🛡️ FCSC 2025 – _Le calme avant la TEMPEST

🎯 Objectif

Le but du challenge est de récupérer un flag caché dans un signal vidéo capté par attaque TEMPEST.
Une attaque TEMPEST consiste à intercepter les émissions électromagnétiques produites par un appareil électronique (ex : écran), afin de reconstruire l'information affichée à l'écran.

📁 Fichier fourni

- `le-calme-avant-la-tempest.bin`
  → Fichier brut contenant un flux de données audio/vidéo représenté sous forme :
  - d'**entiers 16 bits signés**
  - avec une **fréquence d'échantillonnage de 20 MHz**

🔍 Étapes de résolution

1️⃣ Tentative naïve avec résolution classique – trop de lignes

Première hypothèse : l'écran suit une résolution standard (comme 1280x1024).
On tente une visualisation brute en supposant une largeur de 1280 pixels.

```
import numpy as np
from PIL import Image

data = np.fromfile("le-calme-avant-la-tempest.bin", dtype=np.int16)

width = 1280
height = len(data) // width
image = data[:width * height].reshape((height, width))
image = ((image - image.min()) / (np.ptp(image)) * 255).astype(np.uint8)

Image.fromarray(image).save("try_1280.png")
```

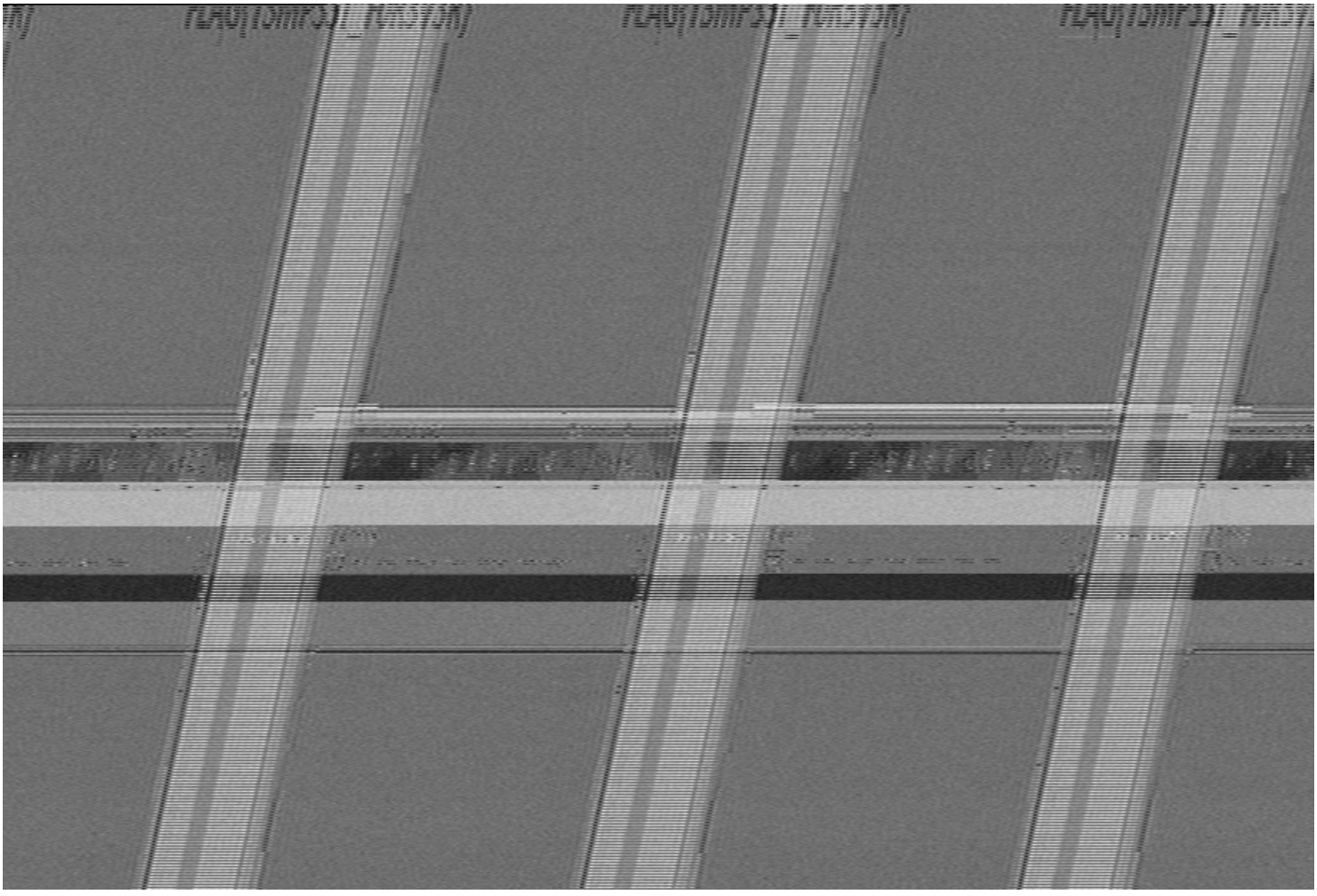
🔴 **Résultat :** image étirée verticalement, le signal est mal aligné.
→ Cela signifie que la **largeur supposée est incorrecte**.

2️⃣ Recherche de la bonne largeur en scannant plusieurs valeurs

On réduit le nombre de lignes affichées (200) pour aller plus vite, et on scanne des largeurs comprises entre 885 et 895.

```python
import numpy as np
from PIL import Image
import os

data = np.fromfile("le-calme-avant-la-tempest.bin", dtype=np.int16)
os.makedirs("cropped_heads", exist_ok=True)

for width in range(885, 896):  # Test dans une plage serrée
    height = len(data) // width
    if height < 200:
        continue
    image = data[:width * height].reshape((height, width))
    image = ((image - image.min()) / (np.ptp(image)) * 255).astype(np.uint8)
    Image.fromarray(image[:200, :]).save(f"cropped_heads/head_{width}.png")
```

✅ **Largeur 890** = alignement parfait du signal vidéo.
On voit apparaître un texte clair en haut de l'image, mais il est **tronqué**.



3️⃣ Extraction finale du flag complet avec 1500 lignes

On garde la largeur 890 (précise à l'échantillon près) et on augmente la portion verticale à 1500 lignes pour capturer le texte entier.

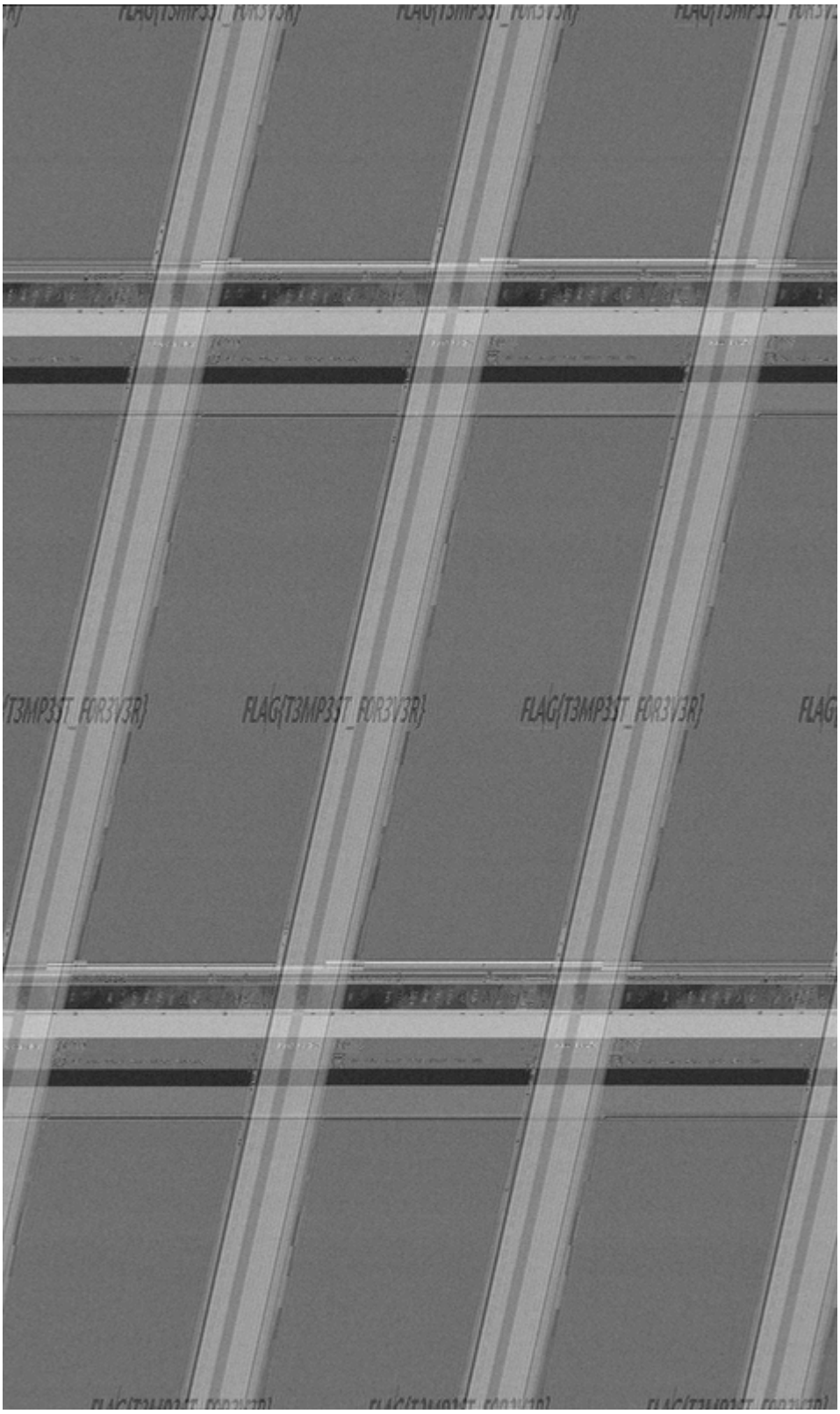
```
import numpy as np
from PIL import Image

data = np.fromfile("le-calme-avant-la-tempest.bin", dtype=np.int16)
width = 890
height = len(data) // width
portion = 1500

image = data[:width * height].reshape((height, width))
image = ((image - image.min()) / (np.ptp(image)) * 255).astype(np.uint8)

Image.fromarray(image[:portion, :]).save("flag_complet_890.png")
```

🔍 Le flag devient parfaitement lisible et aligné : plus d'effet de glissement ou de coupure.

🏁 Flag final récupéré

```
FLAG{T3MP3ST_F0R3V3R}
```

Il apparaît net, en haut de l'image, et est répété plusieurs fois.


FLAG{T3MP33T_F0R3V3R}
FLAG{T3MP33T_F0R3V3R}
FLAG{T3MP33T_F0R3V3R}