TEMPEST sniffing monitor

KAIR neural network to denoise image

Ondrej Mikle • ondrej.mikle@gmail.com • 16.4.2025

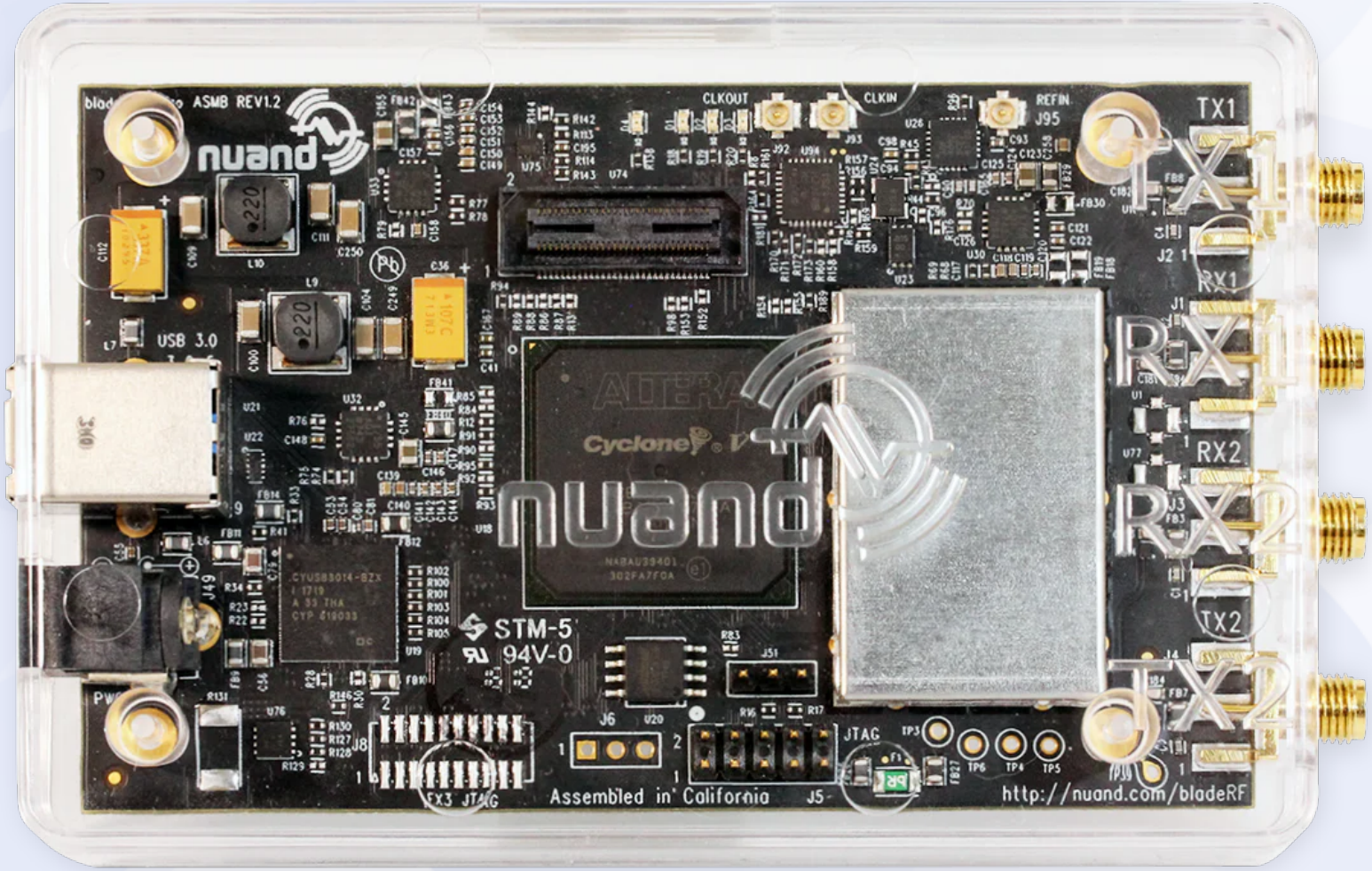
Electromagnetic emanations

- all time-varying voltage and current interfaces emit electromagnetic waves
  - chips, cables, all PCB traces act as transmitter antennas
  - signal can be -50 dBm or lower (less than 0.01 μW), even -130 dBm workable
  - current off-the-shelf SDRs can capture it, generally have >=90 dB dynamic range
  - e.g. BladeRF, USRP B210, H-field probe

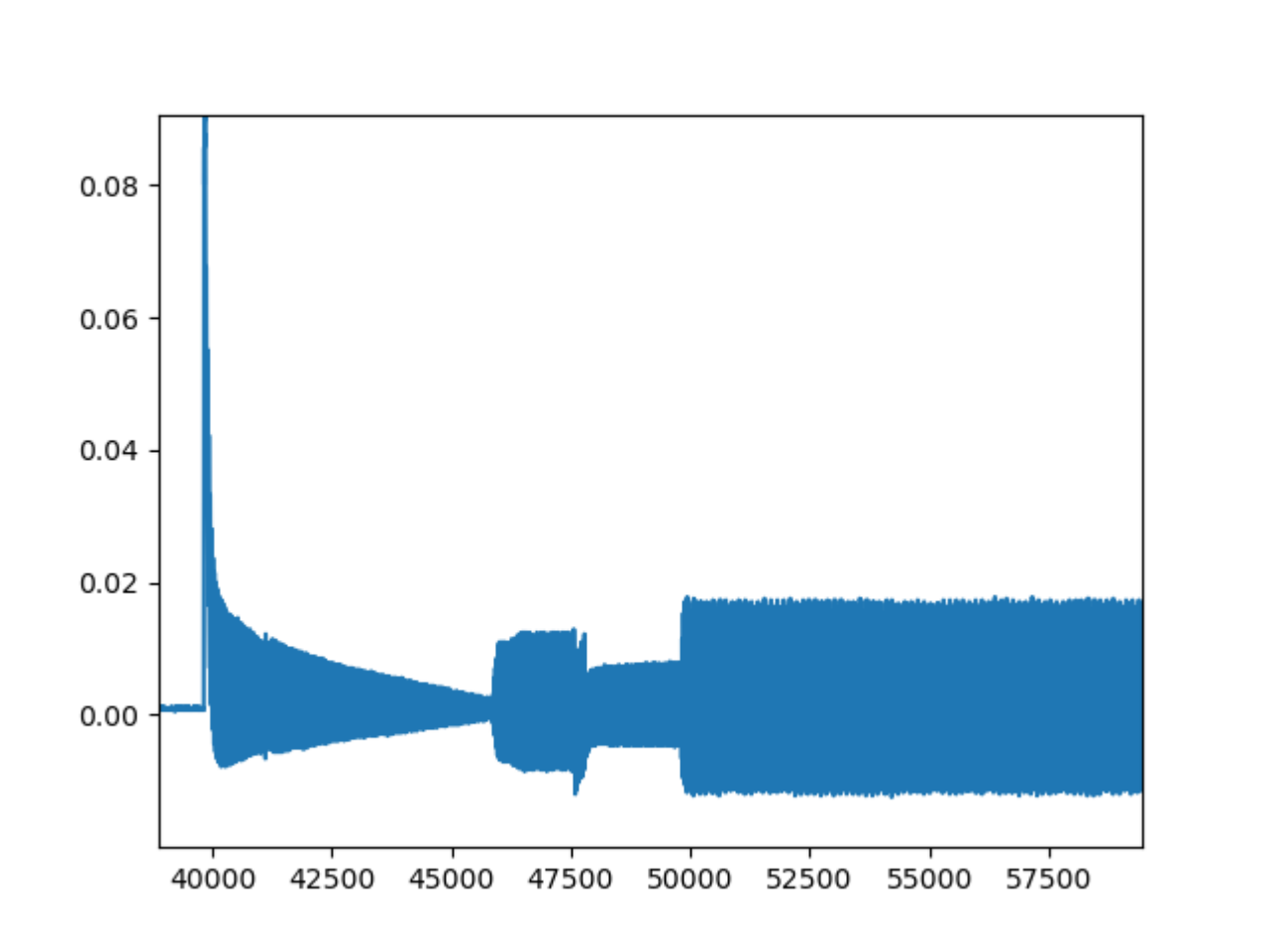
NESTOR - Vietnam war, ~1969

Had to be opened in the heat & humidity in Vietnam thus leaked plaintext voice in FM

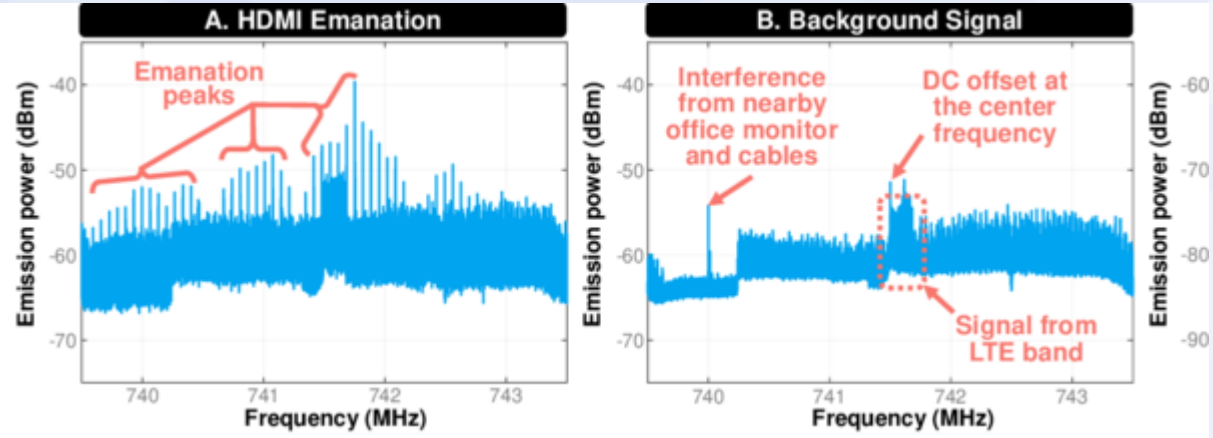
SDR used – bladeRF 2.0 xA4

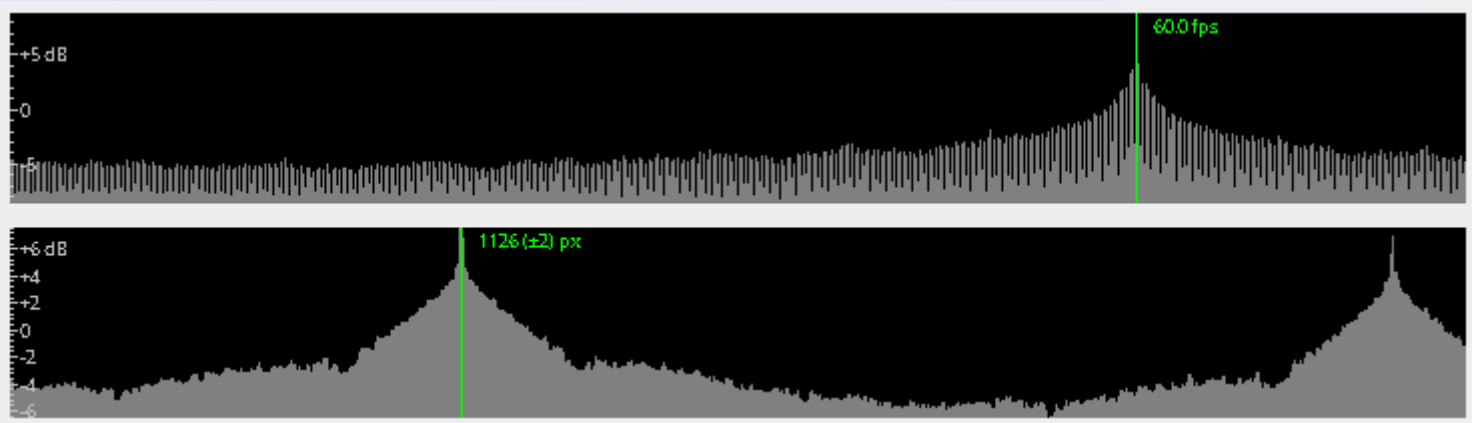


Sample EM trace of ARM chip



HDMI emanation & interference



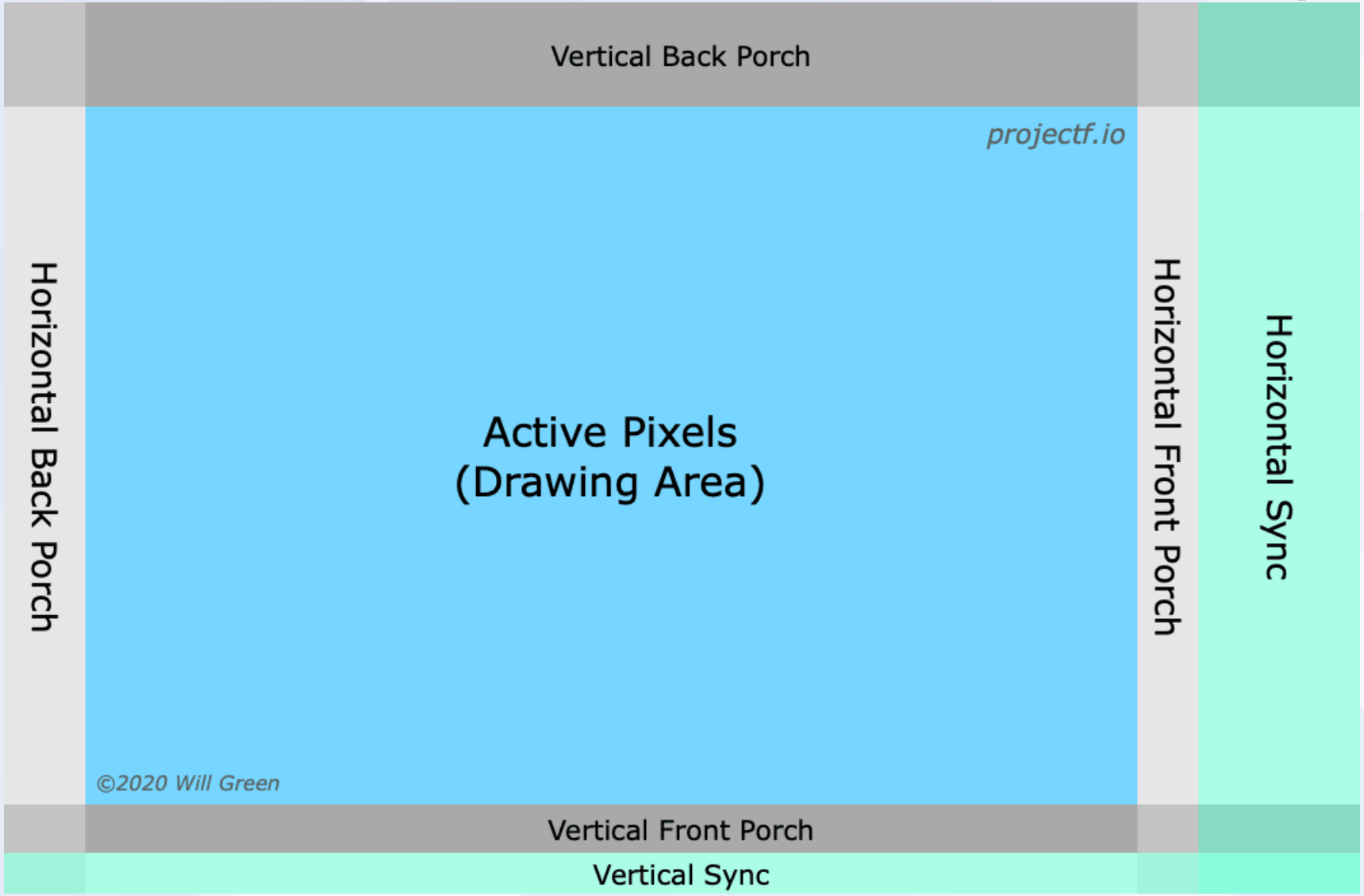


Why around 740 MHz? 742.5 is 5th harmonic frequency of 148.5 MHz pixel clock of one of 1920x1980, most commonly used videomodes.

VGA/HDMI/DisplayPort/USB-C

- VGA is trivial to sniff, analog
- HDMI & LVDS (LVDS is used in notebooks) have differential signalling, but still sniffeable
- DisplayPort is differential signalling, but with some PRNG to not make it so easy, can be partially filtered out, but with loss
- USB-C is network that encapsulates DisplayPort – hard to catch pixelclock

Front/back porch/pixelclock (1)



Front/back porch/pixelclock (2)

Resolution alone does not give horizontal/vertical or pixelclock timings, depends on display and other things

```
1920x1080 (0xa8) 148.500MHz +HSync +VSync
      h: width  1920 start 2008 end 2052 total 2200 skew    0 clock  67.50KHz
      v: height 1080 start 1084 end 1089 total 1125           clock  60.00Hz
1920x1080 (0xa9) 148.500MHz +HSync +VSync
      h: width  1920 start 2448 end 2492 total 2640 skew    0 clock  56.25KHz
      v: height 1080 start 1084 end 1089 total 1125           clock  50.00Hz
1920x1080 (0xaa) 148.352MHz +HSync +VSync
      h: width  1920 start 2008 end 2052 total 2200 skew    0 clock  67.43KHz
      v: height 1080 start 1084 end 1089 total 1125           clock  59.94Hz
1920x1080i (0xab) 74.250MHz +HSync +VSync Interlace
      h: width  1920 start 2008 end 2052 total 2200 skew    0 clock  33.75KHz
      v: height 1080 start 1084 end 1094 total 1125           clock  60.00Hz
1920x1080i (0xac) 74.250MHz +HSync +VSync Interlace
      h: width  1920 start 2448 end 2492 total 2640 skew    0 clock  28.12KHz
      v: height 1080 start 1084 end 1094 total 1125           clock  50.00Hz
1920x1080 (0xad) 74.250MHz +HSync +VSync
      h: width  1920 start 2558 end 2602 total 2750 skew    0 clock  27.00KHz
      v: height 1080 start 1084 end 1089 total 1125           clock  24.00Hz
1920x1080i (0xae) 74.176MHz +HSync +VSync Interlace
      h: width  1920 start 2008 end 2052 total 2200 skew    0 clock  33.72KHz
      v: height 1080 start 1084 end 1094 total 1125           clock  59.94Hz
1920x1080 (0xaf) 74.176MHz +HSync +VSync
      h: width  1920 start 2558 end 2602 total 2750 skew    0 clock  26.97KHz
      v: height 1080 start 1084 end 1089 total 1125           clock  23.98Hz
```

Front/back porch/pixelclock (3)

- pixelclock in MHz gives you timing for whole image, all these values neeed to be correct
- horizontal and vertical pixels are needed to make sense of the signal
- I found 5th harmonic frequency of pixelclock works best for me
  - e.g. pixelclock of 148.5 MHz has 3rd harmonic 445.5 MHz and **5th 742.5 MHz**

Antennas, filters & amplifiers

- better antennas and filters result in better images, i.e. better specify what is sniffed
- used omnidirectional antennas, also log-periodic (log-periodic is better)

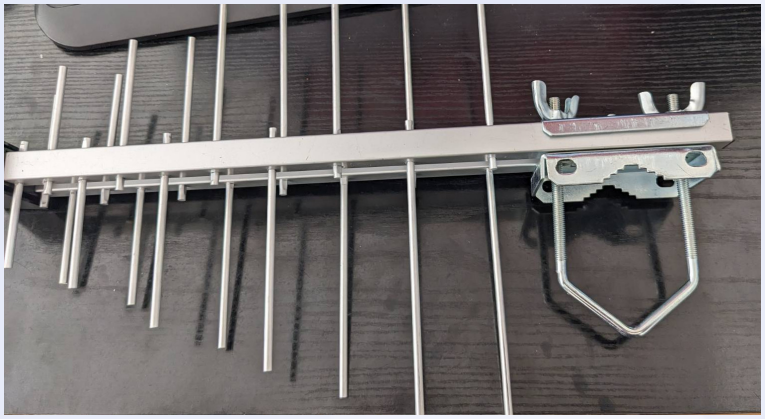
log-periodic antenna
(stock depleted now!)

other 3-band LTE for sniffing LTE data, not much monitors (omni, noisy)

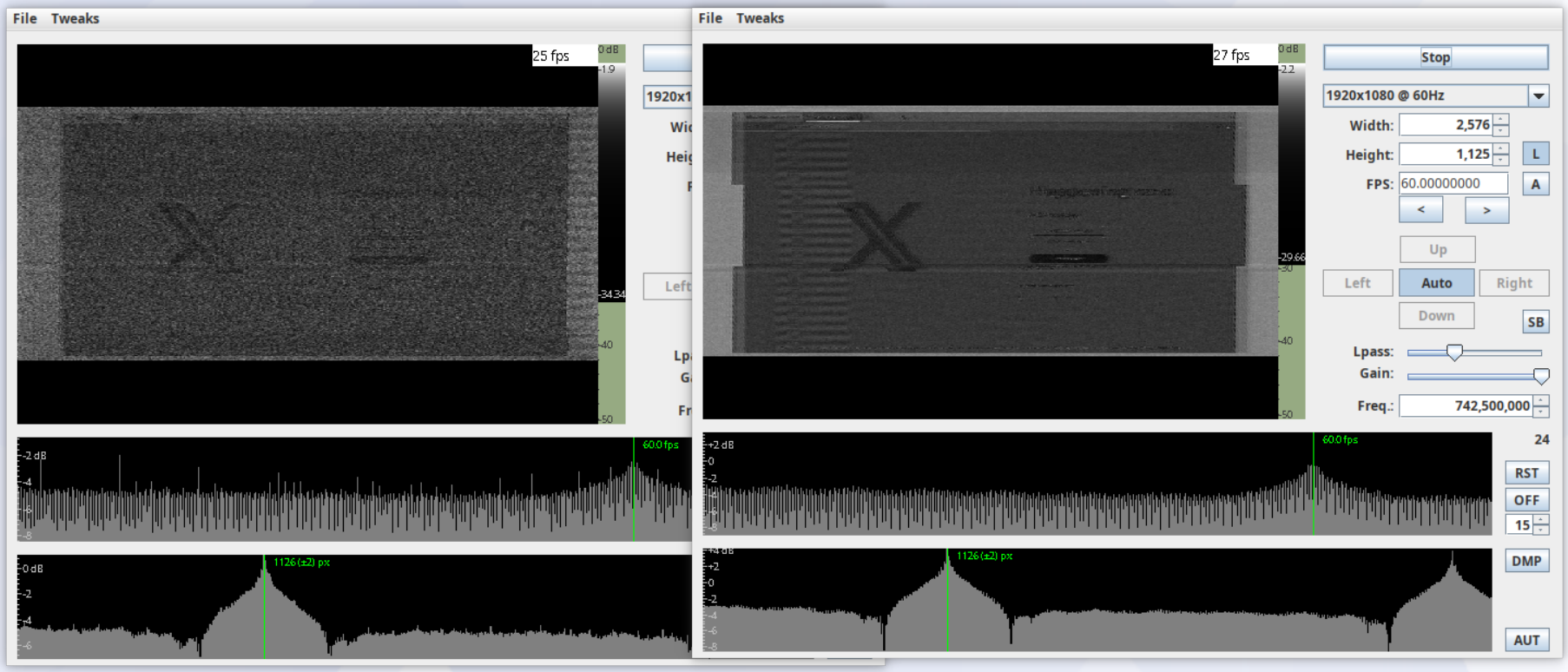
3-band LTE omnidirectional antennas
Taoglas has 82-page datasheet, best datasheet

Used cheap TV log-periodic

- it's tuned to 75 Ω impedance
- rest of SDR is tuned to 50 Ω impedance
- impedance mismatch causes VSWR 1.5:1 but still better than omni 3band Taoglas
- less noisy (50 Ω tuned w/SMA costs more)
- avoid signal filter
- could be lot better
- 800 CZK w/F-SMA



Bad day omni vs log-periodic

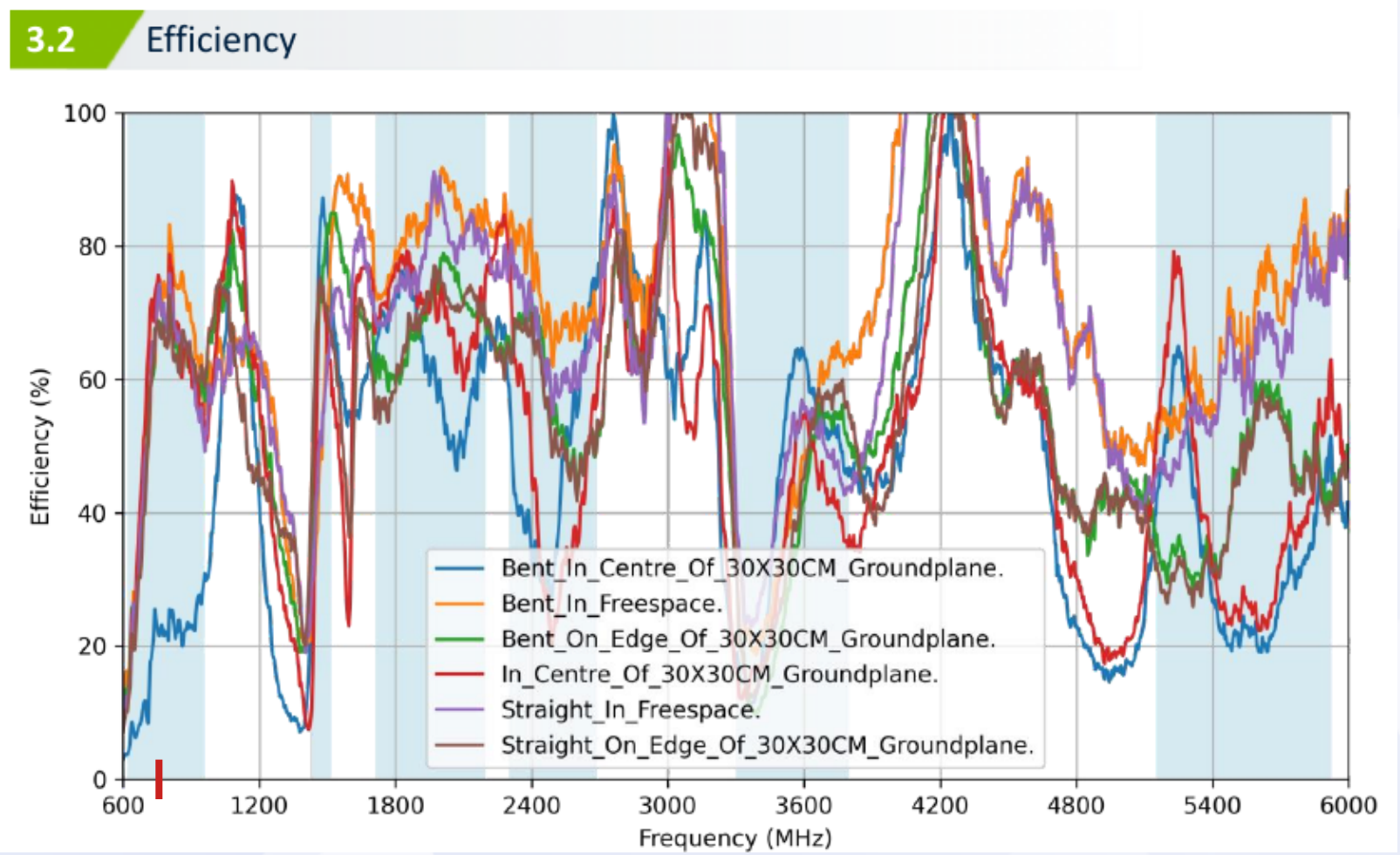


Bad placement for omni, omni antenna picks up more noise from every direction cca

Impedance-mismatched log-periodic does better than I expected

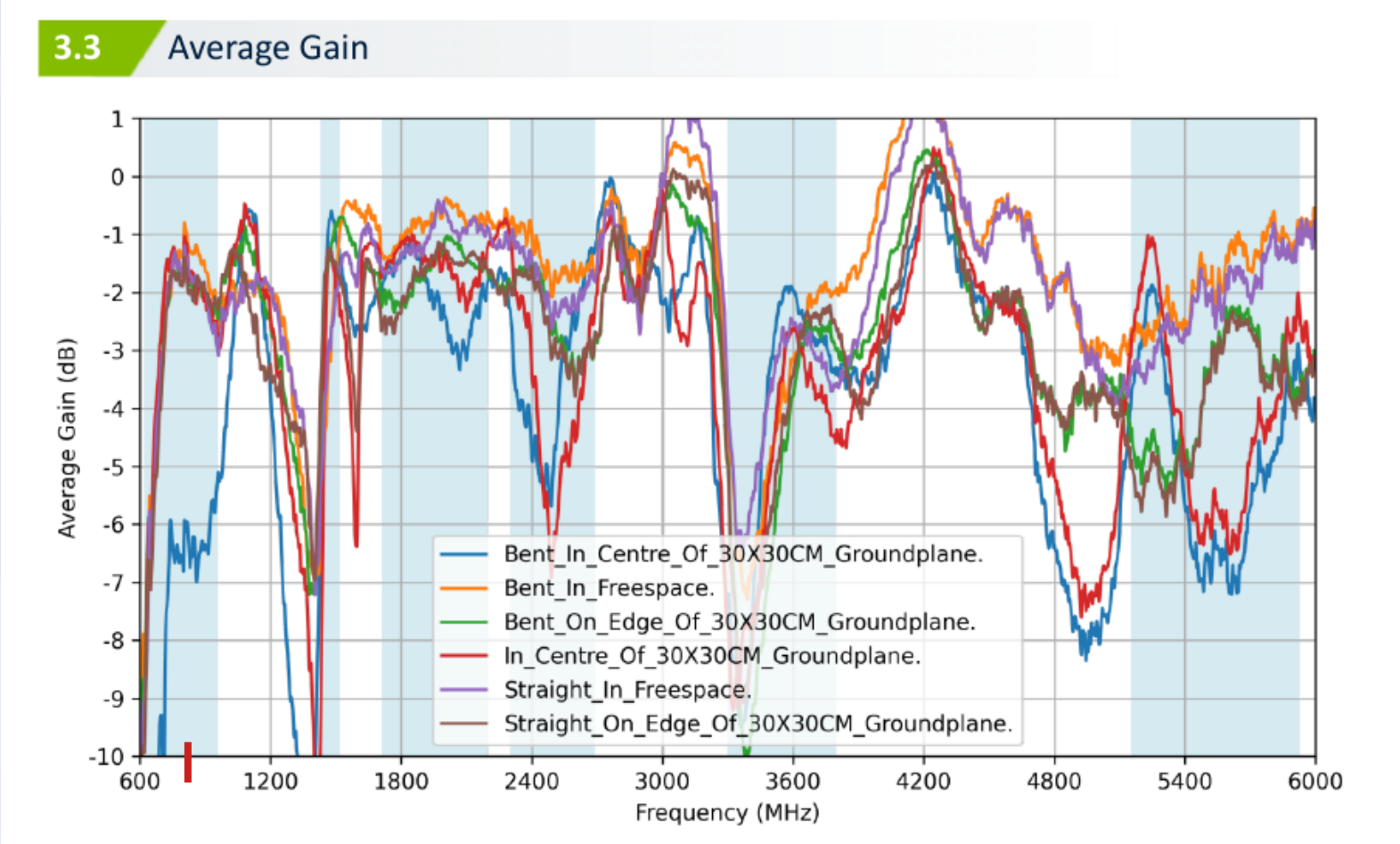
More serious setup

Short about antennas (1)



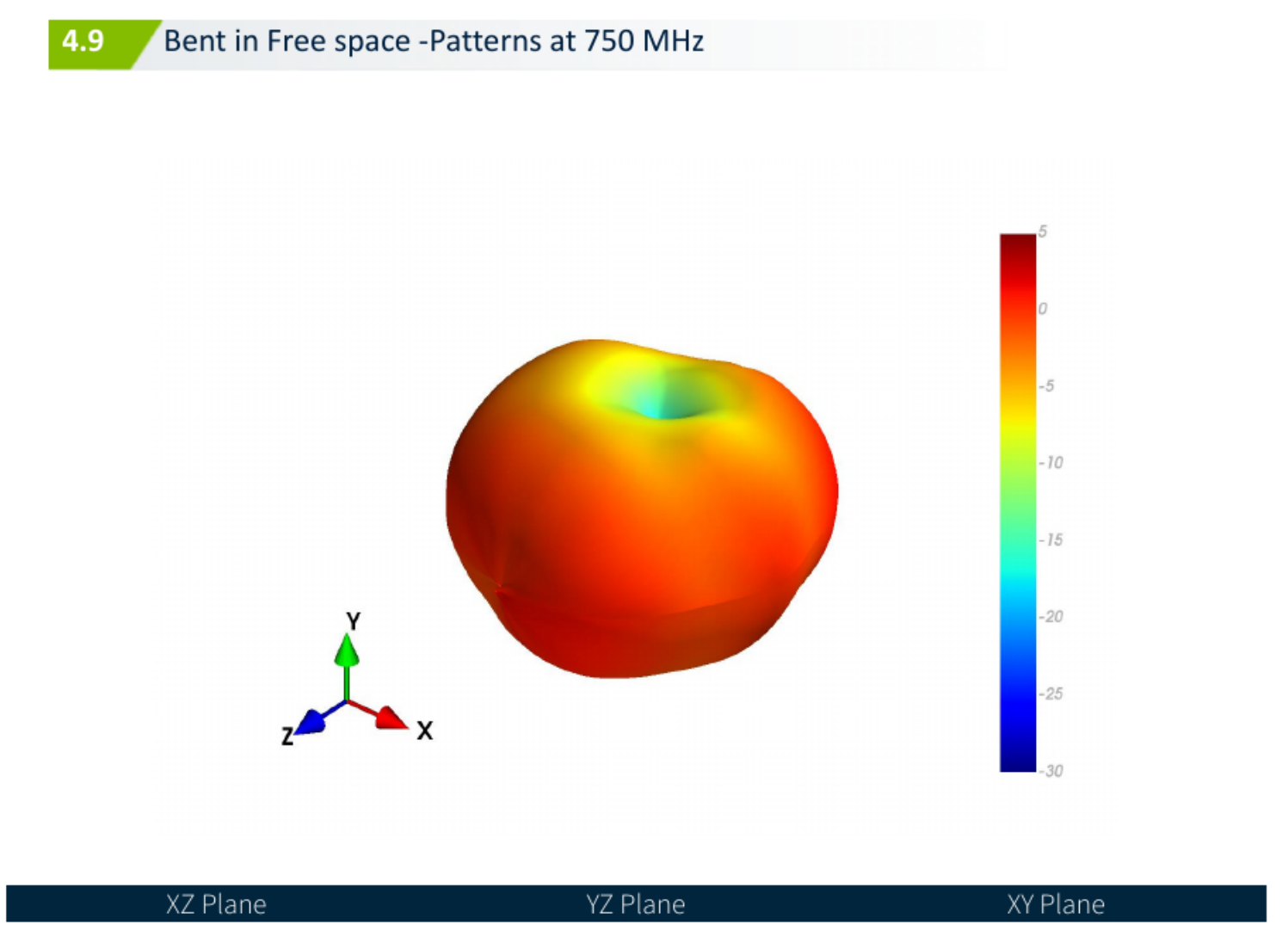
Efficiency and gain are correlated, based on directivity
We will be interested in those 742.5 MHz
This is Taoglas omidirectional TG.30.8113

Short about antennas (2)

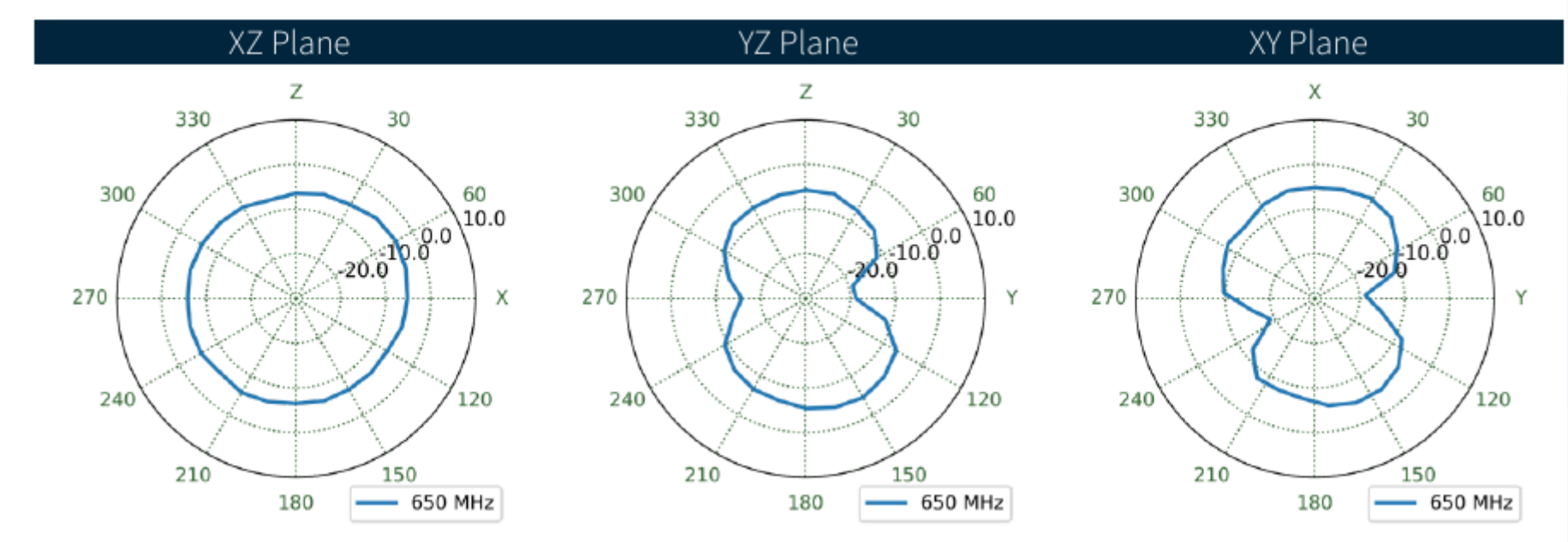


Taoglas Omnidirectional TG.30.8113
No other antenna had this good datasheet

Antenna radiation patterns (1)



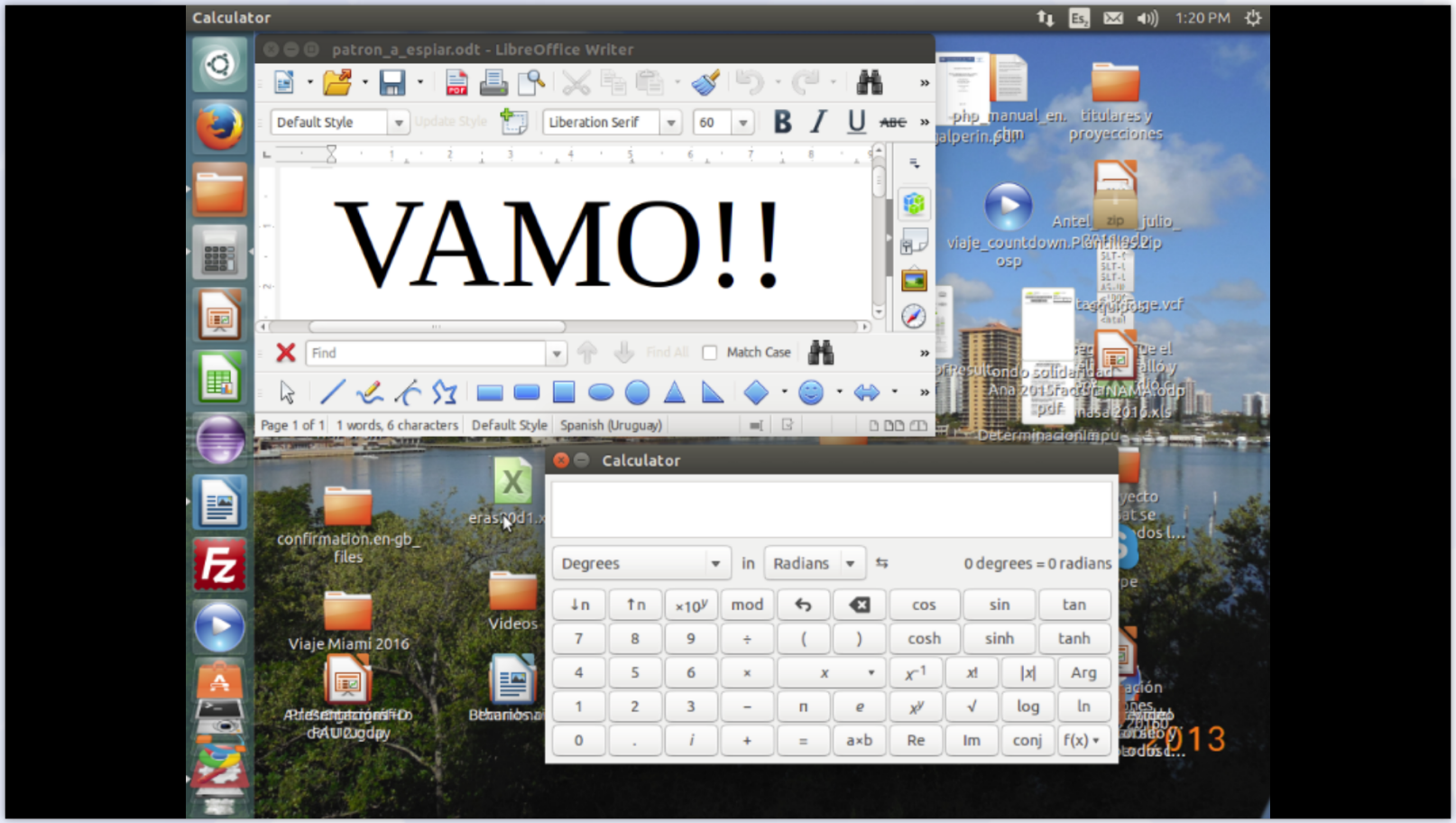
Antenna radiation patterns (2)



Software stacks

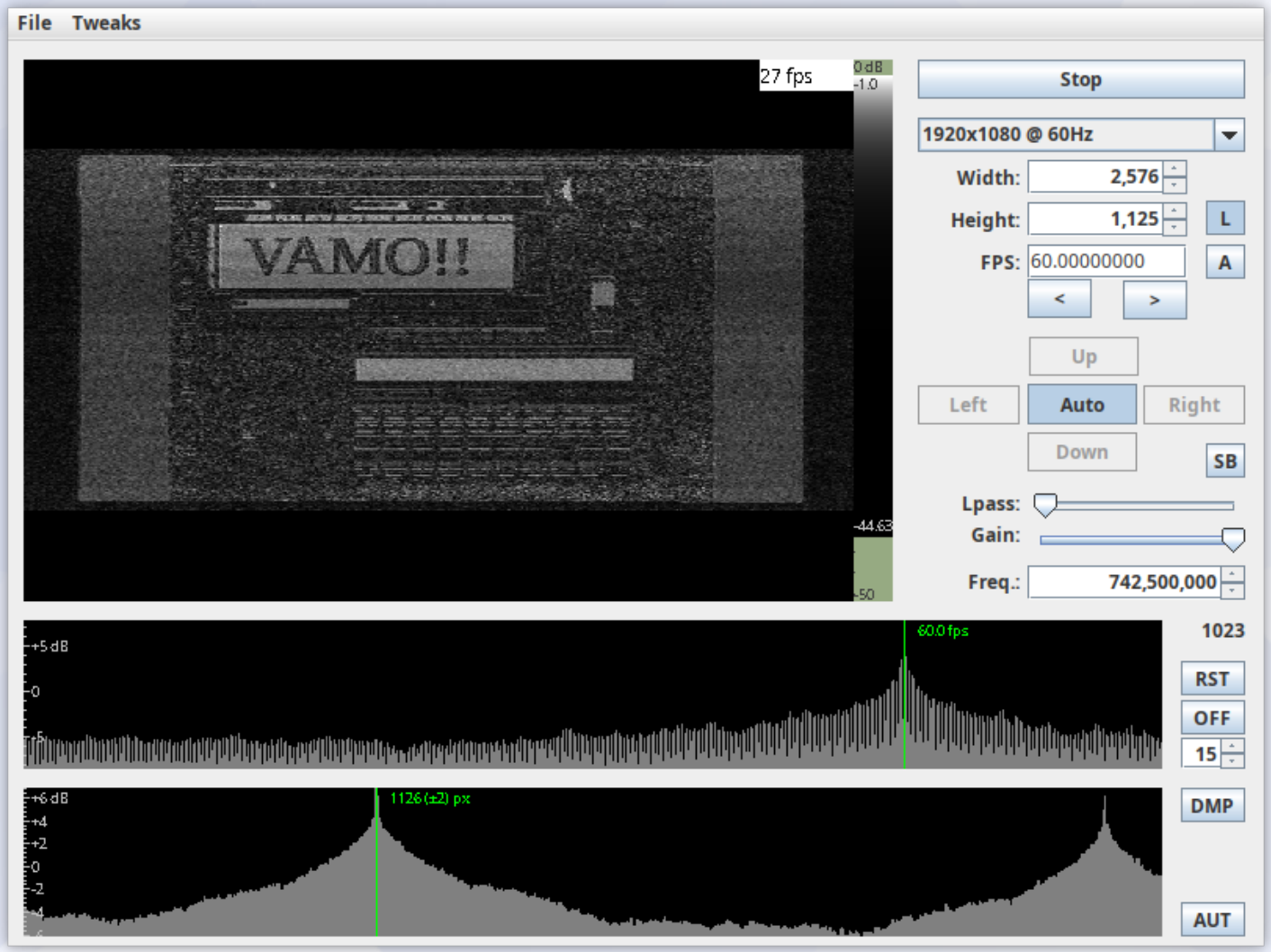
- kinda all
  - Java + native libs
  - C++
  - python + C/C++ native libs multiprocessing
    - gnuradio (terrible design)

First results, monitor original

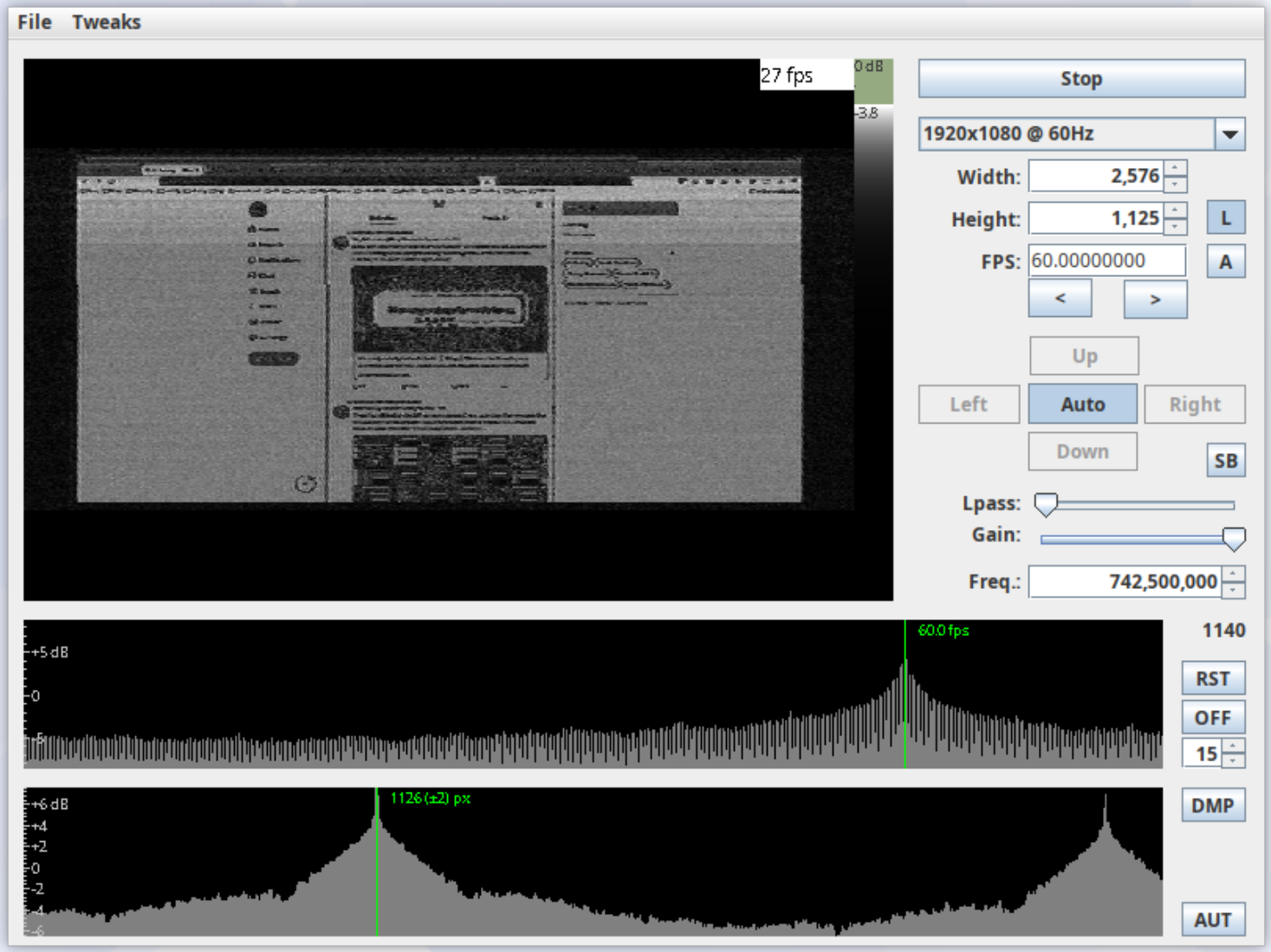


made a mistake to enlarge image via interpolation, thus losing edges
(losing edges <=> losing high frequencies, Fourier/cosine transform)

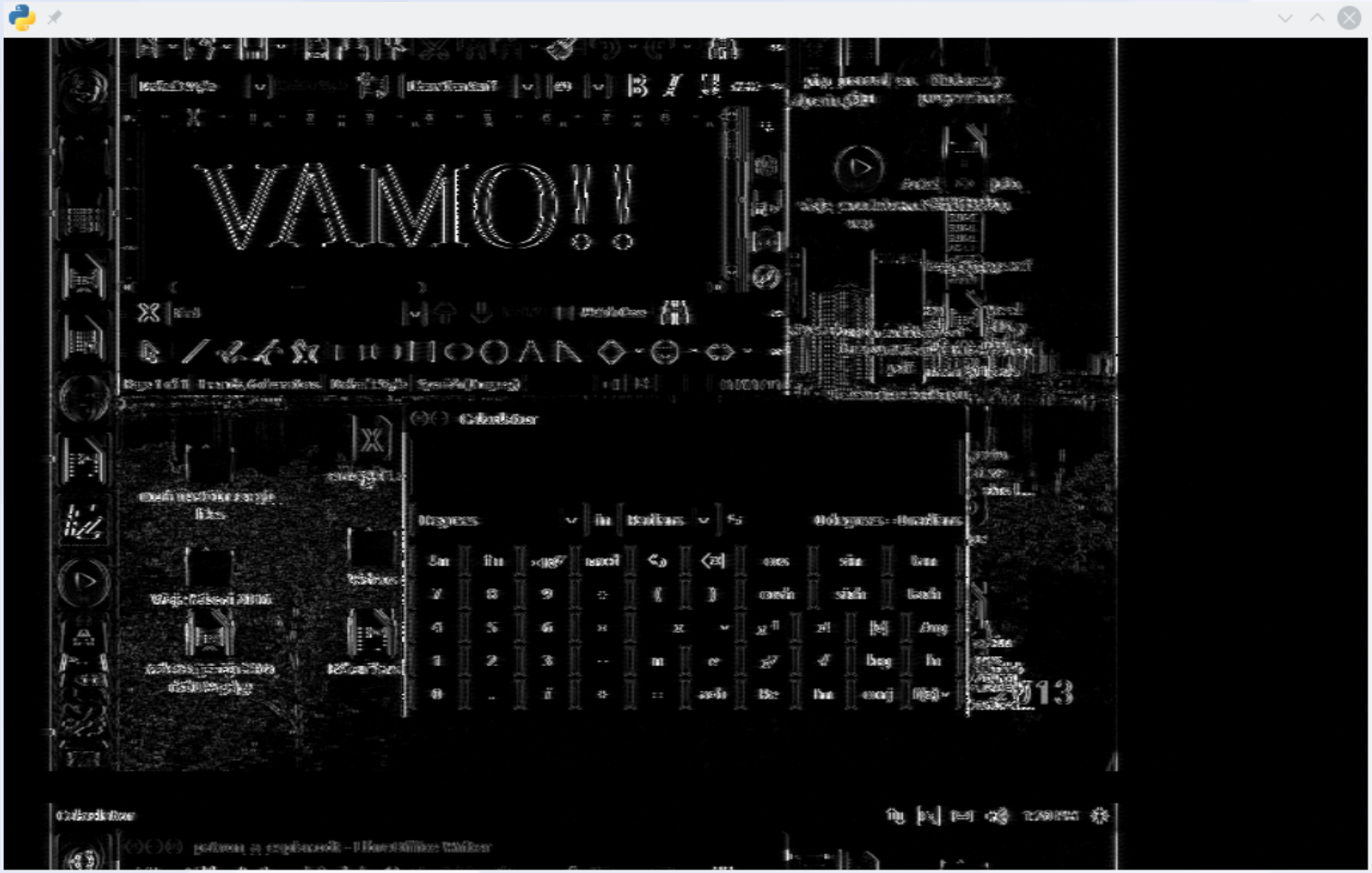
First results, TEMPEST image



High-contrast sharp image



Best you can hope with just SDR

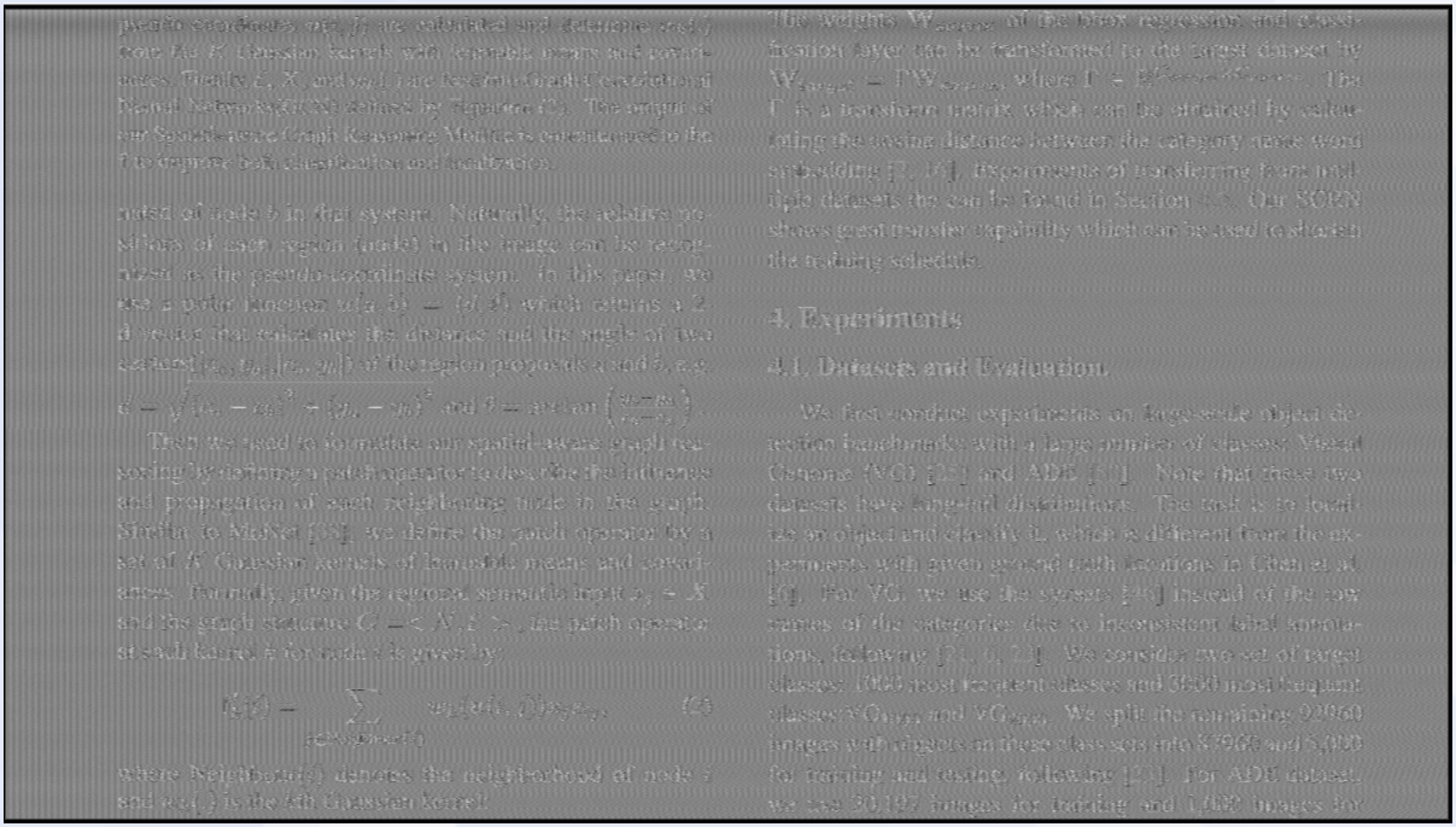


What next?

- enhance image somehow
- there is KAIR image restoration library you can use to train on some images
- training on web pages, PDFs and text is generally useful for attacker, as he wants to see some secret documents

KAIR denoising

Sniffed image before KAIR



After KAIR restoration

(note the artifacts)

pseudo coordinates $u(i,j)$ are calculated and determine $w_k(.)$ from the K Gaussian kernels with learnable means and covariances. Finally, $\mathcal{L}$, X, and $w_k(.)$ are feed into Graph Convolutional Neural Networks(GCN) defined by Equation (2). The output of our Spatial-aware Graph Reasoning Module is concatenated to the f to improve both classification and localization.

nated of node b in that system. Naturally, the relative positions of each region (node) in the image can be recognized as the pseudo-coordinate system. In this paper, we use a polar function $u(a,b) = (d,\theta)$ which returns a 2-d vector that calculates the distance and the angle of two centers($[c_a, y_a, |c_b, y_b|$) of the region proposals a and b, e.g. $d = \sqrt{(c_a - c_b)^2 + (y_a - y_b)^2}$ and $\theta = \arctan\left(\frac{y_b - y_a}{c_b - c_a}\right)$.

Then we need to formulate our spatial-aware graph reasoning by defining a patch operator to describe the influence and propagation of each neighboring node in the graph. Similar to MoNet [38], we define the patch operator by a set of K Gaussian kernels of learnable means and covariances. Formally, given the regional semantic input $x_j \in \boldsymbol{X}$ and the graph structure $G = < N, E >$, the patch operator at each kernel k for node i is given by:

$$f'_k(i) = \sum_{j \in \text{Neighbour}(i)} w_k(u(i,j)) x_j e_{ij}, \quad (2)$$

where Neighbour(i) denotes the neighborhood of node i and $w_k(.)$ is the kth Gaussian kernel.

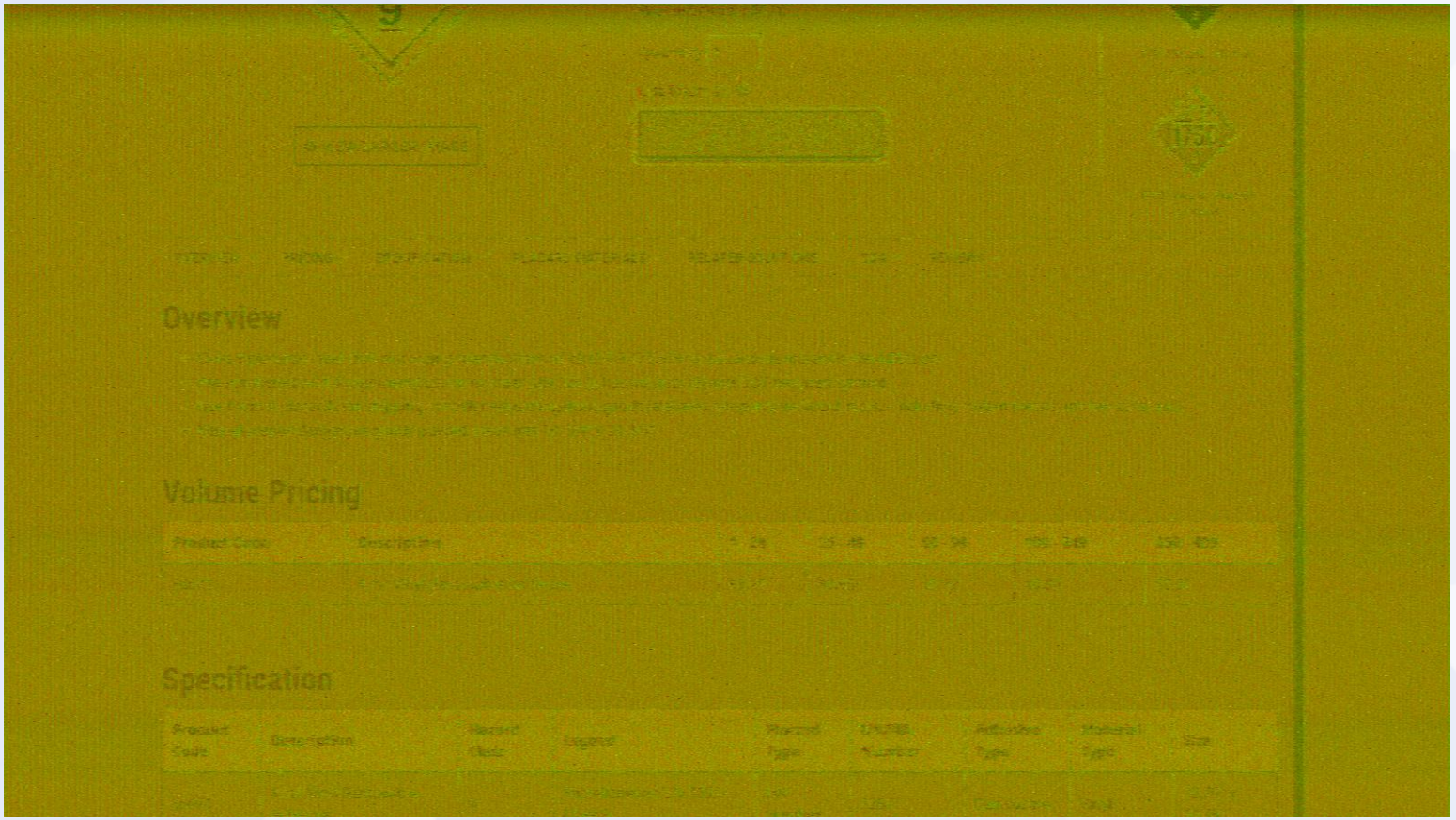
The weights W_{source} of the bbox regression and classification layer can be transformed to the target dataset by $W_{target} = \Gamma W_{source}$, where $\Gamma \in R^{C_{target} \times C_{source}}$. The Γ is a transform matrix which can be obtained by calculating the cosine distance between the category name word embedding [2, 16]. Experiments of transferring from multiple datasets the can be found in Section 4.5. Our SGRN shows great transfer capability which can be used to shorten the training schedule.

4. Experiments

4.1. Datasets and Evaluation.

We first conduct experiments on large-scale object detection benchmarks with a large number of classes: Visual Genome (VG) [25] and ADE [62]. Note that these two datasets have long-tail distributions. The task is to localize an object and classify it, which is different from the experiments with given ground truth locations in Chen et al. [6]. For VG, we use the synsets [46] instead of the raw names of the categories due to inconsistent label annotations, following [21, 6, 23]. We consider two set of target classes: 1000 most frequent classes and 3000 most frequent classes:VO_{1000} and VG_{3000}. We split the remaining 92960 images with objects on these class sets into 87960 and 5,000 for training and testing, following [23]. For ADE dataset, we use 20.197 images for training and 1,000 images for

Training KAIR, SDR snaphot



Training KAIR+deep tempest

- PITA to make it work
  - permanently pinched conda package versions (aaaaaaaaaaaah workaround)
  - build does not work (aaaaaaaah)
  - ... UHD driver for radio lags for gnuradio
  - crashes from time to time (gr-tempest)
  - need to take care to not overtrain it
  - placement of signal must be aligned

What could you add next

- OCR via Tesseract to get better text recognition

Training KAIR is tricky

- you need a model for every videomode pixelclock/hsize/vsize
  - when you see all the variables, it's too many resulting in many models to make
- cable measured matters
- not sure if it's reasonable to extrapolate data trained from one cable to another, rather not easily

Training – how?

- preparing training data sound easy
- but you need to make at least a script that changes monitor image, then takes 5 SDR samples per source image, deal with synthetic, ground-truths, simulations and real data
- avoid „overtraining“
- „write a script“ sounds way easier than it is to collect data for a model

Thanks

Ondrej Mikle • ondrej.mikle@gmail.com