GOVERNMENT COLLEGE OF TECHNOLOGY,CBE
DEPARTMENT OF ECE

Design & Development of an Intelligent Obstruction Management System for Rover

GUIDED BY:
Dr. O. SARANIYA, Prof.[CAS] ,ECE

PRESENTED BY:

RAJARATHINAM E - 71772114133

SHYAM M - 71772114139

SURESH KUMAR S - 71772114147

VAIRAVEL S - 71772114148

PROBLEM STATEMENT

fig.a.perseverance rover

fig.b.stone lodged in front left wheel

Design & Development of an Intelligent Obstruction Management System for Rover integrates real-time detection and removal of obstructions lodged in the wheels, along with monitoring for the wheels and robotic arm, optimizing the rover's efficiency and performance in Martian conditions.

- NASA launched the **Perseverance rover**(fig.a) in July 2020, which successfully landed on Mars in February 2021 to collect samples for scientific research

.

- During its mission, **a rock(stone) lodged in one of its wheels**(fig.b) and remained there **for 437 days**, causing minor damage like **cracks, wear & tear.**

why do we need to address this problem?

- **The rover's wheels** are designed to **rotate 360 degrees**, but a heavy lodged rock can obstruct this movement, potentially leading to more serious issues.
- While no major damage has been reported yet, there remains a significant risk of **structural or component failure**.
- If even a **single wheel** is severely **damaged**, the rover could **lose stability,** ultimately compromising the entire mission.

PROPOSED SOLUTION & OBJECTIVE

- ✓ **Rover's wheel health checkup** is conducted after specific missions, often following traversal of challenging terrains, using **onboard cameras**.
- ✓If any **stone**(obstruction) **lodged** in the wheel, it is **identified** during the wheel health checkup through these cameras.
- ✓The **captured images** are analysed by **comparing** them with a pre-existing **dataset** to identify any **anomalies.**
- ✓Once the presence of a stone is confirmed, a **specialized robotic arm** integrated with the rover is activated to **remove** the **stone from** the **wheel** & deposit it on the ground.

NOTE: The robotic arm movements are predefined for each wheel on the rover.

How it addresses the problem ?

- ✓By **swiftly detecting obstructions**, we can prevent further damage to the rover's wheels caused by the obstruction(stone).

OBJECTIVE

- ✓ **"To enhance the rover's durability and performance by integrating a preventive structure, such as a robotic arm, to safeguard against damage from stones lodged in the wheels, ensuring seamless and reliable operation"**

LITERATURE SURVEY

CURIOSITY ROVER’S WHEEL

fig.g

- The wheels(fig.g) are designed with gaps for ejecting sand and small rocks, and sharp strips to minimize crack expansion.
- However, despite these features, the harsh Martian terrain still causes significant damage over time.

fig.h.damaged wheel

PERSEVERANCE ROVER’S WHEEL

fig.i

- NASA enhanced the Perseverance rover's wheel(fig.i) design by increasing the thickness and closing the gaps present in Curiosity's wheels to reduce damage from Mars' terrain.
- However, even with these improvements, a "pet rock“ got lodged in the wheel(fig.j) and stayed with the rover for nearly 439 days.

fig.j

CLOSED WHEEL STRUCTURE

fig.k.one side closed wheel

- The Opportunity rover used a wheel(fig.k) with one side closed. While this design may help address stone lodging, it has drawbacks. Crack expansion could occur, leading to complete wheel failure.
- Additionally, stones could still get lodged in the open side, making this design unsuitable for solving the lodging problem effectively.

CURRENT RESEARCH

fig.l.memory shape alloy wheel

- To address the challenges with rover wheels, NASA explored alternatives such as memory shape alloy wheels(fig.l) and legged rover(fig.m).
- Memory shape alloy wheels, while promising, cannot support heavy loads. Legged rovers, though offering improved terrain handling, have high power consumption and inefficient performance.
- Thus, these alternatives remain under research and have not yet been implemented on Mars or the Moon.

fig.m.legged rover

METHODOLOGY

IMAGE CAPTURE:
The **Raspberry Pi 5MP Camera** captures real-time images of the rover's wheels and sends them to the **Raspberry Pi 3 Model B.**

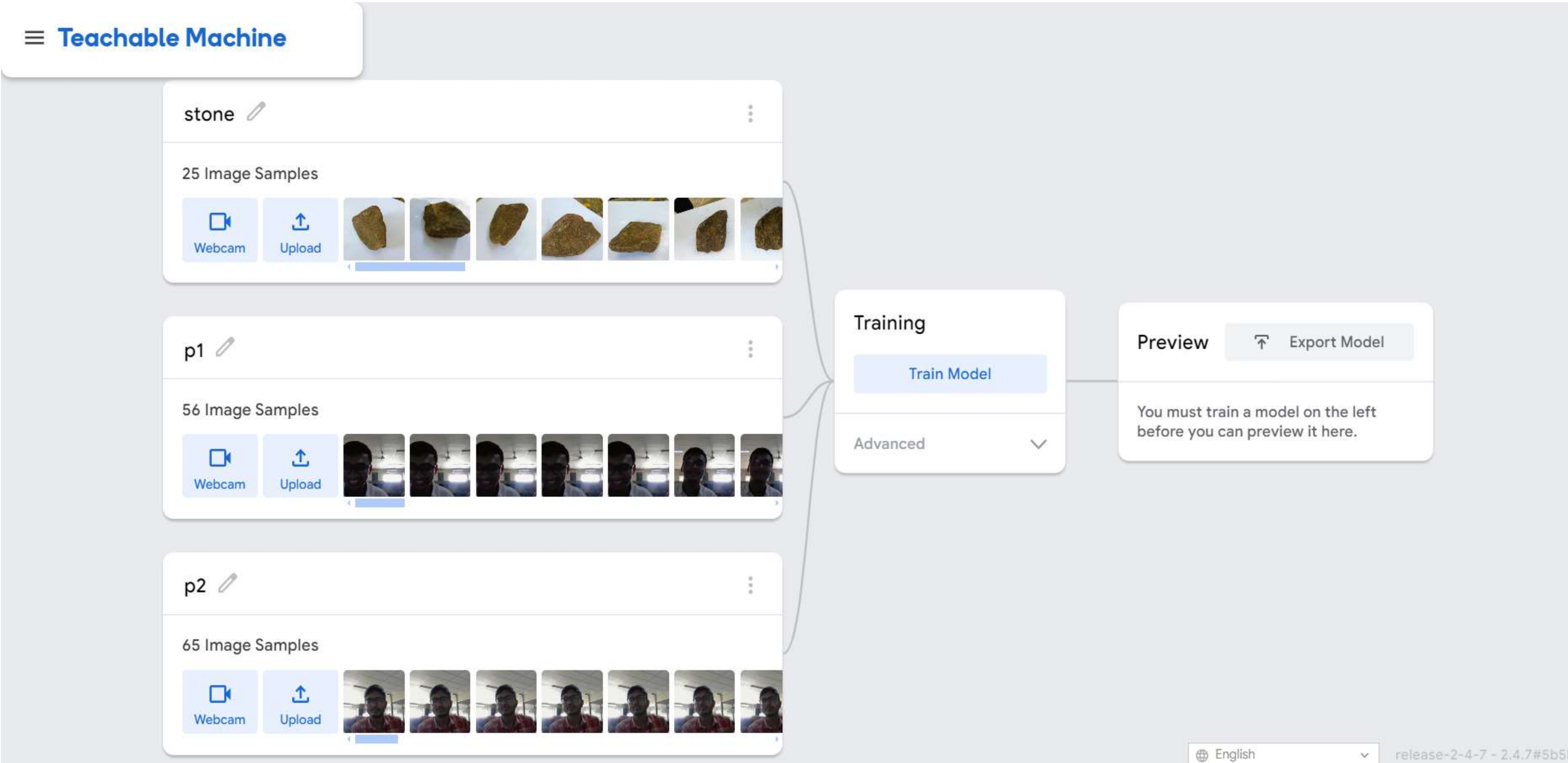
IMAGE PROCESSING:
OpenCV processes the images to enhance quality and extract features like Stones or debris.

OBSTRUCTION DETECTION:
The processed images are fed into a **CNN**, built using **TensorFlow & Keras**, which classifies whether an obstruction(Stone) is present or not..

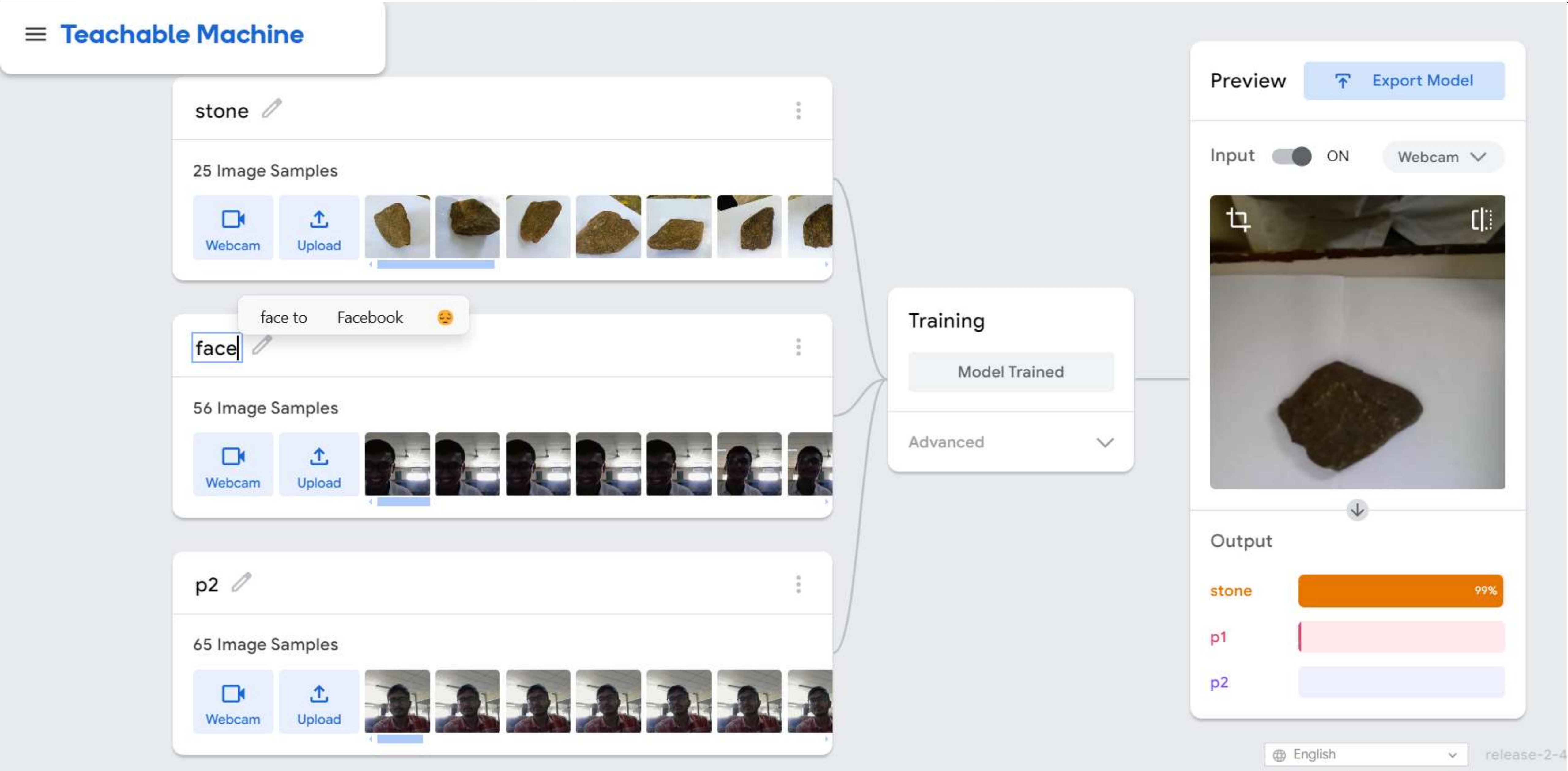
OBSTRUCTION LOCATION & SIZE:
If an obstruction is detected, the Raspberry Pi 3 Model B processes the **CNN output to calculate the location and size** of the obstruction based on the image data.

ROBOTIC ARM ACTIVATION:
Using the calculated location and size, the Raspberry Pi 3 Model B sends commands the MG 995 servo motors to activate the **robotic arm** to move to the obstruction's location and remove it.

TEACHABLE MACHINE (TM)



INPUT : IMAGE SAMPLES



TM - BRIEF

Teachable Machine is a cloud-based machine learning tool that simplifies the process of creating, training, and deploying custom machine learning models. It operates through a web interface, making advanced machine learning accessible without requiring programming knowledge.

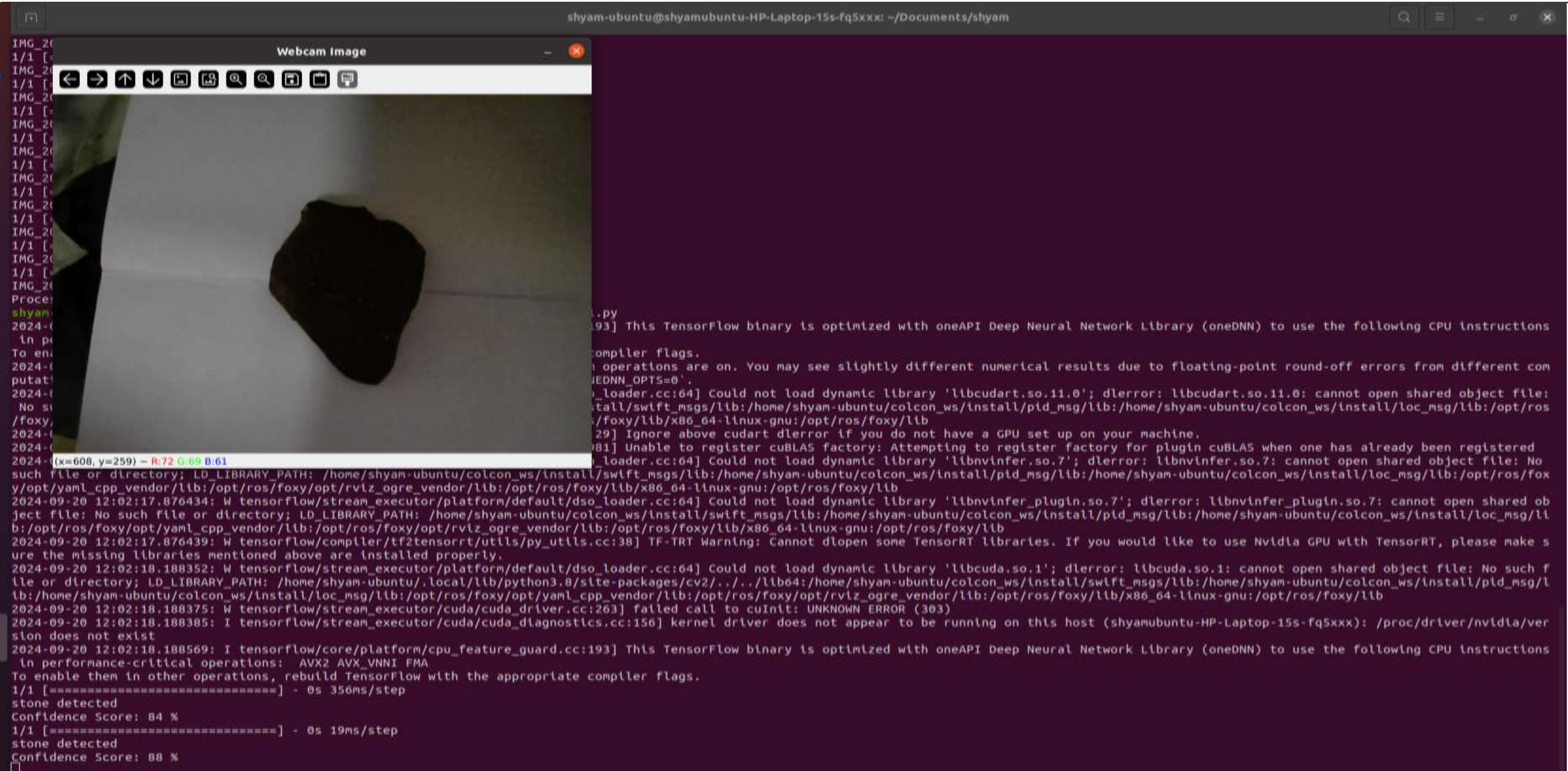
MORPHOLOGICAL FEATURES

- Shape
- Edges and Contours
- Texture
- Colour
- Spatial Relationships

ALGORITHM APPLIED

- Convolution Neural Networks(CNN)

RASPI CAM


shyam-ubuntu@shyamubuntu-HP-Laptop-15s-fq5xxx: ~/Documents/shyam
Webcam Image
(x=608, y=259) ~ R:72 G:69 B:61
193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions
compiler flags.
operations are on. You may see slightly different numerical results due to floating-point round-off errors from different com
_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror: libcudart.so.11.0: cannot open shared object file:
29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
981] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered
_loader.cc:64] Could not load dynamic library 'libnvinfer.so.7'; dlerror: libnvinfer.so.7: cannot open shared object file: No
such file or directory; LD_LIBRARY_PATH: /home/shyam-ubuntu/colcon_ws/install/swift_msgs/lib:/home/shyam-ubuntu/colcon_ws/install/pid_msg/lib:/home/shyam-ubuntu/colcon_ws/install/loc_msg/lib:/opt/ros/foxy/opt/yaml_cpp_vendor/lib:/opt/ros/foxy/opt/rviz_ogre_vendor/lib:/opt/ros/foxy/lib/x86_64-linux-gnu:/opt/ros/foxy/lib
2024-09-20 12:02:17.876434: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libnvinfer_plugin.so.7'; dlerror: libnvinfer_plugin.so.7: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /home/shyam-ubuntu/colcon_ws/install/swift_msgs/lib:/home/shyam-ubuntu/colcon_ws/install/pid_msg/lib:/home/shyam-ubuntu/colcon_ws/install/loc_msg/lib:/opt/ros/foxy/opt/yaml_cpp_vendor/lib:/opt/ros/foxy/opt/rviz_ogre_vendor/lib:/opt/ros/foxy/lib/x86_64-linux-gnu:/opt/ros/foxy/lib
2024-09-20 12:02:17.876439: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Cannot dlopen some TensorRT libraries. If you would like to use Nvidia GPU with TensorRT, please make sure the missing libraries mentioned above are installed properly.
2024-09-20 12:02:18.188352: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /home/shyam-ubuntu/.local/lib/python3.8/site-packages/cv2/../../lib64:/home/shyam-ubuntu/colcon_ws/install/swift_msgs/lib:/home/shyam-ubuntu/colcon_ws/install/pid_msg/lib:/home/shyam-ubuntu/colcon_ws/install/loc_msg/lib:/opt/ros/foxy/opt/yaml_cpp_vendor/lib:/opt/ros/foxy/opt/rviz_ogre_vendor/lib:/opt/ros/foxy/lib/x86_64-linux-gnu:/opt/ros/foxy/lib
2024-09-20 12:02:18.188375: W tensorflow/stream_executor/cuda/cuda_driver.cc:263] failed call to cuInit: UNKNOWN ERROR (303)
2024-09-20 12:02:18.188385: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (shyamubuntu-HP-Laptop-15s-fq5xxx): /proc/driver/nvidia/version does not exist
2024-09-20 12:02:18.188569: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 AVX_VNNI FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
1/1 [==============================] - 0s 356ms/step
stone detected
Confidence Score: 84 %
1/1 [==============================] - 0s 19ms/step
stone detected
Confidence Score: 88 %

ANALYSIS OF TYPES OF STONES IN MARS

- Basalt
- Gabbro
- Volcanic Glass
- Peridotite
- Scoria
- Sandstone
- Mudstone
- Silica
- Magnetite


fig.n.Basalt


fig.o.Gabbro


fig.p.Volcanic Glass


fig.q.Peridotite


fig.r.Scoria

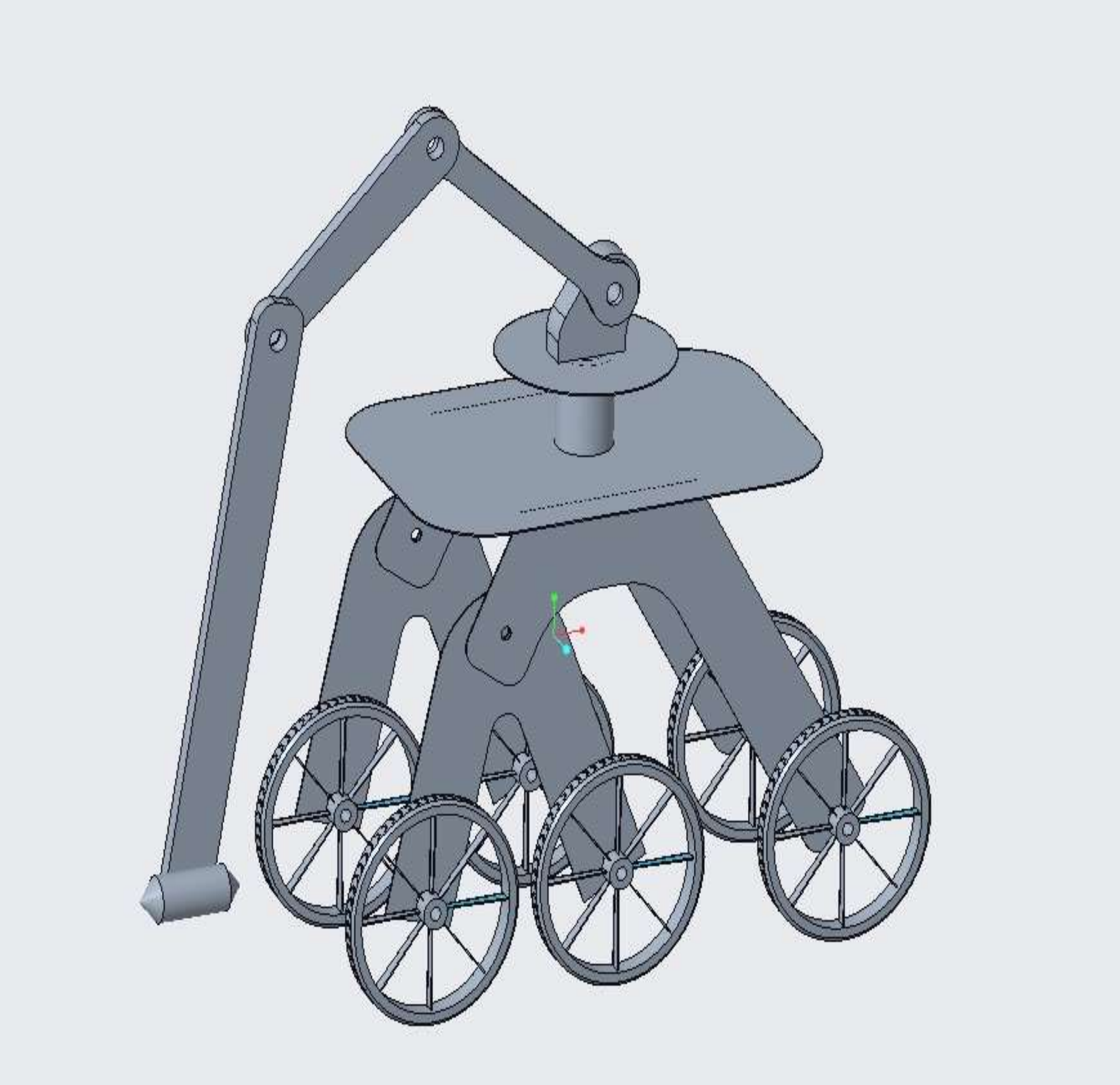

fig.s.Sandstone


fig.t.Mudstone


fig.u.Silica

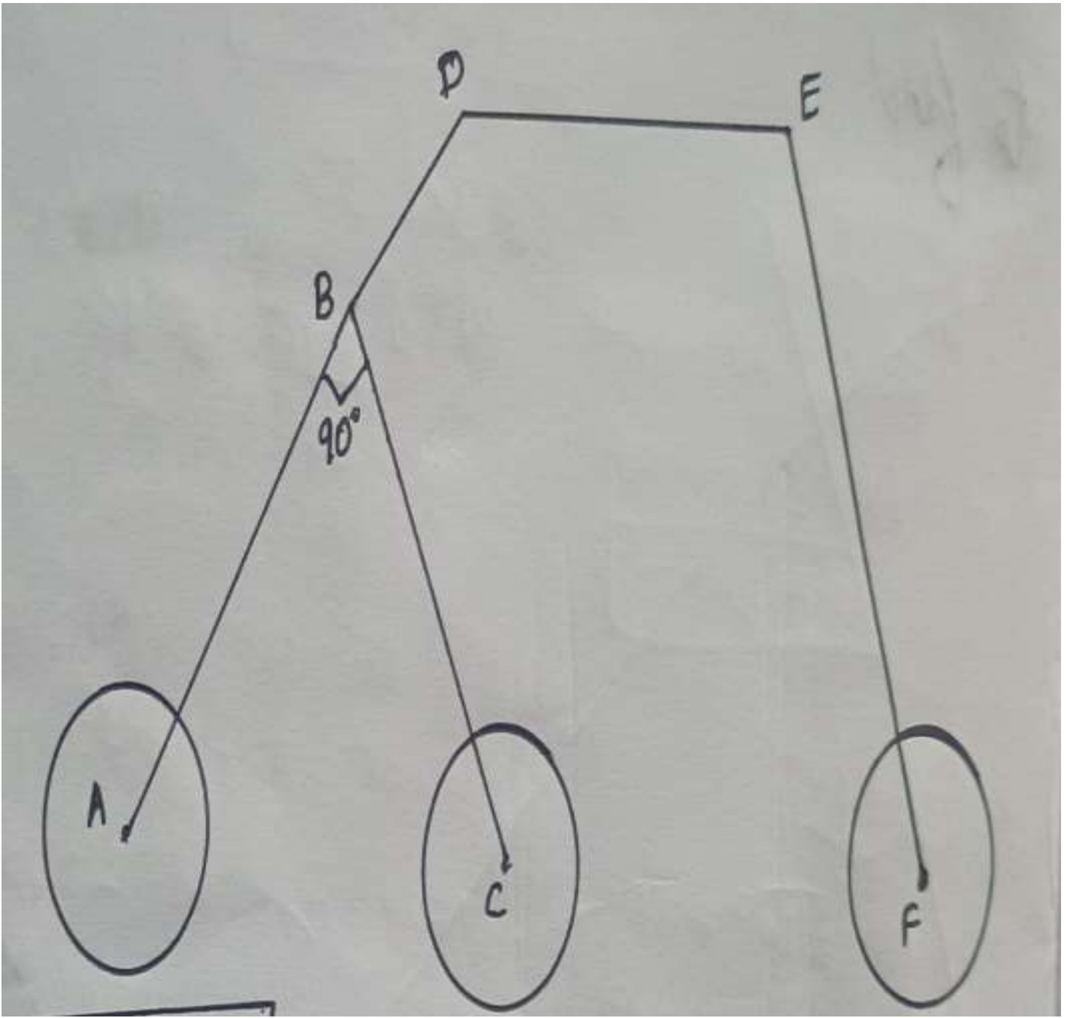

fig.v.Magnetite

DESIGN OF ROCKER - BOGIE



DESIGN OF ROCKER - BOGIE

Let us assume the distance between the wheels of the rover be 200mm, angle of bogie ABC be 90 degree and radius of the wheel be 250mm.



Distance between first and second wheel in bogie,

$$AC = 200\text{mm}$$

As assumed the distance between the wheels be same.so,

$$AC = CF = 200\text{mm}$$

Total distance,

$$AF = AC + CF$$

$$= 200\text{mm} + 200\text{mm} = 400\text{mm}$$

To find the length of the bogie,

let, $\mathbf{AB} = \mathbf{BC} = \mathbf{x}$

DESIGN OF ROCKER - BOGIE

Using Pythagoras theorem,

$$AC^2 = AB^2 + BC^2$$

$$200^2 = x^2 + x^2$$

$$200^2 = 2x^2$$

$$x = 141.55mm$$

Therefore, $AB = BC = 141.55mm$

ROCKER DESIGN

Let us assume the Chassis length be 600mm and the angle of rocker DEF be 120 degree.

To find CD:

Using the Pythagoras theorem,

$$CD^2 = BC^2 + DB^2$$

$$CD^2 = 141.55^2 + 100^2$$

$$CD = 173.26$$

To find DF:

$$DF^2 = CF^2 + CD^2$$

$$DF^2 = 200^2 + 173.26^2$$

$$DF = 264.6mm$$

DESIGN OF ROCKER BOGIE

To find $\mathbf{EF}$:

Chassis length, $\mathrm{ED} = \mathbf{150mm}$

Using law of cosine rule,

$$\mathbf{DF^2} = \mathbf{ED^2} + \mathbf{EF^2} - 2 \text{ x } \mathrm{ED} \text{ x } \mathrm{EF} \text{ x } \cos(\mathrm{DEF})$$

$$-\mathbf{EF^2} = \mathbf{ED^2} - \mathbf{DF^2} - 2 \text{ x } \mathrm{ED} \text{ x } \mathrm{EF} \text{ x } \cos(\mathrm{DEF})$$

$$- \mathbf{EF^2} = \mathbf{600^2} - \mathbf{264.66^2} - 2 \text{ x } 600 \text{ x } \mathrm{EF} \text{ x } \cos(120)$$

$$-\mathbf{EF^2} = \mathbf{289955.0844} - 2 \text{ x } 600 \text{ x } \mathrm{EF} \text{ x } (-0.5)$$

$$-\mathbf{EF^2} = \mathbf{289955.0844} - 600\mathrm{EF}$$

$$\mathrm{EF} = 316.3984\mathrm{mm}$$

Therefore, Linkage length of rocker is $316.3984\mathrm{mm}$.

DESIGN OF ROBOTIC ARM

Horizontal Reach Requirement

- Chassis center to front edge = 600 mm / 2 = **300 mm**
- Rocker-bogie system extends further = AF = **400 mm**
- So, total reach needed from arm (center of chassis) to reach front wheel area = Horizontal reach (H)=300+400=**700mm**

Vertical Reach Requirement

To reach the ground from the chassis:

- **Ground clearance** = 500 mm
- Add height of object (~100 mm) → for safe picking

Vertical reach (V)=500+100=600 mm

DESIGN OF ROBOTIC ARM

Total Diagonal Reach

Use Pythagoras theorem to calculate full arm reach:

Total reach **(R)=** $\sqrt{(H^2 + V^2)}$= $\sqrt{(700^2 + 600^2)}$

$$=\sqrt{(490000 + 360000)}$$

$$=\sqrt{850000}$$

R = 922.96mm

But this is just to reach the ground in front. Add extra margin (≈ 30%) for:

- Object pickup further away
- Minor overextension

Design Total Reach = 922.96 + 0.3 × 922.96

= 922.96 + 276.9

= 1199.86 mm **≈ 1200 mm**

Rule of Thumb for Robotic Arm Design:

Let total arm reach ≈ **1200 mm** and break into:

- L1 = 0.28 × 1200 = 336mm ≈ 341 mm
- L2 = 0.25 × 1200 = 300 mm ≈ 306 mm
- L3 = 0.46 × 1200 = 552 mm ≈ 586 mm
- L4 = 0.01 × 1200 = 12 mm ≈ 10 mm

Total Arm Length = L1 + L2 + L3 + L4 =341 + 306 + 586 + 10 = 1243 mm

REQUIREMENTS

TECHNOLOGY STACK

- **Fusion360,Solidworks** – Rover &Robotic Arm Design
- **OpenCV** - Image Processing
- **TensorFlow & Keras** - Deep Learning Frameworks
- **Convolutional Neural Networks (CNN)** - Obstruction Recognition Algorithm
- **Python Programming** - for Streamlined and Simplified Development

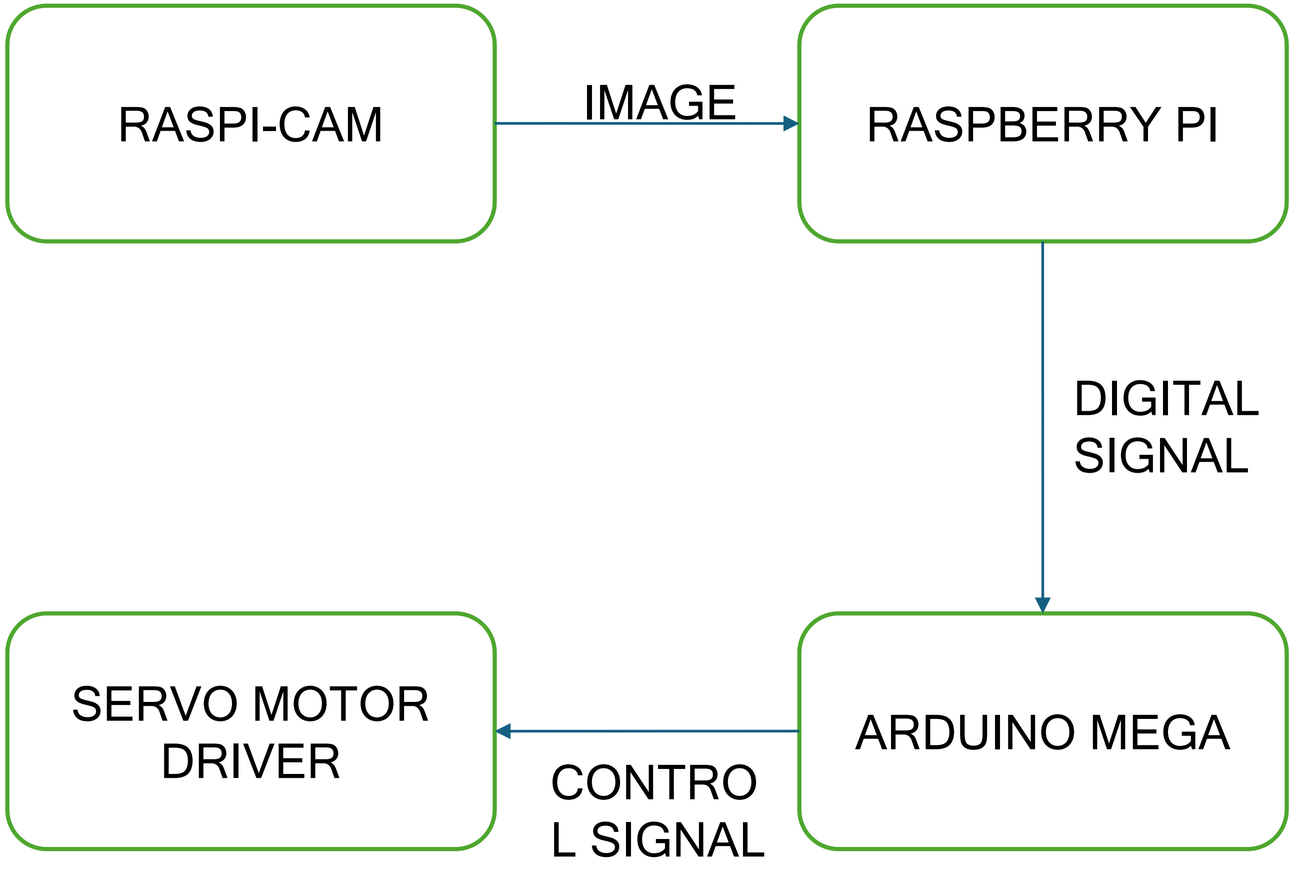
HARDWARE REQUIREMENT

- **Raspberry Pi 3 Model B** - Computing and Control Unit
- **Arduino Mega 2586 -** servo motor control
- **Raspberry Pi 5MP Camera Module** - Image Capture Device
- **MG 995 Servo Motors** - Precision Movement Actuators
- **PCA9685** - Servo motor Driver
- **PVC Pipe –** Robotic arm realization
- **Plywood** - Rocker – Bogie Mechanism

Structural Designing
↓
Structural Realization
↓
Image Processing
↓
Robotic Operation

STRUCTURE REALIZATION

PROTOTYPE


RASPI-CAM
IMAGE
RASPBERRY PI
DIGITAL SIGNAL
ARDUINO MEGA
CONTRO L SIGNAL
SERVO MOTOR DRIVER

WORKING

- ✓ **Capture Image:**

  Raspberry Pi Camera captures an image of the stone.

- ✓ **Image Recognition:**

  Raspberry Pi compares the captured image with a dataset to detect the stone.

- ✓ **Send Signal to Arduino:**

  Once the stone is detected, Raspberry Pi sends a digital signal to the Arduino Mega.

- ✓ **Control Servo Motors:**

  Arduino Mega processes the signal and controls a 16-bit servo motor driver.

- ✓ **Execute Servo Motion:**

  Three MG995 servos (Channels 0, 1, and 2) perform the defined motion patterns:

  Channel 0: 90° → 0°, 0° → 90°

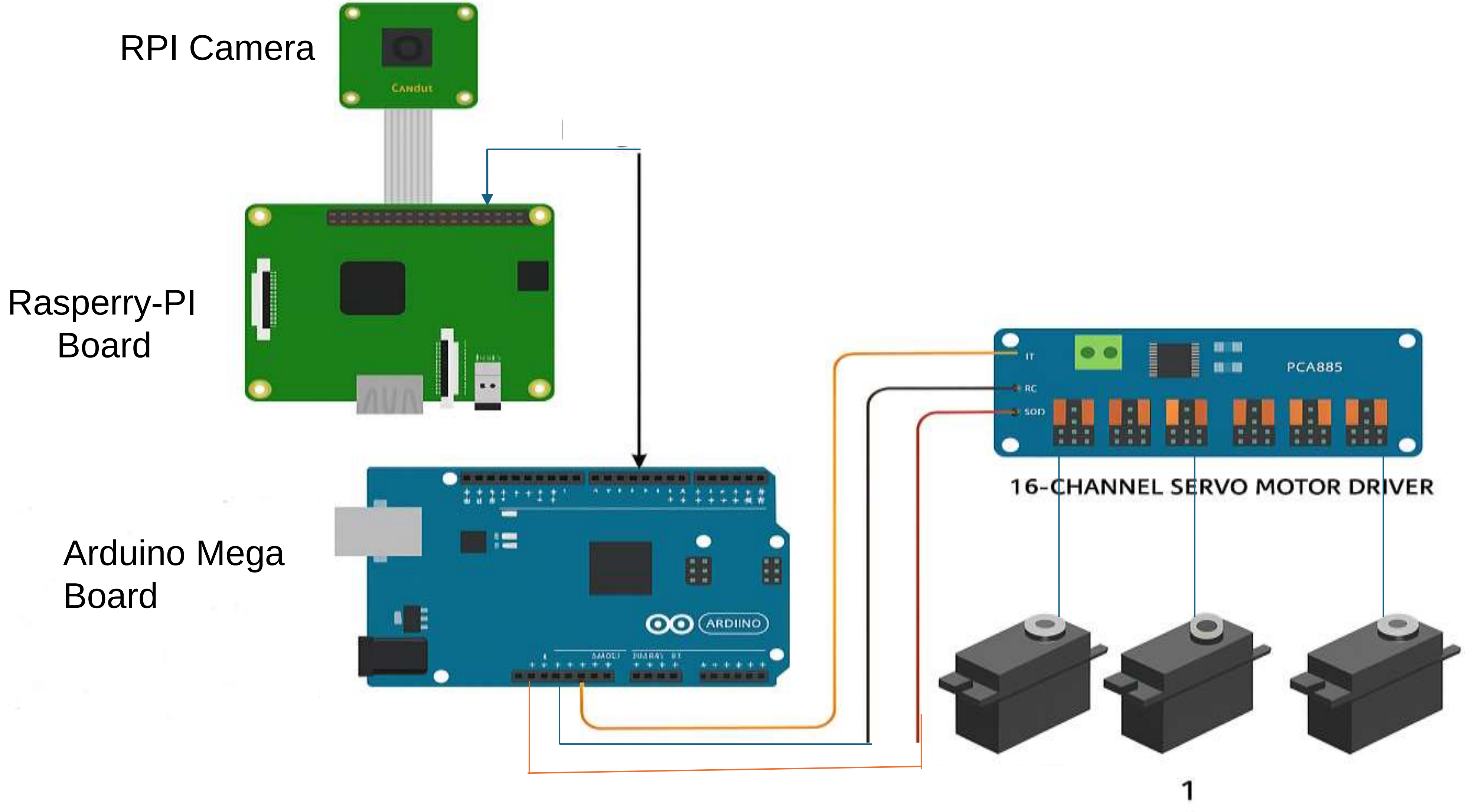
  Channel 1: 0° → 90°, 90° → 0°

  Channel 2: 0° → 90°, 90° → 0° → 90°

- ✓ **Eject Stone:**

  Servos eject the stone from the detected area.

SCHEMATIC DIAGRAM



REFERENCES

[1]Q. Qiao, "Image Processing Technology Based on Machine Learning," in IEEE Consumer Electronics Magazine, vol. 13, no. 4, pp. 90-99, July 2024

[2]K. Kruthika, B. M. Kiran Kumar and S. Lakshminarayanan, "Design and development of a robotic arm," 2016 International Conference on Circuits, Controls, Communications and Computing (I4C), Bangalore, India, 2016

[3]A. Bouton, C. Grand and F. Benamar, "Design and Control of a Compliant Wheel-on-Leg Rover Which Conforms to Uneven Terrain," in IEEE/ASME Transactions on Mechatronics, vol. 25, no. 5, pp. 2354-2363, Oct. 2020

[4] Nithesh Naik, CS Suhas Kowshik, Dashratharaj K Shetty, Rajesh Gopakumar and Namesh Malarout, Design and Analysis of Ugv with Rocker Bogie Mechanism for Landmine Extraction., International Journal of Mechanical Engineering and Technology, 10(01), 2019, pp.1731–1744

[5] https://science.nasa.gov/mission/mars-2020-perseverance/

THANK YOU!