VISION ARCADIA

PROBLEM STATEMENT

PROBLEM STATEMENT

- Develop a Computer vision model that can recognize hand gestures.
  - Collect data (Images).
  - Clean and Standardize the data.
  - Develop an input pipeline.
  - Develop and train the model.
  - Test the model for performance
- Automate tasks using these gestures
  - Begin with automating tasks within your computer
  - Deploy this model to raspberry pi and automate tasks using the microcontroller

DATA COLLECTION

```
PS E:\Python projects\vision arcadia> python .\collectGesture.py
Enter the folder name to save the images: Stone
[ WARN:1] global C:\Users\runneradmin\AppData\Local\Temp\pip-req-bu
mespace'::SourceReaderCB::~SourceReaderCB terminating async callbac
Created folder: Stone
PS E:\Python projects\vision arcadia> q
```

- Use the script that we have provided you to collect data.
  - Execute the following command without quotes:
    - “python .\collectGesture.py”
    - <FOLDER NAME>
  - The camera will start recording and to end collection press ”q“ or kill the program

DATA CLEANING AND STANDARDIZATION

- Make sure there are no noisy images in your data that could affect learning.
  - For Example:
    - Dimly lit images.
    - Blurry/Shaky Images.
    - Images with wrong gesture.
  - Another good step would be to Make sure all images are of same dimensions, preferably training dimension to save time down the line

SOME TIPS WHILE COLLECTING THE DATA

- For a particular gesture try collecting it from multiple angles across different backgrounds and multiple people to induce a sense of generalization across the data.
- Make sure you avoid blurry and shaky images and images captured in very low light or very high contrast situations.
- Try collecting your images in a sense where you will be practically using them.
- Most convolutional neural networks usually work with input dimensions that range from 48x48 to 256x256 so try modifying the data collection script to standardize your images while they are being collected.

INPUT PIPELINE

- We recommend using TensorFlow APIs to develop your input pipelines. They will automatically allocate memory, create batches and apply any augmentations to your images while they are being passed to the model.
- Follow this link for more information https://www.tensorflow.org/guide/data#decoding_image_data_and_resizing_it

DEVELOP, TRAIN AND TEST THE MODEL

- Identify your task and the appropriate architecture you want to use.
- Now train your model using one of the two ways:
  - Train model from Scratch.
  - Use transfer learning.
- Make sure you develop proper methods to test model performance (i.e. testing and validation sets/ training curves etc.)

AUTOMATE

- Begin with automating tasks on your computer. These can include
  - Music Control
  - Window/tab switch
  - Controlling your function keys ie. Sleep, fan speed brightness etc.
- Using the same model, try to automate some tasks using raspberry pi.
- This is an open ended problem and you are free to innovate.

EVALUATION FOCUS AREAS

- Quality of your collected data
  - Number of gestures.
  - Innovative augmentations etc.
- Model Quality
  - Innovative loss functions.
  - Good accuracy and loss scores.
- Innovation in Automation
  - Did you automate something very innovative?