Lung Cancer Detector & Classifier – Dataset Description

1. Project goal

We are building a lung cancer detector and classifier that works on **PET/CT studies**.

For every patient, the model should answer:

- **Detection / diagnosis:** Does this patient most likely have lung cancer or not?
- **Classification:** If there is cancer, which broad type does it most look like
  (e.g. adenocarcinoma, squamous, etc.)?

The system is meant to **support** doctors, not replace their judgement.

---

2. What our raw data looks like

For each patient we receive a folder that contains **three 3-D DICOM series**:

- **CT** – standard chest CT scan showing anatomy.
- **PET_AC** – attenuation-corrected PET scan (the main PET image used clinically).
- **PET_NC** – non-corrected PET scan, useful for spotting artefacts.

So: **one patient = {CT, PET_AC, PET_NC}**.

We **do not have manual segmentations** of the tumours and, for this project, we also **do not create any**.
All supervision is at the **patient level** (cancer / no cancer, cancer type).

---

3. Data cleaning and formatting

The goal of cleaning is to make the data consistent and easy to feed into a model.

3.1 Basic checks

- Remove patients where any of **CT**, **PET_AC** or **PET_NC** is missing or clearly corrupted.
- De-identify all scans (no names, IDs or dates in the DICOM headers).

3.2 Converting images

- Convert each DICOM series into a 3-D array:
  - CT in **Hounsfield Units (HU)**.
  - PET in standard PET intensity units.
- Store them in a simple format (e.g. Numpy arrays or NIfTI files).

We keep a small index file that maps:

`patient_id → CT_file, PET_AC_file, PET_NC_file` .

3.3 Normalisation and resizing

To make scans comparable:

- **Resampling:** resample all volumes to a common voxel spacing
  (so 1 mm in one patient means the same as 1 mm in another).
- **Resizing:** crop or pad to a fixed 3-D size so we can process batches.
- **Intensity normalisation:**
  - CT: clip extreme HU values and scale to a stable range.
  - PET: clip very high uptakes and normalise per study.

3.4 Labels

We keep a single CSV file with one row per patient, for example:

patient_id	cancer_binary	cancer_type	split
P001	1	adenocarcinoma	train
P002	0	none	val
P003	1	squamous	test

- **`cancer_binary`** : 0 = no cancer, 1 = cancer.

- `cancer_type` : broad histologic type (if known).
- `split` : train / validation / test.

No masks, no bounding boxes – just **patient-level labels**.

4. Exploratory data analysis (EDA)

Before training any models, we did some simple EDA to understand the dataset.

4.1 Counts and splits

- Counted how many patients are:
  - cancer vs no cancer,
  - each cancer type.
- Created **patient-level** train/validation/test splits
  so the same person never appears in more than one split.

4.2 Image quality and shapes

- Checked a few random patients visually (CT + PET) to confirm:
  - all three series load correctly,
  - they are reasonably aligned.
- Plotted distributions of:
  - voxel spacing before and after resampling,
  - CT and PET intensity histograms.

This confirmed that our cleaning pipeline produces consistent inputs.

4.3 Class balance and bias

- Looked at class balance across:
  - cancer vs no cancer,
  - different cancer types.
- As expected, some tumour types are less common.
  - During training we plan to use **class-balanced sampling** and
    data augmentation (e.g. flips, slight rotations) to compensate.

We also noted that scan protocols differ slightly between patients,
which we keep in mind when evaluating generalisation.

5. How this dataset powers our application

Thanks to the cleaning steps, every patient in our dataset is represented as:

- A **CT volume** (anatomy),
- A **PET_AC volume** (metabolic activity),
- A **PET_NC volume** (artefact check),
- Plus a **patient-level label** (cancer yes/no and type).

This is enough for our models to:

1. Learn patterns that distinguish healthy lungs from cancer.
2. Push further into **basic subtype classification**.
3. Be improved over time as we add more patients using the same pipeline.

Even without segmentation, the combination of multi-modal images and
clean labels provides a solid foundation for our lung cancer detector and classifier.