REAL-TIME SMART REACTIONS IN VIRTUAL MEETS

A PROJECT REPORT

Submitted by

Raghav Agarwal (19MIM10024)
Indrashish Paul (19MIM10046)
Shobhit Vatsya (19MIM10040)

in partial fulfillment for the award of the degree

of

MASTER OF TECHNOLOGY(Integrated)

In

COMPUTER SCIENCE AND ENGINEERING

Specialization in

Artificial Intelligence and Machine Learning

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRI KALAN, SEHORE
MADHYA PRADESH - 46611

May 2021

VIT BHOPAL UNIVERSITY, KOTHRI KALAN, SEHORE
MADHYA PRADESH – 466114

BONAFIDE CERTIFICATE

Certified that this project report titled "**Real-time smart reactions in virtual meets"** is the bonafide work of "Raghav Agarwal (19MIM10024)", "Indrashish Paul (19MIM10046)" and "Shobhit Vatsya (19MIM10040)", who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported here does not form part of any other project / research work on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR

Dr. V Pandimurugan,

Program Chair, M.Tech Integrated

School of AI & ML division

VIT BHOPAL UNIVERSITY

PROJECT GUIDE

Dr. Paras Jain,

Senior Assistant Professor,

School of AI & ML division

VIT BHOPAL UNIVERSITY

The Project Exhibition II Examination is held on Month of May 2021

ACKNOWLEDGEMENT

First and foremost we would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

We wish to express our heartfelt gratitude to Dr. V. Pandimurugan, Program Chair, School of AI & ML division, for much of his valuable support and encouragement in carrying out this work.

We would like to thank my internal guide Dr Paras Jain, for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

We would like to thank all the technical and teaching staff of the School of Computer Science Engineering, who extended directly or indirectly all support.

Last, but not the least, We are deeply indebted to our parents who have been the greatest support while we worked day and night for the project to make it a success.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

Virtual meetings are hyped up in this new normal. While attending any online conference/ lecture, it is quite **difficult to mute and unmute** time and again for small reactions, especially when someone else is speaking.

We have to unmute ourselves and interrupt the speaker in the process, even for small reactions like- ‘yes’, ‘no’, ‘Hi’, ‘I have a question’, ‘Boring’, ‘Funny’ etc. Doing this, interrupts the flow of the meeting and creates an uncomfortable noise if several people give their reactions at the same time using their microphones.
This problem can be smartly solved using **Computer Vision** by giving Smart Emoji based reactions.

We have used a pre-trained Hand Gesture Recognition model to classify our hand gesture into its respective class. Once the classification is done, we have mapped that classification result with real time emojis which are displayed alongside the video of the user.

PROBLEM STATEMENT

It is quite difficult and irritating in an online meeting to again and again unmute and mute oneself, just for small reactions like ‘yes’, ‘no’, ‘Hi’, ‘I have a question’, ‘Boring’, ‘Funny’ etc.

Proposed a work which can overcome the above stated problem.

EXISTING WORKS AND LITERATURE REVIEW

All the virtual meeting platforms like- Google Meet, Zoom, Microsoft Teams, Cisco Webex, etc, have the Virtual Reactions feature through which users can conduct non-verbal communication with the help of clickable emoticons.

In August, 2014, Zoom first added the "Raise Hand" feature during a meeting where the participants can simulate raising a hand in virtual meetings using a clickable emoticon. It was followed by further enhancements to the raise hand feature and the addition of support for emojis in group messaging. **[1]**

On February 28, 2017, Zoom added and enhanced the non-verbal feedback feature. During a presentation, participants can provide non-verbal feedback without interrupting the flow of the presentation. Participants have the ability to raise their hand, signal yes or no, indicate to go faster or slower, and use emojis.

In June, 2020, Microsoft Teams **[2]** and on Nov, 2020, Google Meets **[3]** also added the "Raise Hand" feature using a clickable icon. This was a great improvement to their respective platforms but no Participant was able to express their emotions during a virtual meet unlike Zoom.

On August 4, 2020, additional meeting reactions were added to Zoom meetings. In addition to the clapping and thumbs up reactions, meeting participants could then react with a heart, surprised face, laughing face, or party emoji. Those receiving the reactions would also have needed to have the latest version of the Zoom client to see them.

On December 10, 2020, Microsoft Teams publicised their release of a new version with "Live Meeting Reactions" in addition to the "Raise Hand" feature using clickable emoticons and live animations depicting the respective emotions. Since then MS Teams had 4 exclusive emoticons for Meeting reactions. The Meeting Reactions feature allows users to choose from four different reaction types – like, love, applause, and laugh. Users can click on a reaction to send it to the meeting.

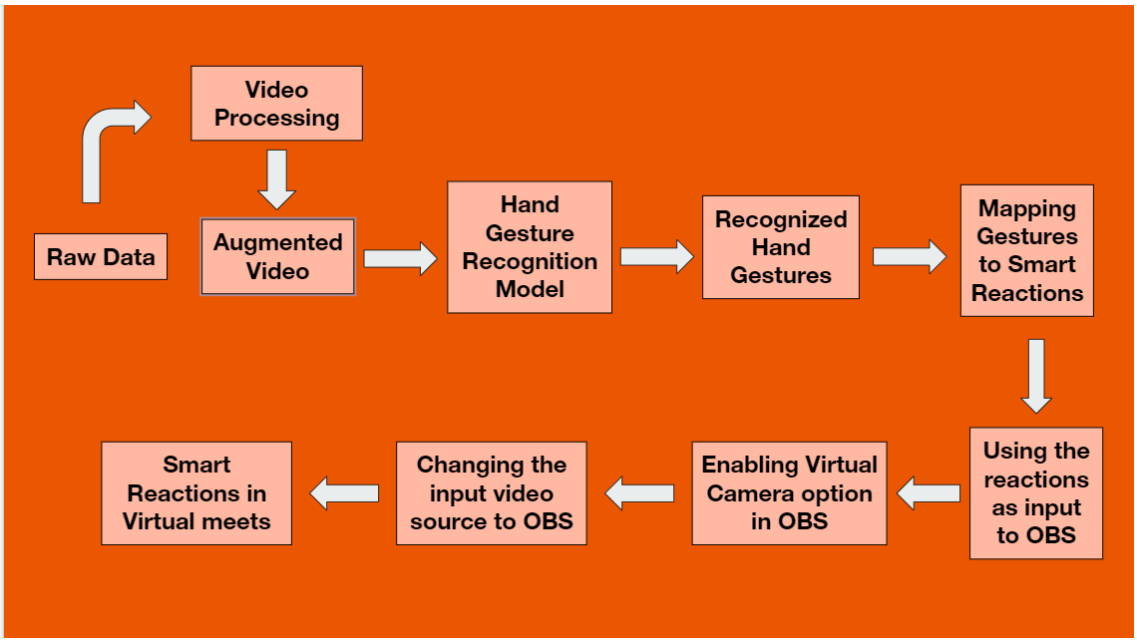
Participants in the meeting will see the reaction over the user's position on the meeting stage, and the user will also see the reaction displayed on the user's "ME" box at the lower right of the meeting stage. **[4]**

On December 21, 2020, Zoom meetings enhanced their nonverbal feedback and reactions. Nonverbal feedback options were moved to the Reactions button, and all reactions would have been shown in the corner of the participants' video. The host would see each participant's reactions in the participant list, as well as the aggregate numbers of each reaction at the bottom of the list. **[5]**

On April 19, 2021, Zoom meetings added a full emoji suite for Non-verbal reactions during a meeting. Any emoji available for Zoom chat can be used as a reaction in a meeting. The host can also restrict reactions to the standard set, or allow use of the full emoji suite. This is the latest update of Meeting Reactions in any major platforms. **[6]**

PROPOSED WORK AND METHODOLOGY

Proposed work is to find and develop a real-time solution of the above stated problem statement by using a Hand Gesture Recognition model to classify the input video stream into the desired number of classes (like- 'yes', 'no', 'Hi', 'I have a question', 'Boring', 'Funny' etc).



[Fig 1 - Workflow]

As shown in Fig. 1, overall steps involved in the process are as follows:

1. Building a Hand Gesture Recognition model or selecting a pre-trained model
2. Using the above model to classify and recognize hand gestures from the input video stream.
3. Mapping the recognized hand gestures to their respective smart reactions.
4. Deploying our build over any cloud server- GitHub or Heroku.
5. Using the above resulting video as an input for **OBS**(Open Broadcaster Software)[7].
6. Enabling the virtual camera option in **OBS**.
7. Changing our input video source from default webcam to **OBS** virtual camera in the online meeting platform (In our case - **Zoom**).

NOVELTY OF THE PROJECT

- There are lots of emoji reactions available in virtual meeting applications but for sending those, we have to select a reaction by clicking over the emoticons.

- Instead of leaving the stickers or emoji in the chat, we can directly display the emoji filters alongside the users' real-time video.

- All we need is to show the hand gestures in front of the camera to express our reactions and this can be done without interrupting the flow of the meeting.

- It will really add to the interactivity of the meetings and will be loved by the attendees too.

REAL-TIME USAGE

This solution could be very helpful in this new normal where online meetings and virtual education is essential.
We can use this solution in any online meeting- lectures, classes, seminars, meetings, conferences etc.
One very interesting use-case is for using sign language. For the people who can not speak, it's very hard to communicate with other people as most of the time the other party does not know the sign language. But, with this innovative technology, they can communicate very easily. We just have to make some custom gestures and let our machine understand them. The person can show their sign, which is ofcourse a hand gesture in front of the camera, our model will detect the sign made by that hand gesture and will display an appropriate emoji to the other users in the meeting.

For example, the sign language hand gesture for 'I love you' is made by full curl of the middle finger and the ring finger, and the rest of the fingers are in vertical up direction with no curl. So, once we trained our model with this custom gesture, our model can classify this sign.

[Fig 2- Sign language detection]

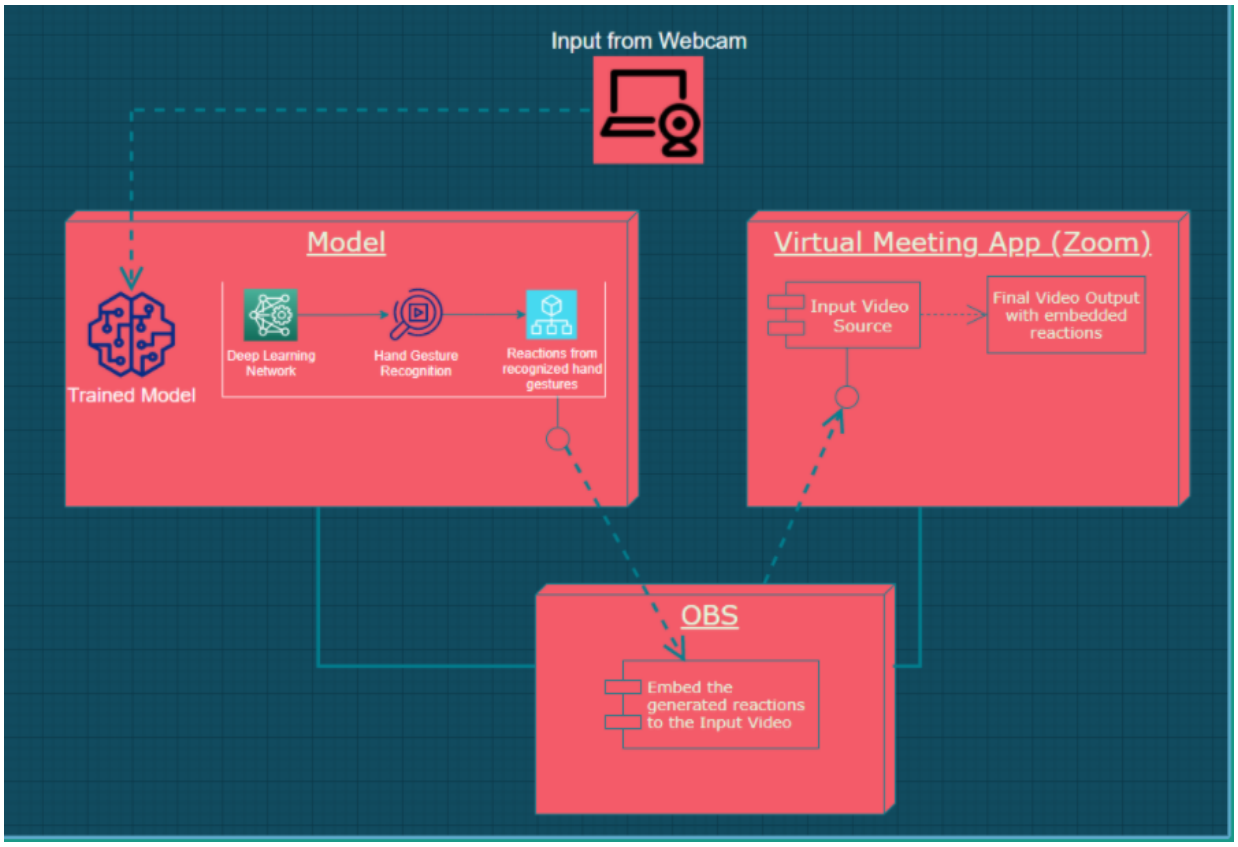
In figure-1, we can clearly see that Raghav has made a sign for "i love you", which is recognised by the model and a corresponding emoji is added into the video of the person itself. By this the other members of the meeting, who were before not understanding the sign language, can easily understand what the other person has to say.

HARDWARE AND SOFTWARE REQUIREMENTS

1. **Development Requirements**
    a. **Hardware:** A desktop or a laptop with:
        i. RAM: At least 8 GB
        ii. External or Internal WebCam
        iii. Memory: At least 5 GB of disk space and 1 GB of virtual memory
        iv. CPU: Quad core Intel Core i5 or higher
        v. GPU: Optional but recommended
    b. **Software:**
        i. Programming Language support: Python(>=3.6), JS, React
        ii. Additional support for node.js and git CLI
        iii. VS Code or any other IDEs and Jupyter notebooks
        iv. Open Broadcasting Software (OBS)
        v. Zoom - A virtual meeting application

2. **Testing and User Requirements**
    a. **Hardware:** A desktop or a laptop with:
        i. RAM: At least 4 GB
        ii. External or Internal WebCam
        iii. Memory: At least 1.5 GB of disk space and 750 MB of virtual memory
        iv. CPU: Quad core Intel Core i3 or higher
        v. GPU: Optional
    b. **Software:**
        i. Open Broadcasting Software (OBS)
        ii. Zoom - A virtual meeting application

OVERALL SYSTEM ARCHITECTURE DIAGRAM



[Fig 3- Module Architecture Diagram]

MODULE DESCRIPTION

Our entire workflow has been divided into following modules:

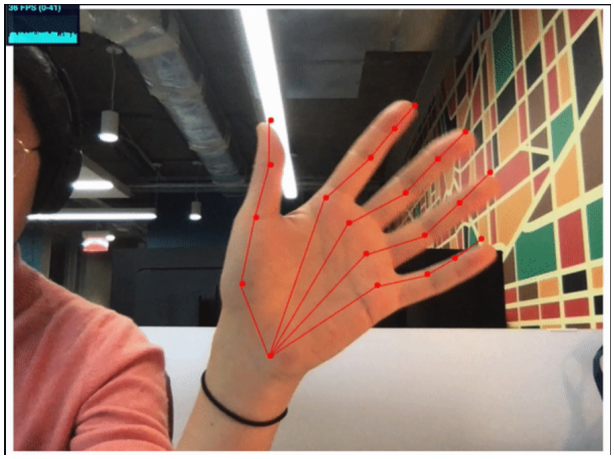
- Data Collection:
  - Hand Gesture image dataset of size ~2GB from Kaggle.
- Hand Gesture Recognition using self trained model (Python)
  - Training Deep Learning model: Building and training a Deep Learning based model to classify hand gestures.
  - Real-Time Hand Gesture Recognition: Recognising the gesture shown by the hand out of the entire image frame using the trained model.
- Hand Gesture Detection and Recognition using Posenet and fingerpose **[9]**.
  - Hand Pose Estimation: Estimation of Hand Pose and drawing a mesh over the detected state of hand in real time.
  - Hand Gesture Recognition: Classifying the gesture into the desired class.
- Deployment:
  - Deploying our build in a cloud server- GitHub or Heroku.
- Smart Reactions in Zoom meetings:
  - Projecting the output video as an Input video for Zoom video app using OBS.

MODULE WORKFLOW EXPLANATION

We used two approaches to build or have a Gesture Recognition Model:
First, we trained our own model using a hand gesture image dataset of size ~ 2GB using Deep Learning technique. All that implementation is added into the GitHub repository of our project, which can be found here**.**
But, the results we were getting were not up to mark for smooth implementation of our idea. So, we looked up for an alternative and found a pre trained model called- 'fingerpose' **[9].** We will discuss that approach in this section in detail.

1. **Hand Gesture Detection using Handpose model:**

   The pre-trained model used for this purpose is handpose model **[8]** from Tensorflow.js-models.
   MediaPipe Handpose is a lightweight ML pipeline consisting of two models: A palm detector and a hand-skeleton finger tracking model. It predicts 21 3D hand keypoints per detected hand.Given an input, the model predicts whether it contains a hand. If so, the model returns coordinates for the bounding box around the hand, as well as 21 keypoints within the hand, outlining the location of each finger joint and the palm.



[Fig4 - HandPose model]

   For installation of handpose model - "npm install @tensorflow/tfjs @tensorflow-models/handpose", run this command in the terminal of your project directory.

The substeps under this module are:

- Installing dependencies- tensorflow/tfjs, tensorflow-models/handpose, react-webcam.
- Importing dependencies
- Setting up webcam and canvas
- Defining references to webcam and canvas
- Loading our handpose model
- Writing detect function
- Drawing the hand skeleton and keypoints over canvas.

2. Hand Gesture Recognition using fingerpose:

The model used for this purpose is- ‘fingerpose’ model[**9**].
Finger pose classifier for hand landmarks detected by TensorFlow.js' handpose model. It can detect hand gestures like "Victory" ✌️or "Thumbs Up" 👍inside a webcam source picture.

Gesture detection works in three steps:

- Detect the hand landmarks inside the video picture.
- Estimating the direction and curl of each individual finger.
- Comparing the result to a set of gesture descriptions.

First step is performed using TensorFlow’s “handpose”, the other two steps are done by this library.

[Fig. 5 - FingerPose Model]

In our project we have used 5 gestures, namely- ‘thumbs-up’, ‘thumbs-down’, ‘victory-sign’, ‘love’, ‘palm raised’.

Among these, only two gestures are predefined- ‘thumbs-up’ and ‘victory’. Rest gestures are manually custom made by us using the gesture description provided by fingerpose.

We have created a sign language gesture- ‘i love you’, which can be very useful for people who can not speak.
For creating this we have given curl and directions of all 5 fingers, which can be found in the github repository- here.

Main substeps in this module are:

- Installing fingerpose - npm install fingerpose
- Importing emojis and fingerpose
- Importing the images
- Updating the above written detect function for gesture handling
- Adding emoji image to the display screen

3. **Deployment:**
   We have deployed our project into github using ‘gh-pages’. The detailed steps for deployment can be found here.
   And our deployed project can be found here.

4. **Smart Reactions in Zoom meetings:**
   For this, we have used an open source software called “Open Broadcaster Software.
   Firstly, we installed the OBS form here and Zoom meetings app from here.

   After that, we can open our deployed project. Once the project is running smoothly we can create a window capture in OBS and capture the browser window in which the react app is running. Once our scene in OBS is ready we can enable the ‘virtual camera’ option in OBS. Now we just have to create a new meeting in the zoom app and select - ‘virtual camera’ as an input for the video.

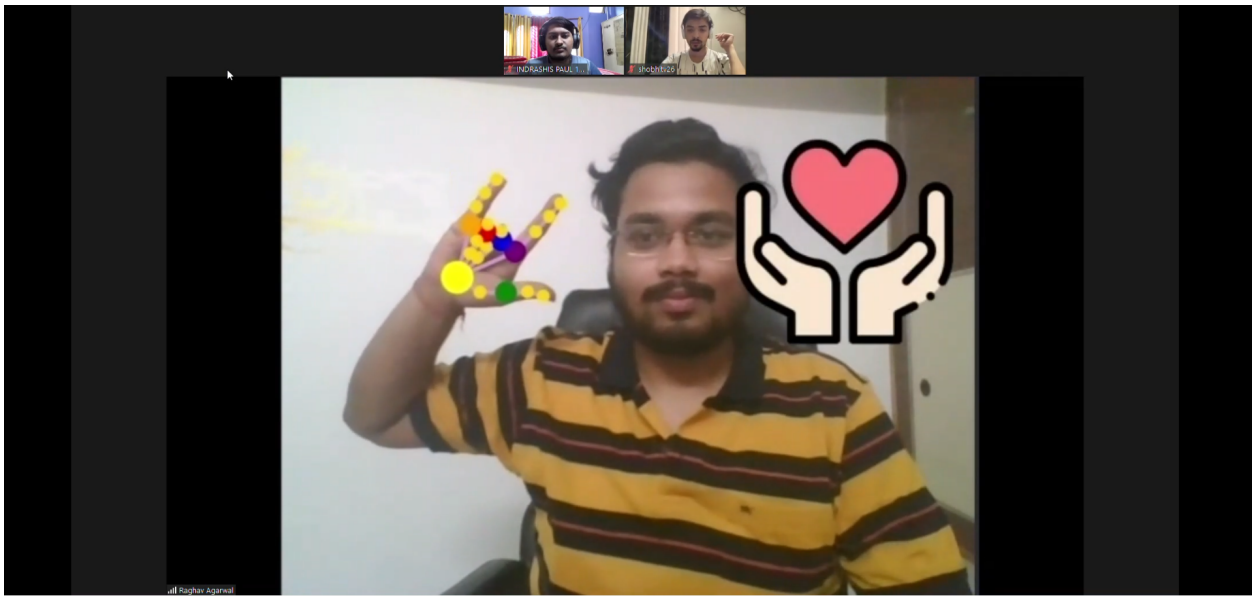
IMPLEMENTATION AND CODING

Entire implementation can be found in our github repository, which can be found here.

DEMO VIDEO

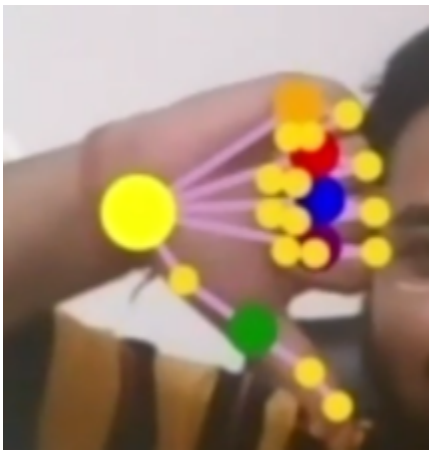
The demo video of our project can be found here.

SNAPSHOT OF THE PROJECT



[Fig. 6 - Snapshot of Project]

RESULT AND DISCUSSION



INPUT **OUTPUT**

With input as our hand gestures, we are getting the mapped emoji alongside the video which will serve the problem statement of this project. Also, We can add more number of custom gestures in the app by which people with speaking and hearing disabilities can easily communicate via our app in live meetings.

CONCLUSION

This project can be used by anyone who wants a hassle free small communication in virtual meets. But, this is especially a boon for the people who can't speak or can not listen. They can use this feature as a sign language detection app. For now we have added only one custom made sign language hand gesture and that is - 'i love you'. We can easily add more in the future, which is our scope of improvement.

REFERENCES

1. https://support.zoom.us/hc/en-us/articles/200912909-Getting-Started-With-Group-Messaging
2. https://docs.microsoft.com/en-us/microsoftteams/raise-hand-meetings
3. https://support.google.com/meet/answer/10159750?co=GENIE.Platform%3DDesktop&hl=en
4. https://techcommunity.microsoft.com/t5/microsoft-teams-public-preview/now-in-public-preview-meeting-reactions/m-p/1978451
5. https://support.zoom.us/hc/en-us/articles/360053176572-New-updates-for-December-21-2020
6. https://support.zoom.us/hc/en-us/articles/201361953-New-updates-for-Windows
7. https://obsproject.com/
8. https://github.com/tensorflow/tfjs-models/tree/master/handpose
9. andypotato/fingerpose: Finger pose classifier for hand landmarks detected by TensorFlow.js handpose model (github.com)