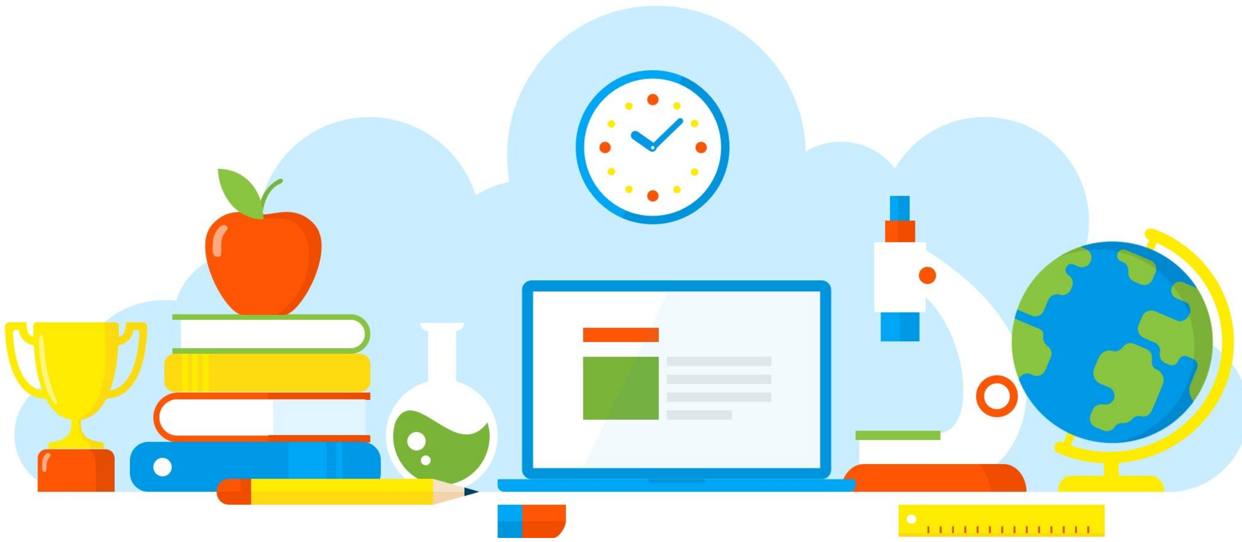


PROJECT REPORT

FACE-MASK DETECTION on Live webcam feed using OpenCV, Facenet

SUHAS S
19/04/2021

Introduction

Coronavirus disease 2019 (COVID-19) is an emerging respiratory infectious disease caused by Severe Acute Respiratory Syndrome coronavirus 2 (SARS-CoV2) . At present, COVID-19 has quickly spread to the majority of countries worldwide, affecting more than 14.9 million individuals, and has caused 618,017 deaths, according to the report from the World Health Organization (WHO) on 23 July 2020 (https://covid19.who.int/). To avoid global tragedy, a practical and straightforward approach to preventing the spread of the virus is urgently desired worldwide. Previous studies have found that facemask-wearing is valuable in preventing the spread of respiratory viruses . For instance, the efficiencies of N95 and surgical masks in blocking the transmission of SARS are 91% and 68%, respectively . Facemask-wearing can interrupt airborne viruses and particles effectively, such that these pathogens cannot enter the respiratory system of another person . As a non-pharmaceutical intervention, facemask-wearing is a non-invasive and cheap method to reduce mortality and morbidity from respiratory infections. Since the outbreak of COVID-19, facemasks have been routinely used by the general public to reduce exposure to airborne pathogens in many countries . In addition to patients suspected of actual infection with COVID-19 being required to wear facemasks for the prevention of virus spreading, healthy persons also need to wear facemasks in order to protect themselves from infection . Facemasks, when fitted properly, effectively disrupt the forward momentum of particles expelled from a cough or sneeze, preventing disease transmission . However, the effectiveness of facemasks in containing the spread of airborne diseases in the general public has been diminished, mostly due to improper wearing . Therefore, it is necessary to develop an automatic detection of face mask on crowd in public places, which can contribute to personal protection and public epidemic prevention.

In this project, we will be developing a face mask detector that is able to distinguish between faces with masks and faces with no masks. In this report, we have proposed a detector which employs Facenet for face detection and a neural network to detect presence of a face mask. The implementation of the algorithm can be used on images, videos and live video streams.

Dataset

To train a deep learning model to classify whether a person is wearing a mask or not, we need to find a good dataset with a fair amount of images for both classes:

1. wearing a mask
2. not wearing a mask

Real World Masked Face Dataset (RMFD) suits our requirement.

The RMFD provides 2 datasets:

1. Real-world masked face recognition dataset: it contains 5,000 masked faces of 525 people and 90,000 normal faces.
2. Simulated masked face recognition datasets.

In this project, we are going to use only the first dataset.

After downloading and unzipping the dataset, its structure looks as follows:

```
self-built-masked-face-recognition-dataset
├AFDB_masked_face_dataset
│ ├subject-id
│ │ ├image-id.jpg
│ │ └...
│ └...
└AFDB_face_dataset
  ├subject-id
  │ ├image-id.jpg
  │ └...
  └...
```

- We create our pandas DataFrame by iterating over the images and assigning to each image a label of 0 if the face is not masked, and 1 if the face is masked. The images of this dataset are already cropped around the face, so we won't need to extract the face from each image.
- Since Images in dataset have different dimensions, all images are resized to 224*224 dimension.
- Since the dataset is large , only 1000 images with face mask and 1000 images without face mask are considered for training the model.
- The dataset is split into Training and Test data in the test data ratio of 0.2

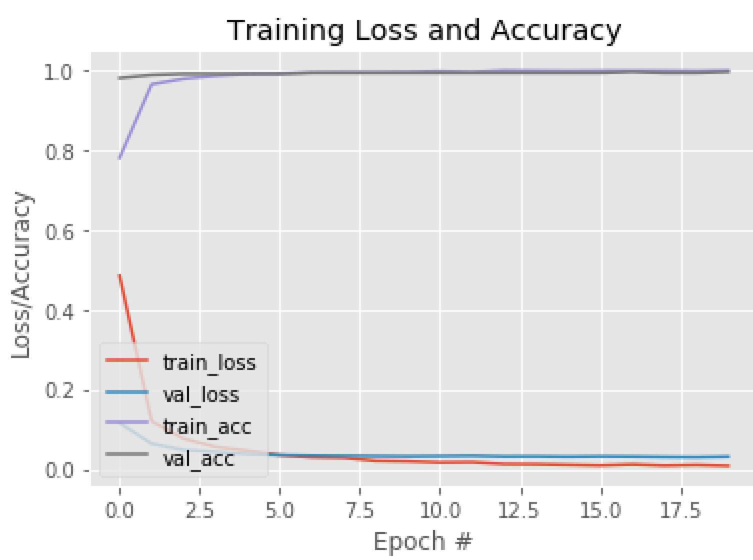
Training the Model

- We use Keras and TensorFlow to train a classifier to automatically detect whether a person is wearing a mask or not.To accomplish this task, we'll be fine-tuning the MobileNet V2 architecture, a highly efficient architecture that can be applied to embedded devices with limited computational capacity

Fine-tuning setup is a three-step process:

- Load MobileNet with pre-trained ImageNet weights, leaving off head of network
- Construct a new FC head, and append it to the base in place of the old head
- Freeze the base layers of the network . The weights of these base layers will not be updated during the process of backpropagation, whereas the head layer weights will be tuned.

- Here we compile the model with the Adam optimizer, a learning rate decay schedule, and binary cross-entropy for a batch size of 32. We run it for 20 epochs
- The model is saved in '.H5' format.
- The summary of the model training is as follows



Implementing the Model

.

We follow the following steps to implement the model

- Load live stream from a webcam
- Detect faces in the image
- Apply our face mask detector to classify the face as either with_mask or without_mask
- Display appropriate Status Message and counts

We define a **detect_and_predict_mask -**This function detects faces and then applies our face mask classifier to each face ROI

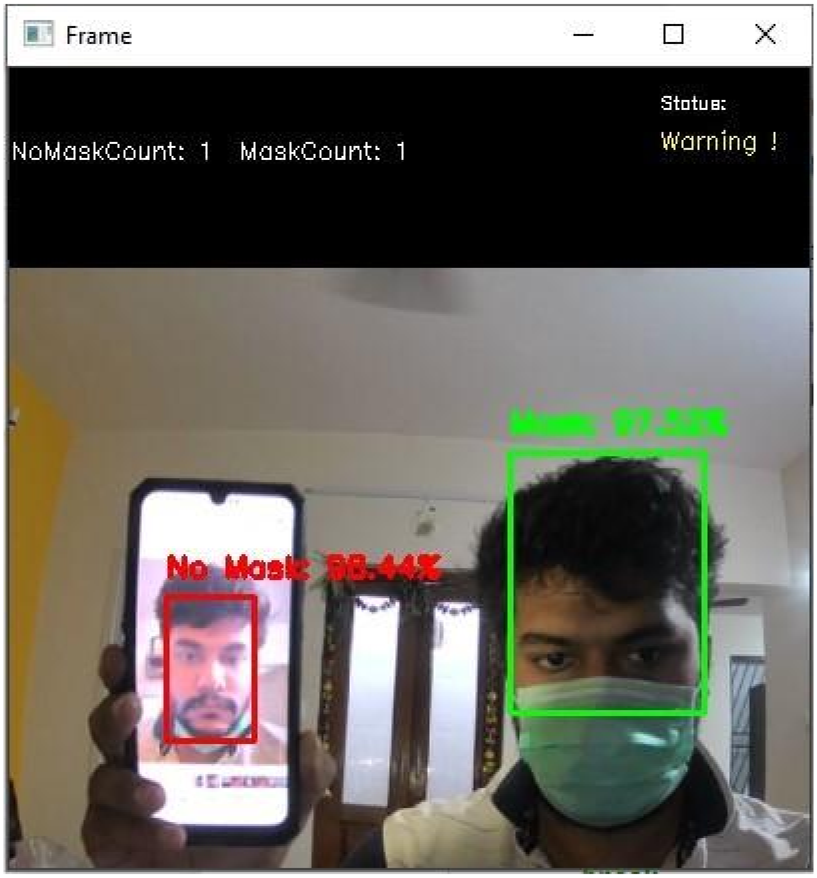
Our detect_and_predict_mask function accepts three parameters:

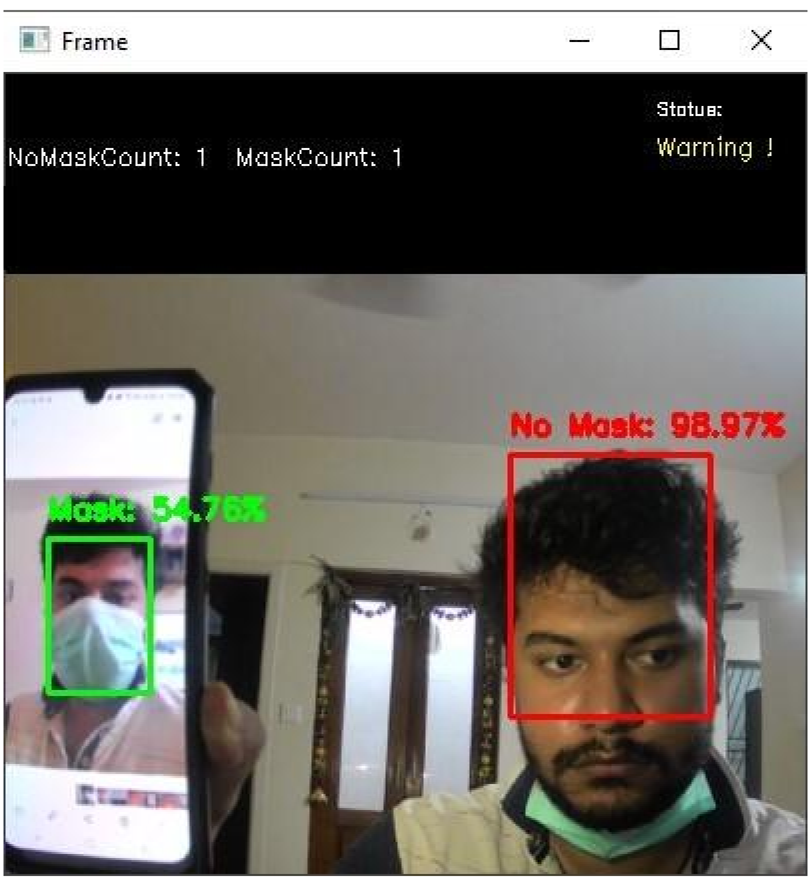
- frame: A frame from our stream
- faceNet: The model used to detect where in the image faces are
- maskNet: Our COVID-19 face mask classifier model

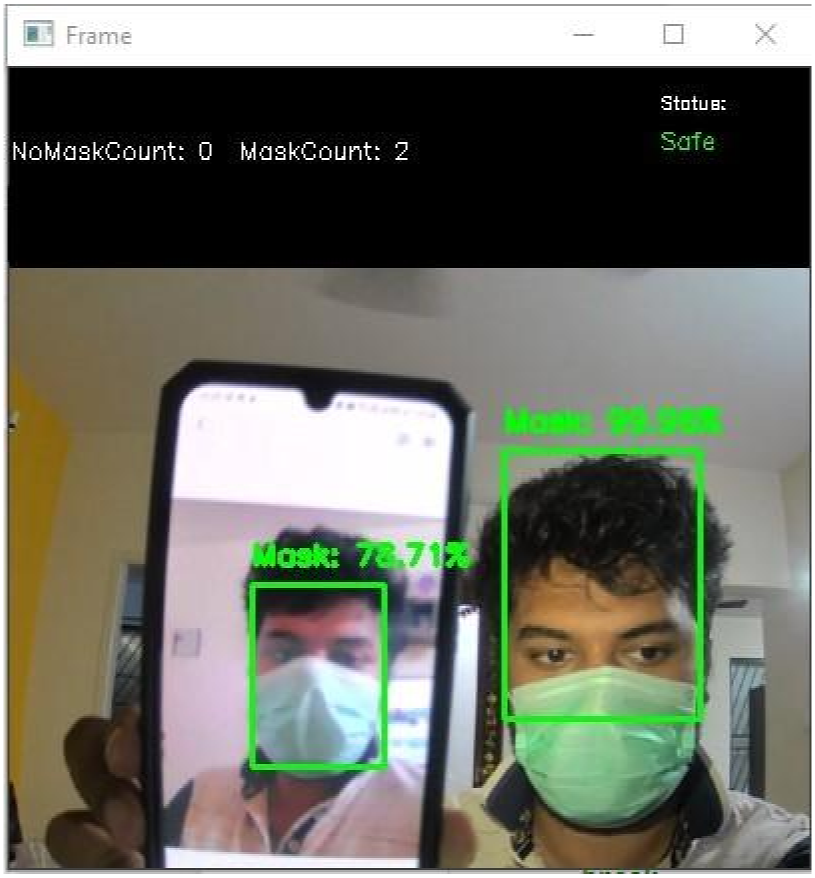
Inside, we construct a blob, detect faces, draw box around the face and make predictions.We follow the steps:

- Unpack a face bounding box and mask/not mask prediction
- Determine the label and color
- Annotate the label and face bounding box
- Update the number of Masked and Unmasked Face count
- Finally, we display the results.

OUTPUT


Frame
Status:
Warning !
NoMaskCount: 1
MaskCount: 1


Frame
Status:
Warning !
NoMaskCount: 1
MaskCount: 1
No Mask: 98.97%
Mask: 54.76%



DEPLOYMENT

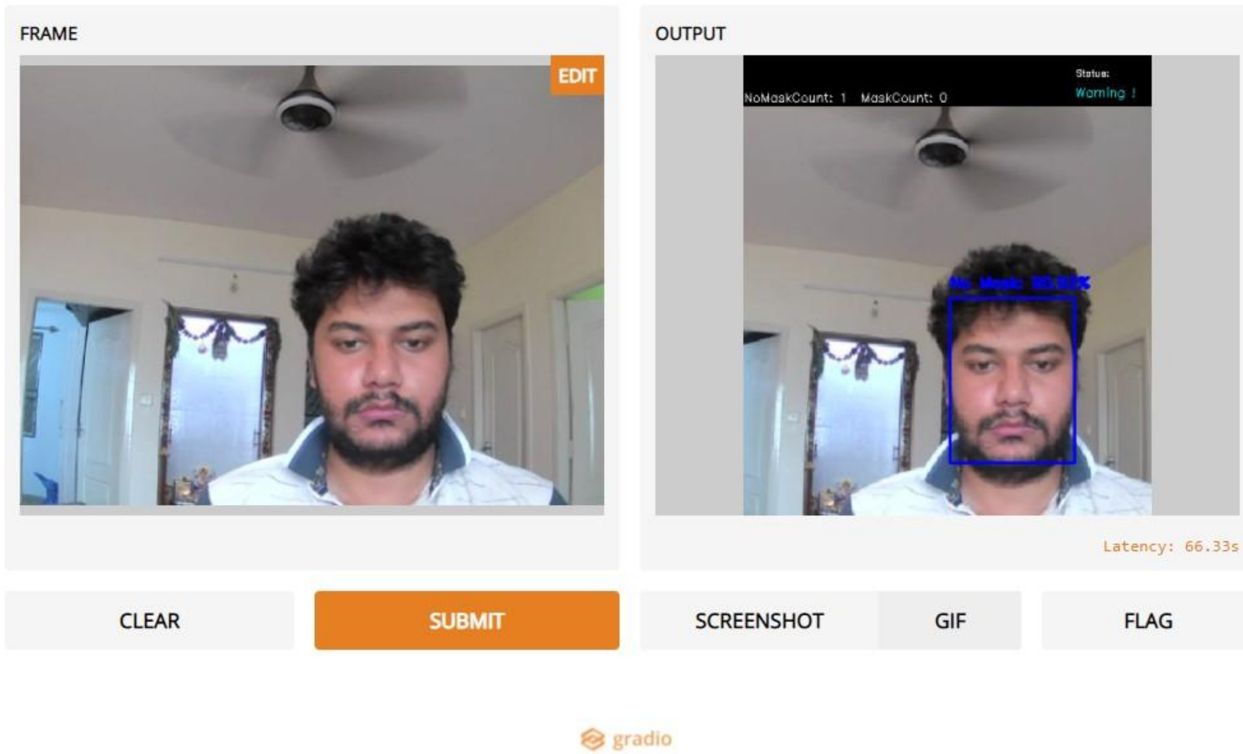
The deployment of the model was done using Gradio

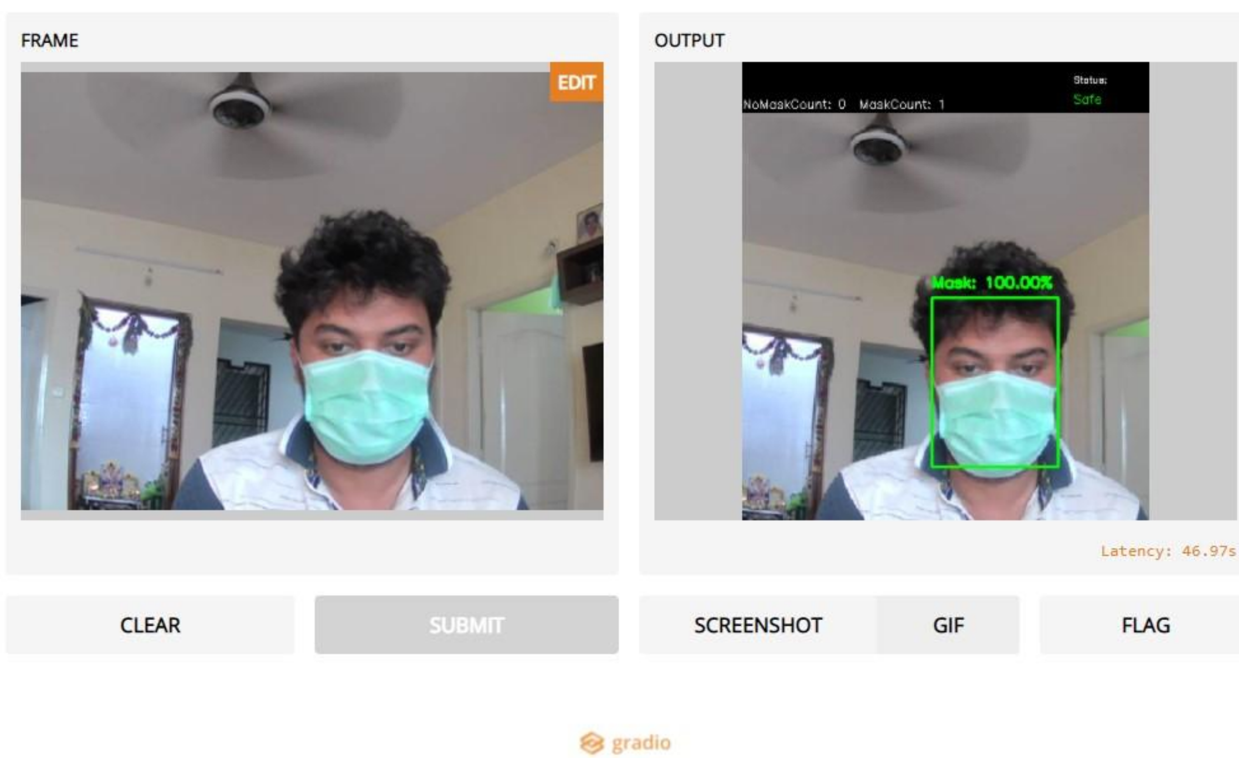
Gradio is a Python library that lets you create UIs around your machine learning model by specifying 3 things: a Python function, input component, and an output component.

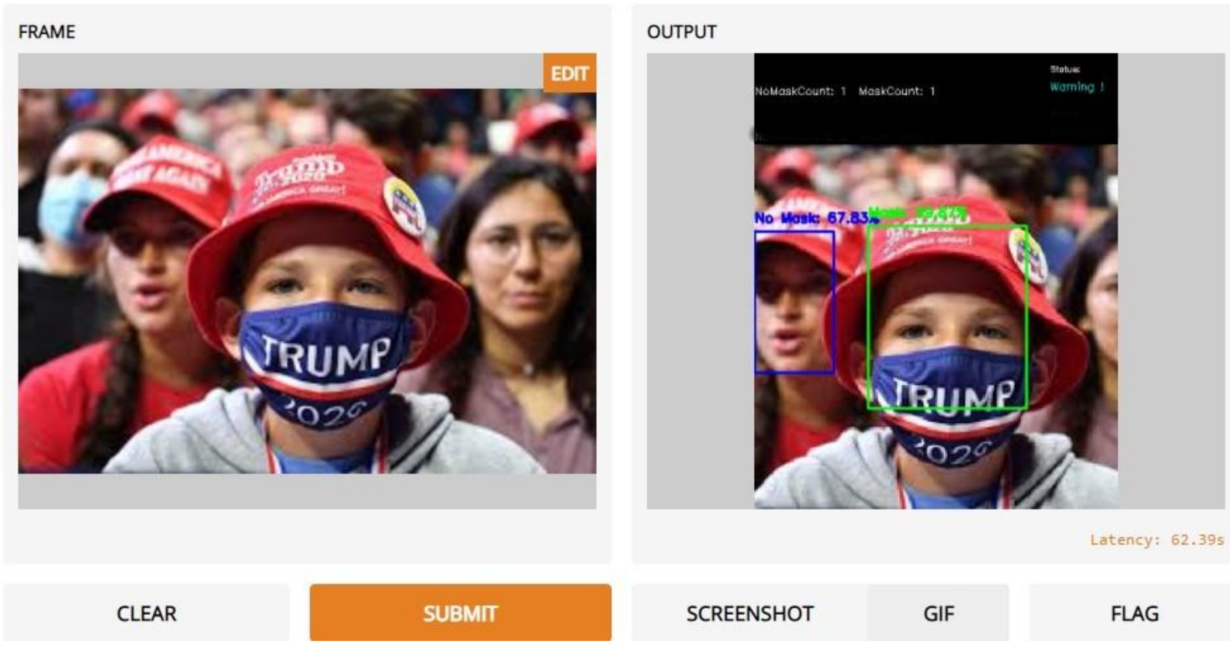
Python Function =Mask detection model

Input component =Webcam input video

Output component = Image with results


FRAME
EDIT
OUTPUT
NoMaskCount: 1 MaskCount: 0
Status:
Warning !
Latency: 66.33s
CLEAR
SUBMIT
SCREENSHOT
GIF
FLAG
gradio


FRAME
EDIT
OUTPUT
NoMaskCount: 0 MaskCount: 1
Status:
Safe
Mask: 100.00%
Latency: 46.97s
CLEAR
SUBMIT
SCREENSHOT
GIF
FLAG
gradio


FRAME
EDIT
OUTPUT
NoMaskCount: 1 MaskCount: 1
Status:
Warning !
No Mask: 67.83%
Latency: 62.39s
CLEAR
SUBMIT
SCREENSHOT
GIF
FLAG

Thank You.