VIT Hackathon 2024 –
Problem Statement 02: Surgical Smoke Detection

Background

Laparoscopic surgery commonly known as Minimally Invasive Surgery (MIS), in which operations are performed through small incisions elsewhere in the body rather than through large incisions as with traditional open surgery. During this surgical process, a laparoscope (highresolution camera) and a light source are inserted through one of the small incisions. This allows the surgeons to view the inside of the body on a monitor.

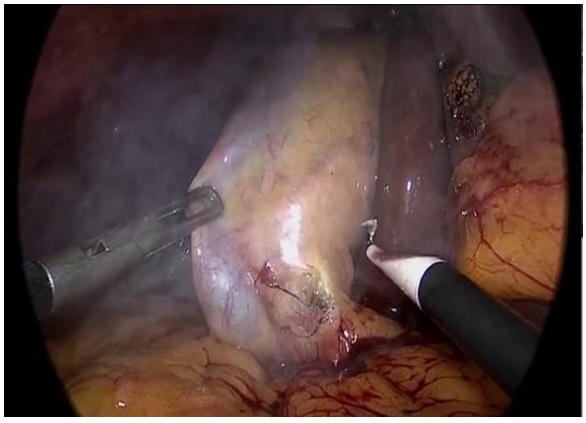
One of the many challenges that surgeons face during laparoscopic surgery is heavy smoke development (caused due to cauterization) that reduces the visibility of the surgery on the monitor. Therefore, there is a necessity in accurate and reliable recognition of smoke that can trigger automated smoke evacuation systems. We are looking to develop a deep learning model to detect smoke and non-smoke frames during surgical process.

Task Overview

Develop a deep learning model to classify given sample (set of frames) as **SMOKE** (label=1) or **NON-SMOKE** (label=0). The Model can be any deep learning-based Architecture. (custom CNN model is allowed).

Note: The prediction should be based on samples, not individual frames.

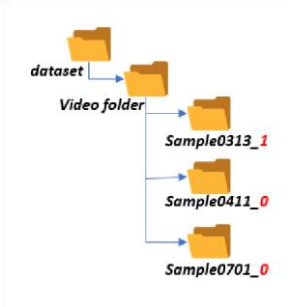
Non- Smoke



Smoke

Dataset

- The dataset contains 60 surgical videos. Each video folder contains several sub-folders (each sub-folder is considered as a sample). The sub-folder name contains class label, '*folder*_1' resembles smoke sample and '*folder*_0' resembles non-smoke sample.
- Each sample consists of a set of frames which are of 640x360 pixels.



Evaluation Metrics

- Metrics: overall Accuracy, F1 score, Recall, Precision, Confusion matrix
- Participants are required to plot metric and loss curves.

Submission Guidelines

- Participants should submit their trained models along with inference code.
- Participants should upload all the code into a GitHub repository. The submitted code should be able to run on our end without any changes other than a simple path change.
- Participants should submit a pdf along with the code outlining the detailed steps to run the code and the thought process and approach taken in detail to solve the problem statement.
- Results will be evaluated on a separate test set (not provided during the hackathon).
- Teams can use any deep learning framework (e.g., PyTorch, TensorFlow).
- In case a Bonus Challenge is attempted it should be mentioned along with all its details.

Bonus Challenges **(additional points for successful completion)**

- **Real-Time Inference**: Optimize your models for real-time performance.

Good luck to all participants! May your neural networks converge swiftly and your loss functions decrease steadily!