Jak zrobić lepszą segmentację zębów?

1. Zebranie jak największej ilości ‘unlabaled data’
2. Zebranie jak największej ilości ‘labaled data’
3. Stworzenie algorytmu dyfuzji (jakiś unet3d albo 2d z hugging face wystarczy), który na bazie Poisson noise (lepiej) lub Gaussian Noise (gorzej) odszumi dane wejściowe (można się zastanowić jak pozbyć się tych pierścieni na obrazach, to też na bank jest jakiś artefakt)

Poisson:

https://tomroelandts.com/articles/gaussian-noise-is-added-poisson-noise-is-applied

**2. Discrete Diffusion with Poisson Sampling**

Define a **forward process** as:

q(xt∣xt−1)=Poisson(λt⋅xt−1)q(x\_t | x\_{t-1}) = \text{Poisson}(\lambda\_t \cdot x\_{t-1})q(xt​∣xt−1​)=Poisson(λt​⋅xt−1​)

Where λt\lambda\_tλt​ is a time-dependent scaling (e.g., increasing noise).

The reverse process must then model a **conditional denoising Poisson process**, which is **non-trivial**.

✅ *True to Poisson statistics*  
⚠️ *Hard to design reverse process; often needs approximation*

Example: You may need to train a model to predict either:

The **mean of the original image** from a noisy sample

Or the **log-intensity**, using reparameterization tricks or score-based estimation.

Wystarczy wytrenować dyfuzję dodając taki szum i odszumiać potem orginalny obraz dla bardzo małych timestamp i powinno działać. Można też wrzucać obraz jako embedding i algorytm sam stwierdza jak duży jest szum i ile powinien go odrzucić, fajny sposób żeby pipeline był jednolity, mniej ingenercji człowieka tym lepiej.

1. Stworzenie potężnego Masked Transforer Autoencodera dla snippetów 3D (wymiary około 64, 64, 96) i stworzenie potężnej latent space
2. Połączenie tego z UNetem i uczenie na danych labalowanych