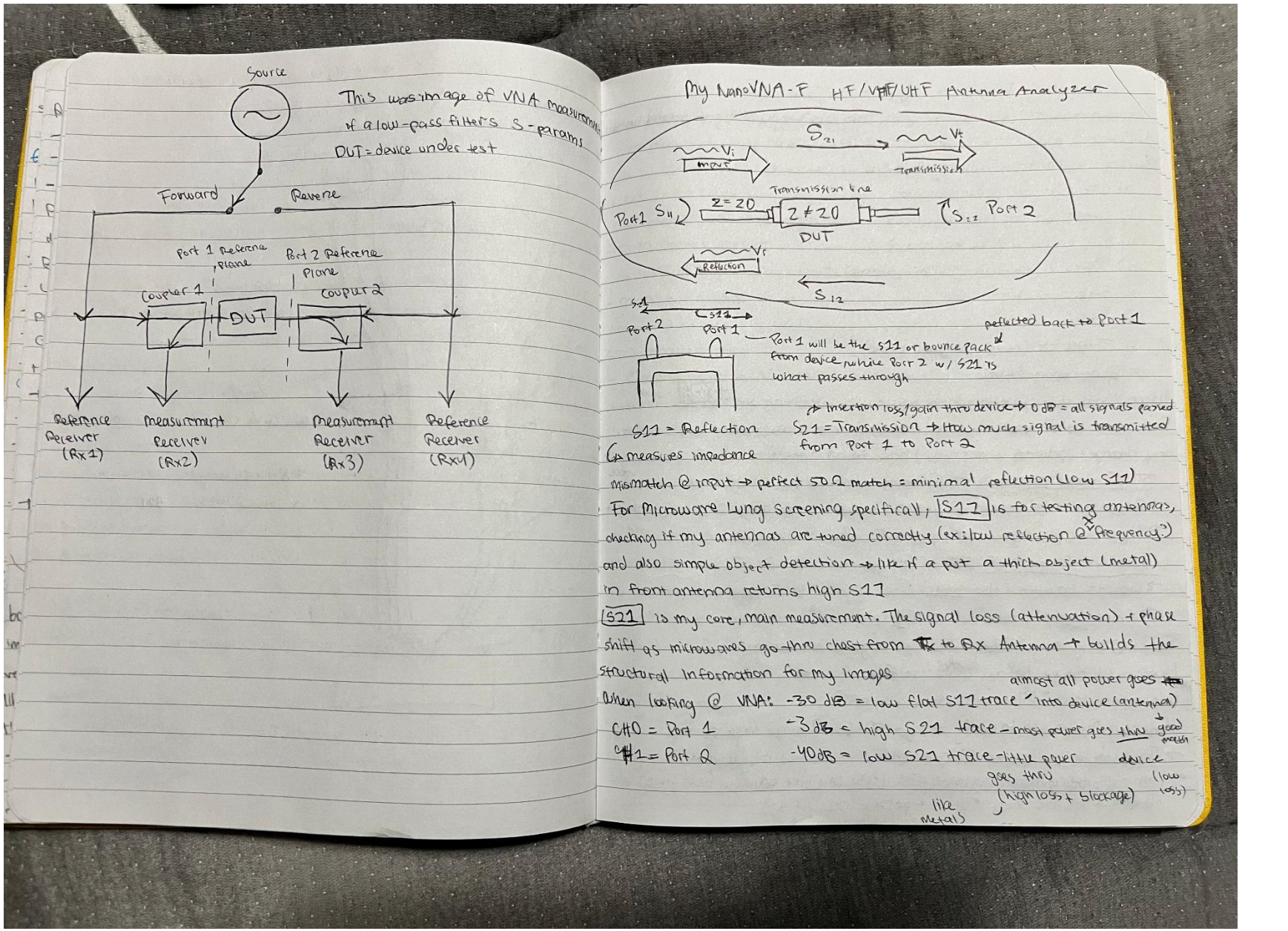
Next Steps (First Scanning Array):
- Two antennas on polyethylene foam, 10–15 cm apart facing each other
- Data capture script on Python to automate S21 sweep; VNA to Pi via USB → logs frequency, S21 magnitude (dB), and S21 phase
- Metal + Air Experiment → scan w/ nothing between (baseline_air.csv) and with metal object like sheet between (blocked_metal.csv) → Result = S21 should drop
- Create Phantom (gelatin) → Healthy vs Tumorous → check for differences in S21

I will record the results of each step on next page

What do I loose when I have bad S11?
- I think since S11 is reflecting, I have less structural data, and reduced sensitivity to the superficial aspect of scanning?
- Also, images aren't based on reflected energy anymore
- However, my previous research said the defining factor for tumorous vs healthy is dielectric + conductive properties, which have to do a lot w/ transmission? ←
- I cannot get perfect measurements of my antenna's input match/resonance frequency ✓ so I just have to use them as they are lol

SETUP Simple:

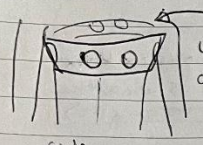
Had to connect VNA to PC bc it was glitching out the Pi → had to plug it in to powerbank so it would send data to comp instead of taking power


Source
This was image of VNA measuring
of a low-pass filter's S-params
DUT = device under test
Forward
Reverse
Port 1 Reference Plane
Port 2 Reference Plane
Coupler 1
DUT
Coupler 2
Reference Receiver (Rx1)
Measurment Receiver (Rx2)
Measurment Receiver (Rx3)
Reference Receiver (Rx4)
My NanoVNA-F HF/VHF/UHF Antenna Analyzer
S_{21}
V_i
input
V_t
transmission
Transmission line
Port 1 S_{11}
Z = 20
Z ≠ 20
DUT
S_{22} Port 2
V_r
reflection
S_{12}
S11
S11
Port 2
Port 1
Port 1 will be the S11 or bounce back from device while Port 2 w/ S21 is what passes through
reflected back to Port 1
S11 = Reflection
measures impedance
mismatch @ input → perfect 50Ω match = minimal reflection (low S11)
Insertion loss/gain thru device → 0 dB = all signals passed
S21 = Transmission → How much signal is transmitted from Port 1 to Port 2
For Microwave Lung Screening specifically, S11 is for testing antennas, checking if my antennas are tuned correctly (ex: low reflection @ frequency) and also simple object detection → like if a put a thick object (metal) in front antenna returns high S11
S21 is my core, main measurment. The signal loss (attenuation) + phase shift as microwaves go thru chest from Tx to Rx Antenna → builds the structural information for my images
When looking @ VNA:
CH0 = Port 1
CH1 = Port 2
-30 dB = low flat S11 trace
-3 dB = high S21 trace - most power goes thru
-40 dB = low S21 trace - little power goes thru (high loss + blockage)
almost all power goes into device (antenna)
good match
device (low loss)
like Metal)

Updated Microwave Setup

I just found an article working on the exact microwave imaging I'm doing!
→ Design of AI-driven microwave imaging for lung tumor monitoring - Singh al

They proposed a wearable belt (worn around upper torso) w/ antenna sensors embedded



What I will do since I only have 6 antennas
↳ very similar in structure to the breast cancer device → it encircles the area

1 each side, 2 front, 2 back

I only have 2 Rx switches so I will use time-multiplex? Noticed that I can't activate multiple antennas at a time as Rx and Tx

Model Pipeline

↓

XGBoost on raw S-params (The article mentions they experiment CNNs in classifying S-params)
↳ processes S21 transmission data collected from NanoVNA
↳ tumor detection / "yes or no" for abnormality presence

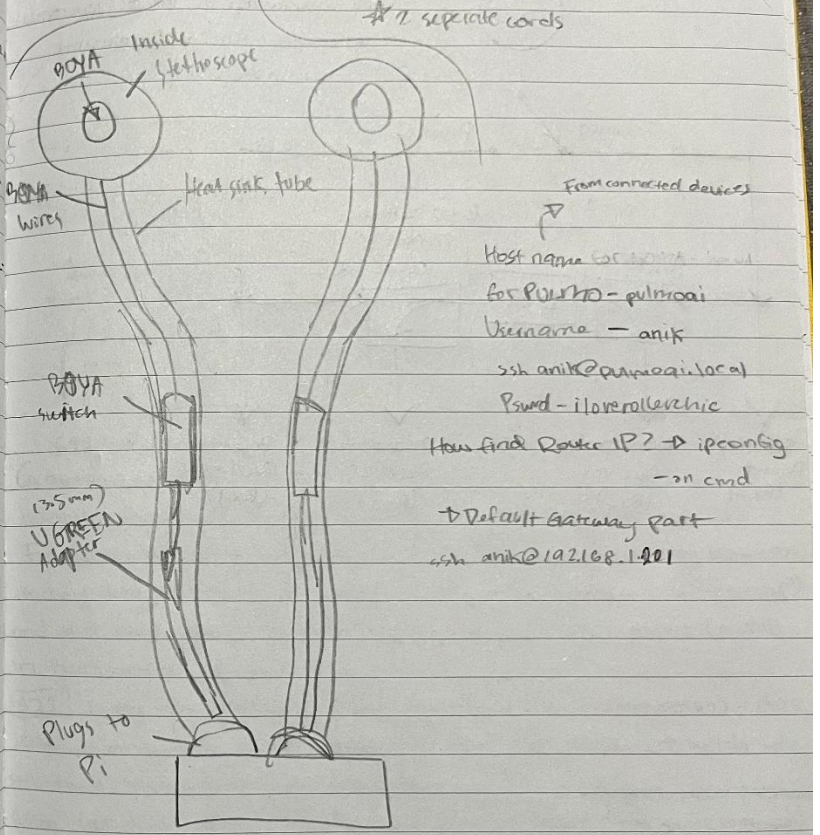
↓

(Delay-and-Sum) CNN for image reconstruction
↳ 2D microwave image creation

↓

CNN Quantification (regression model) reconstructed 2D image is fed in to analyze feature + estimate size of region

Design of Audio



From connected devices

Host name for Pi - pulmoai
Username — anik
ssh anik@pulmoai.local
Pswd - iloverollerchic

How find Router IP? → ipconfig
— in cmd
→ Default Gateway part
ssh anik@192.168.1.201