V-Strykathon
Team: Leastcoders
Problem Statement: 2

Members:
Vinithra Balaji - 21BCE1178
Punnoose Punnen - 21BCE1175
Thejas Sanjeev -21BCE1797

1. STEPS TO RUN OUR MODEL

- *Environment*

We trained, tested and ran our models and implementation on Kaggle's private virtual machines (dataset integrity was maintained - it was uploaded as private only access)

- *Tools used*

Tensorflow **2.15.1,** keras, opencv

- *Steps to run:*

Note: please ensure while running the model that **Tensorflow 2.15.1** specifically is being used as the model was exported in 2.15.1 and might not be backward compatible.

For Testing: The pixel based custom CNN model
In the codes folder, open the "testing-cnn.ipynb" notebook and run all the code blocks in order. All blocks have been commented accordingly for thorough understanding and inspection. "data_dir" is the path variable where the path of the testing data set has to be replaced with. The results will be in a csv format as given per the instructions. (the "predictions-cnn.csv" is in the "results" folder of the google drive)

For Testing: The HSV thresholding technique
In the codes folder, open the "testing-hsv.ipynb" notebook and run all the code blocks in order. All blocks have been commented accordingly for thorough understanding and inspection. "data_dir" is the path variable where the path of the testing data set has to be replaced with. The results will be in a csv format as given per the instructions. (the "predictions-hsv.csv" is in the "results" folder of the google drive)

For Testing: The combined weighted approach where we fuse the predictions of both the CNN & and the HSV technique
In the codes folder, open the "testing-combined.ipynb" notebook and run all the code blocks in order. All blocks have been commented accordingly for thorough understanding and inspection. "data_dir" is the path variable where the path of

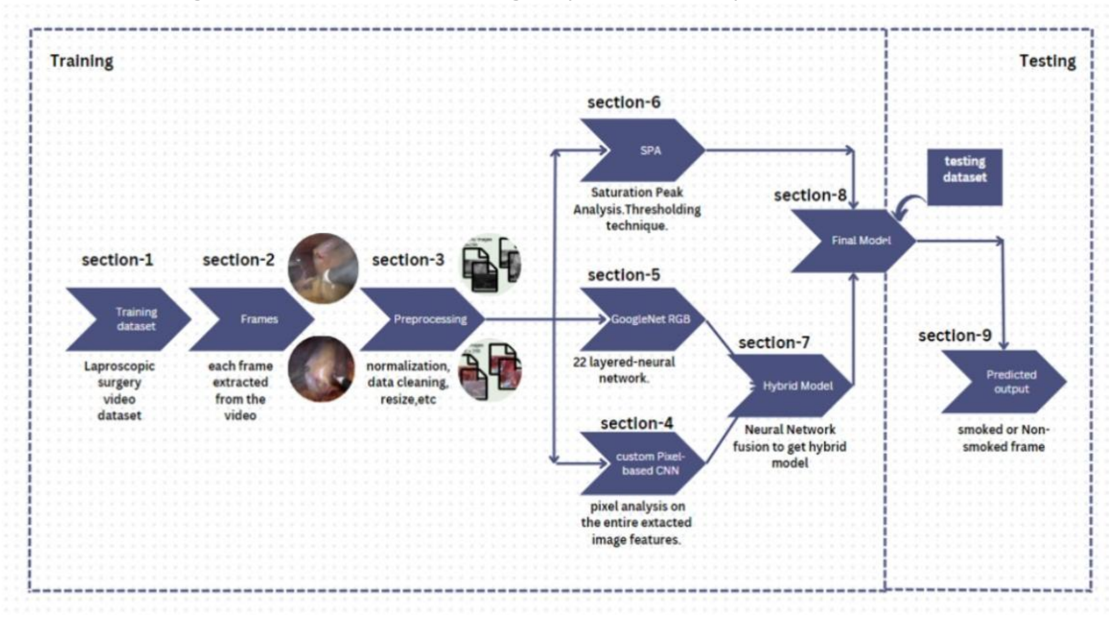
the testing data set has to be replaced with. The results will be in a csv format as given per the instructions. (the "predictions-combined.csv" is in the "results" folder of the google drive)

For Training: The pixel based custom CNN model
In the codes folder, open the "training-cnn.ipynb" notebook and run all the code blocks in order. All blocks have been commented accordingly for thorough understanding and inspection. "data_dir" is the path variable where the path of the testing data set has to be replaced with.

<u>**BONUS CHALLENGE**</u>
For Real-Time detection: In the codes folder, open the "real-time.ipynb" notebook and run all the code blocks in order. All blocks have been commented accordingly for thorough understanding and inspection. "data_dir" is the path variable where the path of the testing video in "mp4" format has to be uploaded. The output will be saved as "detected.mp4" where the frames with smoke detected will be labeled appropriately.

2. OUR THOUGHT PROCESS

The following was our initial thought process as part of the idea submission:


Training
Testing
section-1
Training dataset
Laproscopic surgery video dataset
section-2
Frames
each frame extracted from the video
section-3
Preprocessing
normalization, data cleaning, resize,etc
section-6
SPA
Saturation Peak Analysis.Thresholding technique.
section-5
GoogleNet RGB
22 layered-neural network.
section-4
custom Pixel-based CNN
pixel analysis on the entire extacted image features.
section-7
Hybrid Model
Neural Network fusion to get hybrid model
section-8
Final Model
testing dataset
section-9
Predicted output
smoked or Non-smoked frame

But as the hackathon proceeded based on the valuable inputs from the jury panel we came to a conclusion to simplify the architecture and focus more on the custom pixel based CNN model's development rather than the pre-trained InceptionV3 model which tends to ineffective for the Surgical scenario as it can't leverage it's image-net database also considering the temporal factor for the video sequences we decided not to make use of Inception V3.

3. APPROACH TAKEN TO SOLVE PS-2

The approach we took to finally solve this problem statement of surgical smoke detection was to develop and train a custom pixel based CNN in which we made use of 3 convolutional 3D layers as the base taking the temporal consistency of 10 frames per sample present in the dataset given as the depth for the input shape. Along with the CNN we thought of leveraging digital image processing techniques to solve the problem statement. We went with a HSV based threshold technique to detect the non smoke pixels in a frame which is then used to account for whether smoke is present or not in a frame. We then took the average of the predictions of 10 frames for each sample to predict and label each sample as "smoke"(1) or "no_smoke" (0).

Both the CNN and the thresholding technique are combined through a weighted approach where we consider both predictions and prefer the one where it excels in particular. Example: during testing we found out that the frames labeled by the thresholding technique as "no_smoke" where always very accurate, so we gave the prediction of "no_smoke" from thresholding technique more weight over the CNN's prediction which helped us to increase our accuracy

4. BONUS CHALLENGES

- **Real-Time smoke detection**

We process the video 10 frames at a time to consider the temporal correlation of the sequences for finding the smoke development across frames. Each of these sequences are sent for prediction to our combined Pixel CNN model & HSV threshold technique which gives a weighted prediction of the individual predictions ie. smoke or no smoke. After getting the predictions we label all the frames using PILLOW library and then stitch the frames back together and add it to the output video at real time.

Our inference is that sequence by sequence detection for smoke over say 10 frames is better than frame by frame detection as the temporal consistency ensures we can gather more features in regards to the smoke's development in the video at real time