

**e-Yantra Robotics Competition - 2018**

**Task 3C: Theme and Implementation Analysis HC#4674**

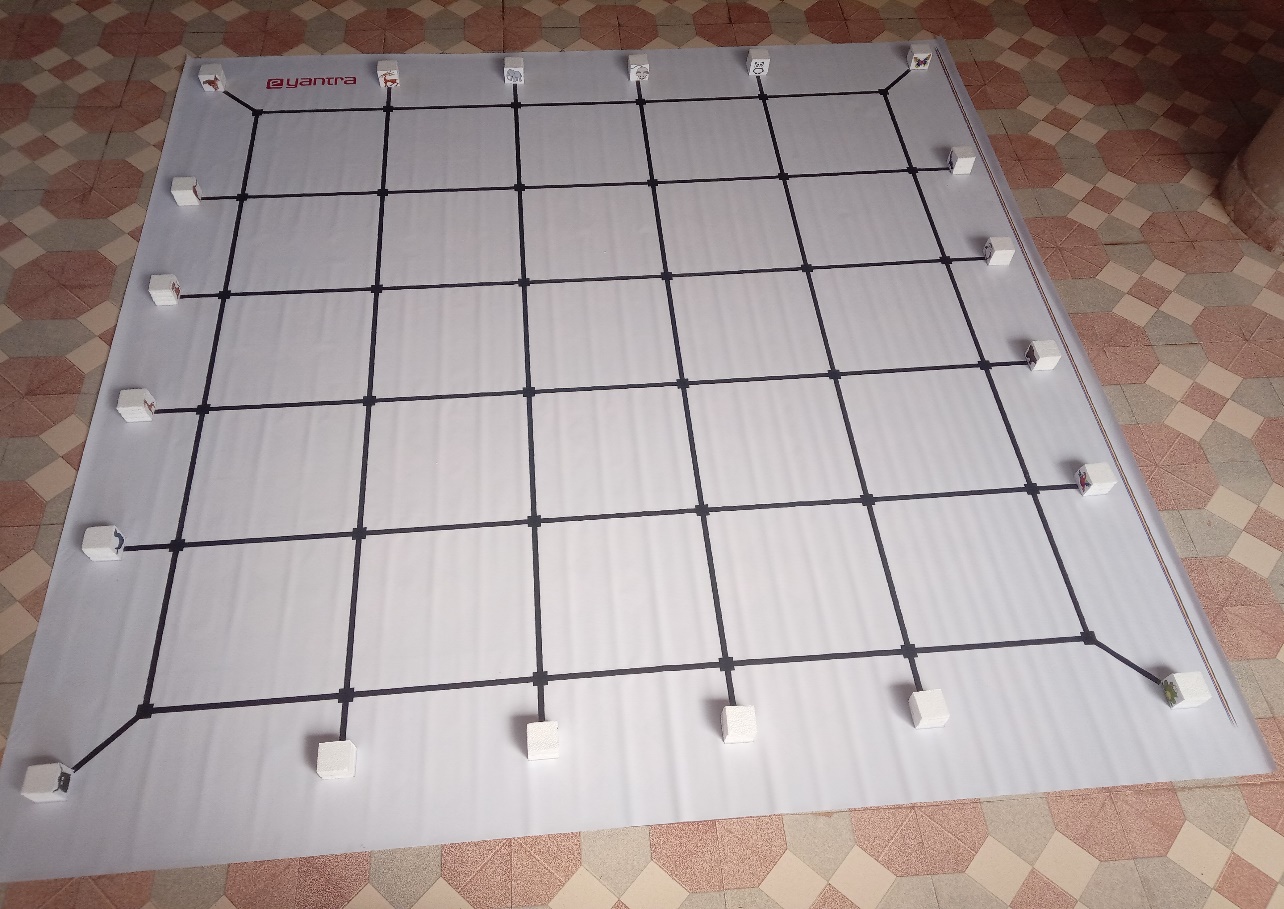
**Scope and Preparing the Arena**

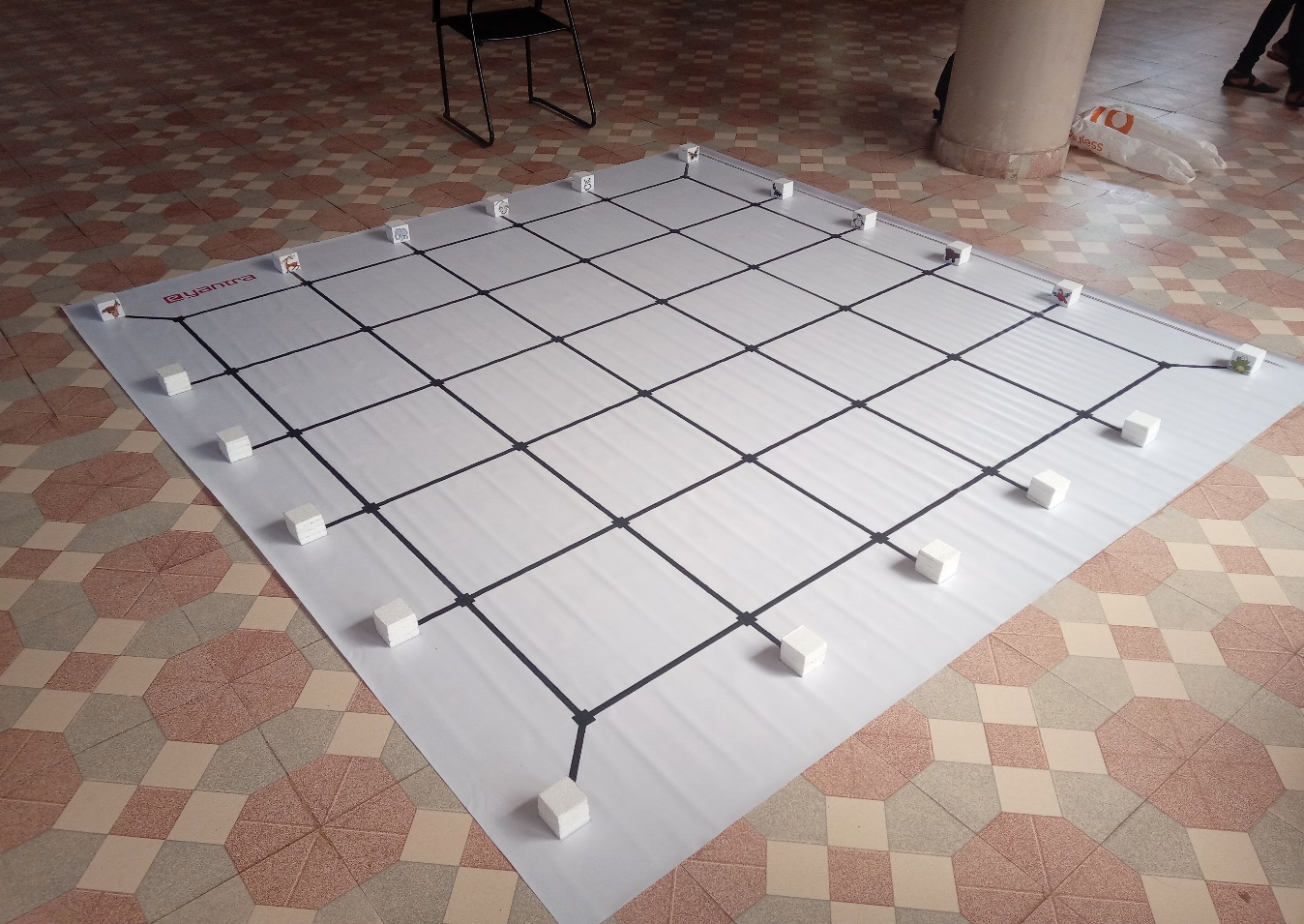
**Q1 a. State the scope of the theme assigned to you. (2)**

India is a home to a number of varied species of flora and fauna. The different habitats include Oceans, Wetlands, Forest, Grasslands, Deserts, Mountains etc. After visiting a plethora of fauna in our Jungle Safari we find different types of animals and their natural habitats.

The theme includes building and training a robot to negotiate a path on the arena, which is an abstraction of eco system in grid form, and visit animals and their habitats. Identification is done by algorithms based on Machine Learning and path traversal using sensors. The identified animals have to be picked and placed in their respective habitats.

1. **Attach the Final Arena Images. (3)**







**Building Modules**

**Q2. Identify the major components required for designing the robotic system for the solution of the theme assigned to you. (3)**

ELECTRONIC COMPONENTS:

1. Sensors:

* Sharp IR Range Sensors: When an obstacle comes in front of the robot it emits Infrared light emitted from the sensor strikes the object and reflects back. It is used to give the distance of the obstacle in front of the robot in the range of 20cm.
* IR Proximity Sensor: It is used for obstacle detection and distance measurement within the range of 10cm.
* White Line Sensor: It is used
* Position Encoder: It is used to measure the distance the robot has moved or the angle in which it is rotated.

1. Actuators:

* 60 RPM DC Motor: Attached to the wheels for its motion.
* Servo Motor: it is used to add an arm or triple structure to the platform for Pick and place animals in the given habitant

1. Communication:

* RS 232 Serial Cable
* USB cable for communication between robot and system, also used to load the .hex file to the flash memory.

4. Output Devices

* Buzzer to indicate start and Stop of process
* LCD display to indicate sensor values

**Design Analysis**

**Q3. Teams have to design an arm mechanism for deposition of the Animals in their respective Habitats.**

1. **Choose adoption to position this mechanism on the robot and why? (2)**
2. **Front 2. Back 3. Right/Left**

**Answer: \_ Front\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

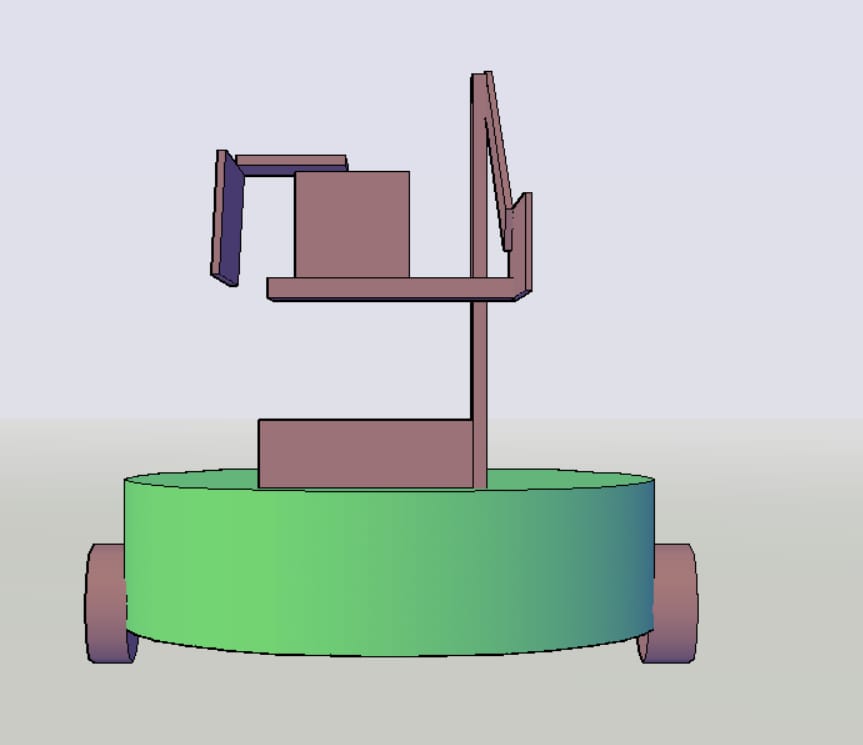
Since we can sense an obstacle (animal) with proper distance using sharp sensor given at the front of a robot and can precisely direct robotic arm to pick the animal, also it helps us in navigation of robot with uniform and minimum turns.

1. **Explain the design of the mechanism and how it is mounted on the robot. Also explain what challenge/s do you expect to face and how you will overcome them? (6)**

In order to design a robotic arm for the theme we will prefer thick cardboard or fly wood materials. As per the theme we need to pick an animal from the specified node and place it in specified habitant from the machine learning algorithm. Hence it requires two motions in the robotic arm such as vertical motion of the arm for adjusting its height and horizontal motion to grab the animal in robotic hand. Since we were provided with two Servo motor we are using it for this purpose.

First servo motor will be fixed at top of the robot and its rotor will be connected to the robotic arm vertical moving section and second servo motor will be fixed at arm and its rotor connected to horizontal moving section as shown in figure1. Both the Servo motors are connected to the robot in the servo pod1 and servo pod2 connector pins and programmed as per the movement needed.

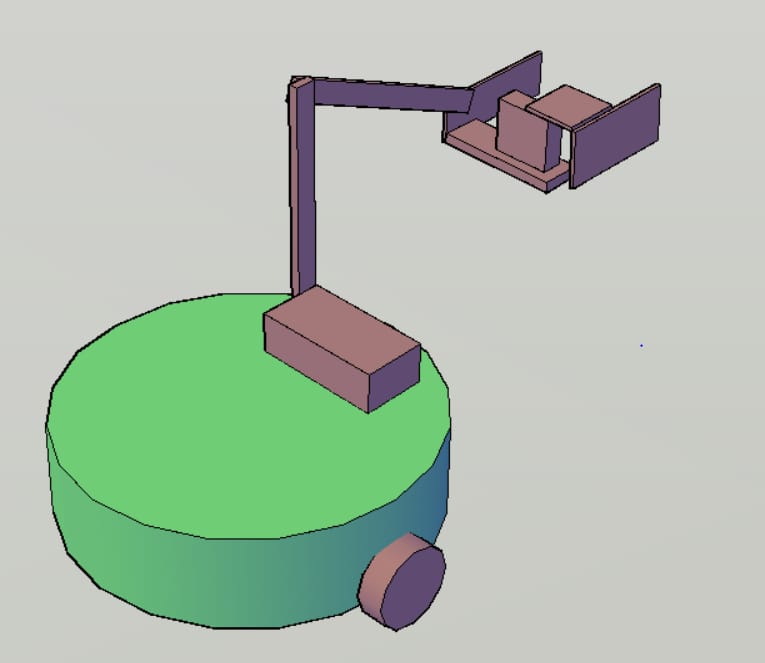
The entire structure of robotic arm top view is as shown in figure2.

****

First Servo

Second Servo

**Figure1: Front View of Robotic Arm**

****

**Figure2: Top view of the robotic Arm**

Different challenges which we can expect in the mechanism and how its tackled

* Strength of the robotic horizontal mount to hold the servo motor for horizontal motion. This problem can be solved by using thick card board or fly wood materials.
* Robotic arm may block the communication of Sharp sensor for obstacle or animal detection. This problem can be solved by keeping robotic arm default position as shown in figure1 if sharp sensor detects animal then the arm will be brought down to pick the animals.
* Third challenge we may encounter like to decide angle of rotation for both the servo motor. This we can solve by trial and error method and it depends upon arm grip which we can obtain to grab the animal. Also hold it without dropping in wrong deposition area.

1. **choose the actuator/s you will use to design the mechanism. (2)**
2. **DC-Motor 2. Servo Motor 3. Stepper Motor 4. Others**

**Answer: \_\_\_\_two Servo Motor \_\_\_\_\_\_\_\_\_\_\_\_\_**

Since robotic arm needs angular motion, it requires two motion upward/downward direction and to hold/release. In both the cases servo motor provides precise angular rotation to pick and place animals.

If DC motor is used then precise degree of rotation we cannot obtain as compared with Servo motor and steeper motor precise velocity cannot be obtained as compared with Servo motors. Hence we choose Servo motor for design mechanism of arm in a robot.

**Environment Sensing**

**Q4. Explain how you will use the provided sensors to implement the theme. (2)**

* Sharp IR Range Sensors: When an obstacle (animal) comes in front of the robot it senses and provides a clue that animal is at proper distance, robotic arm can pick the animal
* White Line Sensor: It is used for sensing black lines and nodes provided in the arena for navigate the robot proper path to complete the task.
* Position Encoder: It is used to measure the distance the robot has moved or the angle in which it is rotated. So precise rotation or motion can be given for pick and place the animals to its proper habitants.

**Testing your knowledge (theme analysis and rulebook-related)**

**Q5. What is the purpose of “activation function” in a Neural Network? How do different type of activation functions affect the network? (5)**

Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to introduce non-linearity into the output of the neuron.

* Linear Function: If all the layers are linear in nature, the final activation function of last layer is nothing but just a linear function of the input of first layer. It is just used at one place i.e.; output layer.
* Sigmoid Function: usually used in output layer of a binary classification, where result is either 0 or 1, as value for sigmoid function lies between 0 and 1 only so, result can be predicted easily to be 1. If value is greater than 0.5 and 0 otherwise.
* Tanh Function: Used in hidden layers and helps in centring the data by bringing mean close to 0. This makes learning for the next layer much easier.
* RELU: It is less computationally expensive than tanh and sigmoid because it involves simple mathematical operations. At a time only a few neurons are activated making the network sparse making it efficient and easy for computation.
* SoftMax Function: Usually used when trying to handle multiple classes.
* The SoftMax function would squeeze the outputs for each class between 0 and 1 and would also divide by the sum of the outputs**.**

**Q6. What are the different hyper-parameters in a CNN that affect its performance? Explain the different parameters and their effects (in bulleted form). (5)**

1. Window size.

2. Stride.

3. Pooling layers.

4. Number of Hidden Layers and units.

5. Network Weight Initialization.

6. Activation function.

7. Learning Rate.

8. Momentum.

9. Number of epochs.

10.Batch size.

Pooling layer:

* A pooling layer is building block of a CNN.
* Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network.
* Pooling layer operates on each feature map independently. The most common approach used in pooling is max pooling.

Stride:

* The amount by which the filter shifts is the stride.
* Stride is normally set in a way so that the output volume is an integer and not a fraction.
* The stride size in CNN filters not only depend on the properties of pictures in data set, but it is also depend on the way you combine layers together

Window size:

* window size is used in Time Delay Neural Networks and other older neural networks such as NET talk.
* The effect of the window size can be better explained by using an example of reading a text. if you have a sliding window with no overlap then it would be equivalent to not using a time relationship between elements of your training set of character sequences in your text. As you apply an overlap you are applying context of the neigh boring words to the one in the middle of the window.

Number of Hidden Layers and units:

* Hidden layers are the layers between input layer and output layer.
* Many hidden units within a layer with regularization techniques can increase accuracy.
* Smaller number of units may cause underfitting.

Network Weight Initialization:

* Different weight initialization schemes are used according to the activation function used on each layer.
* Mostly uniform distribution is used.

Activation function:

* Activation functions are used to introduce nonlinearity to models.
* Sigmoid is used in the output layer while making binary predictions.
* SoftMax is used in the output layer while making multi-class predictions.

Learning Rate:

* The learning rate defines how quickly a network updates its parameters.
* Low learning rate slows down the learning process but converges smoothly.
* Larger learning rate speeds up the learning but may not converge.

Momentum:

* Momentum helps to know the direction of the next step with the knowledge of the previous steps.
* It helps to prevent oscillations.

Number of epochs:

• Number of epochs is the number of times the whole training data is shown to the network while training.

• Increase the number of epochs until the validation accuracy starts decreasing even when training accuracy is increasing(overfitting).

Batch size:

• Mini batch size is the number of sub samples given to the network after which parameter update happens.

**Q7. Why are the first few layers of a CNN hard to train? (3)**

The initial input layers get raw data with max amount of noise as well as defected data. To extract certain features from the data becomes difficult coz of the noise and other objects with similar feature. For example if the initial convo layer has to detect the outline of a fruit, along with the fruit the input image may contain other objects with similar feature (outline) ,so along with detecting the fruit outline it has to suppress image pixels of other non-important object features. Since the lower layer get input data with useless features already suppressed it is easy to for them to extract their assigned feature. Also while back propagating the error belonging to the initial layer is not properly calculated hence the kernel cannot be properly updated, unlike a linear dense fully connected layer a convo layer contains a lot more no of variables to be updated .

**Algorithm Analysis**

**Q8. Draw a flowchart illustrating the algorithm you propose to use for theme implementation. (12)**

1. **Building Models**

Start

aniPath = Path of animal dataset

habiPath = Path of Habitat Dataset

Extract aniPath into folder animals

Extract habiPath into folder habitats

Create folders train and val in both animals and habitat and extract classes and elements such that elements in folder train is 1.6 times than the elements in folder val

Create Data transforms by normalizing with image net mean and std

Rescale animal dataset to 118\*118 and then to 224\*224

Rescale habitat dataset to 280\*280 then to 224\*224

Create data loader for train and val for both animals and habitat

Use a pretrained resnet18 for both animals and change the last Fc layer

With the respective class numbers of animal dataset and habitat dataset

Create loss function, optimizer for the model

Set hyperparameters such as learning rate, number of epoch and learning momentum

While

i <= no\_of\_epoach

**false**

**true**

Train (animal model and habitat model)

Evaluate loss (animal model and habitat model)

Optimize (animal model and habitat model)

Evaluate (animal model and habitat model)

Save animal model state🡪animal.pth

Save habitat model state 🡪habitat.pth

Stop

1. **Classifying Images using trained Models**

Start

Img = Get the arena image

Detect contours in the image (Img), save them as

Hidden files with filenames as their respective contour location,

Make a list containing the contour detected locations

Use Non-pre trained resnet18 for both animal and habitat model,

Change the last Fc layer to the number of classes of animals and habitat,

Load the previously saved state of the trained model to the animal and habitat model

For animal model and habitat model respectively

If contour detected list has an element

**false**

**true**

Get the hidden image which is named same the list element

Rescale it to 118\*118 and then to 224\*224 if the element is alphanumeric

Rescale it to 280\*280 and then to 224\*224 if the element is numeric

As alphanumeric are animals and numeric are habitat

Pass on the transformed image to the appropriate model based on the element get the prediction number and the class it belongs to base on the Prediction number, save the key as element and class as the value to the resultant dictionary

Return the result dictionary containing contour locations as keys and class names as values

Stop

1. **Flex traversal algorithm**

Start

Notate the starting node of robot to be (0,0)

And the right next to it as (0,1) and so forth till (0,5)

Notate the rest of the node in this matrix manner the

Node above corresponds to (1,0), the next above corresponds to (2,0) and so forth till (5,0)

Each animal location corresponds to a node thus it corresponds the value associated with that specific node and has a direction wrt to node and ‘eyantra’ logo being north

Each habitat location corresponds to 4 nodes thus it corresponds to 4 nodes associated value

Gets the resultant dictionary containing location and class names of both animals and habitat

Associate each animal location to the respective habitat

Based on the inbuilt animal-habitat associativity dictionary

Make a dictionary of animal pick location as key and drop habitat location as value based on the previous matrix notation

Find the minimum distant key from starting node by using [(x1-x2)^2 +(y1-y2)^2]^0.5 and prioritize as 0, and next with 1 and so till there is no more animal habitat associations

Get the key from animal pick-drop dictionary with least associated value given by minimum distant routine and key being the associated node location of the animal, traverse to that specific node using following technique

Move up if x>rx else Move down, where x is the row component of destination node and rx is the row component of robot’s current node

Move right if y>ry else Move left, where y is the column component of the destination node and ry is the column component of robot’s current node

Keep track of the robot’s head direction and initial direction is North at starting point

Pick up the animal object based on direction of animal associated with the node and the current direction of robot, picking the object can be confirmed by IR value to be 0

Compute minimum distant among the 4 node values associated with the animal and select the node having least distance,

Traverse to the selected node using above method, and traverse half distance in between two associated node among 4 nodes place the animal object and traverse to the other half and update the robot current location and direction

Remove the key having least value after placing the object in habitat location

Re-compute the minimum distance to the remaining animal nodes based on current node and update the priority values

Until there is

No key left in animal pick-drop dictionary

**false**

**true**

The generated sequence of motions could be used to auto generate the discrete function script for the robot or

Used to generate sequence numbers that could be fed to the robot which in turn calls specific already fed discrete functions sequentially to achieve the given operation

Stop

**Challenges**

**Q9. What are the major challenges that you can anticipate in addressing this theme and how do you propose to tackle them?**

**(5)**

1. Designing the robotic arm to Pick and Place the animal blocks efficiently.

We focus on mechanism of arm, such a way that the picking that the design suites the cuboid block picking.

2. To obtain the animals and their respective habitant’s location. Also feeding this information to the robot within several minutes.

We’ll come up with the different algorithms and notations to notate the flex nodes in a concise manner, such that robot could find layout easy traversal path

3. Planning about picking up the blocks and placing in proper manner

Several methodologies and logic will be composed in order to pick up the object, determining its direction with respect to the robotic arm head, and the amount of rotation, finally the logic confirming the object lift

4. Image transforms for maximum efficiency in image classification.

Various transformation to the dataset will be done in order to understand the behaviour of convnet as well as model towards classification of the end images.

5. Parking the robots back to home position after the completion of the task.