This document outlines the API services for various medical image analysis and prediction tasks. Each service is exposed via a specific route and accepts defined inputs to produce corresponding outputs.

API Service Documentation

Here's a breakdown of each available service:

---

1. Cancer Subtype Classification

- **Endpoint:** `/cancer-subtype/`
- **HTTP Method:** `POST`
- **Description:** Classifies the subtype of cancer from an uploaded medical image.
- **Request:**
    - `file`: An image file (`UploadFile`). The image will be converted to RGB, resized to 224x224, and normalized before being fed to a `CustomSotaCNN` model.
- **Response (`PredictionResponse`):**
    - `predicted_class` (str): The predicted cancer subtype (e.g., 'adenocarcinoma', 'large.cell.carcinoma', 'normal', 'squamous.cell.carcinoma').
    - `confidence` (float): The confidence score of the prediction, rounded to 4 decimal places.
- **Details:**
    - The service uses a pre-trained `CustomSotaCNN` model with weights loaded from `"app/models/CancerClassification/best_custom_sota_model.pth"`.
    - Input images are transformed using ImageNet statistics (mean: `[0.485, 0.456, 0.406]`, std: `[0.229, 0.224, 0.225]`).

---

2. Patient Cancer Classification

- **Endpoint:** `/patient-cancer-classification/predict_patient_cancer_status`
- **HTTP Method:** `POST`
- **Description:** Predicts the cancer status for a patient based on a ZIP archive of their DICOM scan series.
- **Request:**
    - `scan_zip`: A ZIP file (`UploadFile`) containing the DICOM series for one patient.
    - `patient_id_form` (str, optional): An optional patient identifier. If not provided, it's derived from the ZIP filename.
- **Response (`PatientCancerPredictionResponse`):**
    - `patient_id` (str): Identifier for the patient.

    - message (str): Status message of the prediction.
    - probability_cancer (float, optional): The predicted probability of cancer.
    - predicted_class (int, optional): The predicted class (1 for cancer, 0 for no cancer), based on a threshold of 0.5.
    - model_threshold_used (float, optional): The probability threshold used for classification.
    - processed_scan_shape_zyx (List[int], optional): The shape of the processed scan (Z, Y, X).
    - error_details (str, optional): Details if an error occurred.
- **Details:**
    - The service uses a Simp3DNet model with weights from "app/models/PatientCancerClassification/simp3dnet_model_50_each_64cube_best.pth".
    - DICOM series are preprocessed: resampled to target spacing (1.5, 1.5, 1.5), segmented, normalized (HU range -1000 to 400), zero-centered (mean 0.25), and resized to the model's input shape (e.g., 64x64x64).
    - It automatically tries to find the main DICOM series folder within the extracted ZIP archive.

---

3. Nodule Candidate Detection

- **Endpoint:** /nodule-detection/predict
- **HTTP Method:** POST
- **Description:** Detects potential nodule candidates from a CT scan provided as MHD and RAW files.
- **Request:**
    - mhd_file: The MHD header file (UploadFile) for the CT scan.
    - raw_file: The RAW data file (UploadFile) for the CT scan, corresponding to the MHD file.
- **Response (NoduleDetectionResponse):**
    - message (str): Status message of the detection process.
    - nodules (List[NodulePredictionItem]): A list of detected nodule candidates. Each item includes:
        - coordX, coordY, coordZ (float): World coordinates of the nodule.
        - voxelX_resampled, voxelY_resampled, voxelZ_resampled (int): Voxel coordinates in the resampled image.
        - probability (float): The probability score of the nodule.
        - estimated_radius_mm (float): Estimated radius of the nodule in millimeters.

    - scan_shape_zyx_resampled (List[int], optional): The shape (Z, Y, X) of the resampled scan volume used for detection.
- **Details:**
    - Uses a UNet3D_MSA model with weights from "app/models/NoduleDetection/best_model_msa_unetinf.pth".
    - Preprocessing includes: resampling to target spacing (1.0, 1.0, 1.0), HU normalization (clipped between -1000 and 400 HU, then scaled to 0.0-1.0), and lung segmentation.
    - Inference is performed patch-wise (default patch size 96x96x96, stride 48x48x48) on the processed volume.
    - Heatmap peaks are identified, converted to world coordinates, and subjected to 3D Non-Maximum Suppression (NMS) to yield final candidates.

---

4. Nodule Classification

- **Endpoint:** /nodule-classification/classify_candidates
- **HTTP Method:** POST
- **Description:** Classifies provided nodule candidates from a CT scan (MHD/RAW) as nodule or non-nodule.
- **Request:**
    - mhd_file: The MHD header file (UploadFile) for the CT scan.
    - raw_file: The RAW data file (UploadFile) corresponding to the MHD file.
    - candidates_json_str: A JSON string (str passed as a Form field) representing a list of CandidateCoordinateItem. Each item should contain:
        - id (str or int): A unique identifier for the candidate.
        - coordX, coordY, coordZ (float): World coordinates (X, Y, Z) of the candidate.
- **Response (CandidateClassificationResponse):**
    - message (str): Status message of the classification.
    - processed_scan_filename_base (str): Base name of the processed scan file.
    - predictions (List[CandidatePredictionItem]): A list of classification results for each input candidate. Each item includes:
        - Original id, coordX, coordY, coordZ.
        - probability_nodule (float): The model's predicted probability of the candidate being a nodule.
        - predicted_class (int): The predicted class (1 if probability >= threshold, else 0).

        - `error_message` (str, optional): Any error message specific to processing this candidate.
    - `scan_resampled_shape_zyx` (List[int], optional): The shape (Z, Y, X) of the resampled scan.
    - `model_threshold_used` (float): The probability threshold used for classification (default 0.5).
- **Details:**
    - Utilizes a `Simple3DCNN` model with weights from `"app/models/NoduleClassification/candidate_detection_model_best.pth"`.
    - The input CT scan undergoes preprocessing: resampling to target spacing (1.0, 1.0, 1.0), lung segmentation, HU normalization (-1000 to 400 HU), and zero-centering (mean 0.25).
    - For each candidate, its world coordinates are converted to voxel coordinates in the resampled scan. A patch (default size from `PATCH_SIZE_CFG`) is extracted around this center and fed to the model for classification.

---

5. Lung Cancer Prediction

- **Endpoint:** `/cancer-prediction/predict_cancer_level`
- **HTTP Method:** `POST`
- **Description:** Predicts the risk level of lung cancer (Low, Medium, High) based on patient features.
- **Request (`PatientFeatures`):** A JSON object containing the following integer features:
    - `age` (0-120)
    - `gender` (0 or 1)
    - `air_pollution`
    - `alcohol_use`
    - `dust_allergy`
    - `occupational_hazards`
    - `genetic_risk`
    - `chronic_lung_disease`
    - `balanced_diet`
    - `obesity`
    - `smoking`
    - `passive_smoker`
    - `chest_pain`
    - `coughing_of_blood`

    - `fatigue`
    - `weight_loss`
    - `shortness_of_breath`
    - `wheezing`
    - `swallowing_difficulty`
    - `clubbing_of_finger_nails`
    - `frequent_cold`
    - `dry_cough`
    - `snoring`
- **Response (`PredictionResponse`):**
    - `predicted_class` (str): The predicted cancer risk level ('Low', 'Medium', or 'High').
- **Details:**
    - The service loads a k-Nearest Neighbors (KNN) model from `"app/models/LungCancerPrediction/lcp_model.joblib"`.
    - The input features are converted into a NumPy array for prediction by the KNN model.

---

6. Mortality Prediction

- **Endpoint:** `/mortality-prediction/predict`
- **HTTP Method:** `POST`
- **Description:** Predicts patient survival status based on a comprehensive set of features.
- **Request (`MortalityInput`):** A JSON object with the following fields:
    - `age` (float)
    - `gender` (int)
    - `diagnosis_date` (str, e.g., "2023-01-01")
    - `cancer_stage` (int)
    - `beginning_of_treatment_date` (str, e.g., "2023-01-15")
    - `family_history` (int)
    - `bmi` (float)
    - `cholesterol_level` (int)
    - `hypertension` (int)
    - `asthma` (int)
    - `cirrhosis` (int)
    - `other_cancer` (int)
    - `end_treatment_date` (str, e.g., "2023-06-01")
    - `smoke_Current_Smoker` (int)

    - smoke_Former_Smoker (int)
    - smoke_Never_Smoked (int)
    - smoke_Passive_Smoker (int)
    - treatment_Chemotherapy (int)
    - treatment_Combined (int)
    - treatment_Radiation (int)
    - treatment_Surgery (int)
- **Response (JSON Object):**
    - survived (str): "Will Survive" or "Won't Survive".
    - prediction (int): The raw prediction from the model (e.g., 0 or 1).
    - prediction_probability_class_0 (float): Probability for class 0.
    - prediction_probability_class_1 (float): Probability for class 1.
    - processed_input_features_ordered (dict): A dictionary of the processed features used by the model, in the correct order.
- **Details:**
    - Loads a survival model from "app/models/LungCancerMortalityPrediction/survival_model.joblib".
    - Input data undergoes preprocessing:
        - Renaming of smoking and treatment keys (e.g., smoke_Current_Smoker to smoke_current_smoker).
        - Date strings are converted to Unix timestamps.
        - treatment_duration is calculated in days from beginning_of_treatment_date and end_treatment_date.
    - Features are ordered according to MODEL_FEATURE_ORDER before being passed to the model.

---

7. Medical Image Enhancement

- **Endpoint:** /image-enhancement/adjust_contrast/
- **HTTP Method:** POST
- **Description:** Adjusts the contrast of an uploaded medical image.
- **Request:**
    - file: An image file (UploadFile).
    - contrast_factor (float, optional): The factor by which to adjust the contrast. Default is 2.0 (1.0 is original, >1.0 increases contrast).
- **Response:**
    - If successful: A StreamingResponse containing the enhanced image in PNG format (media_type="image/png").

    - If error (e.g., cannot open image): A `ContrastAdjustmentResponse` (JSON) with:
        - `message` (str): Error message.
- **Details:**
    - Uses PIL (Pillow) `ImageEnhance.Contrast` to modify the image contrast.
    - The enhanced image is returned as a PNG.
