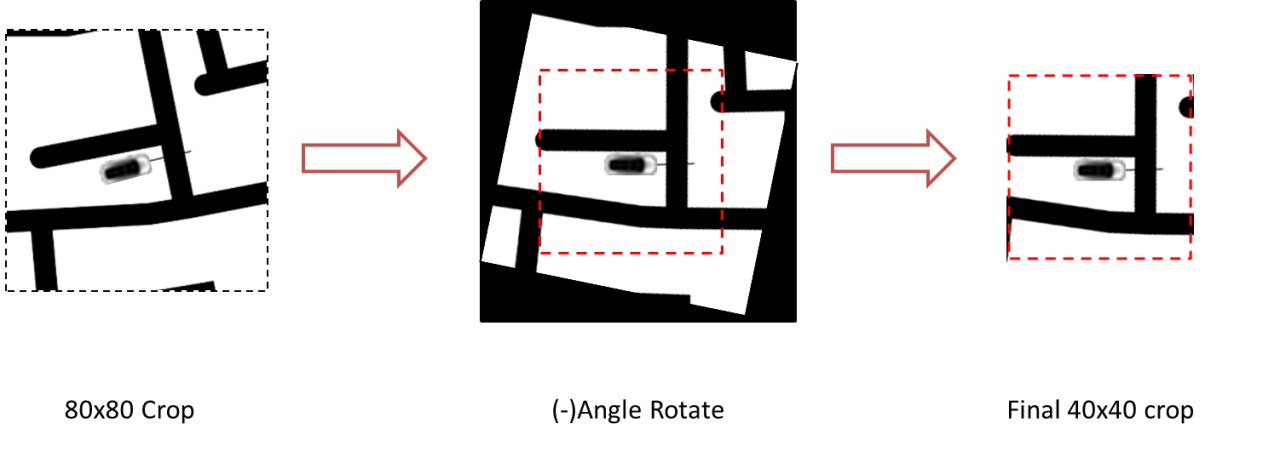
Different Steps Tried for End game

* git rid of sensors and use the car image as the location.

image = sand[int(self.car.x)-20:int(self.car.x)+20, int(self.car.y)-20:int(self.car.y)+20]

* Continuous space
* Passing state to training
* Tried with DQN first
* Went fine to an extent
* Replaced with Td3 kept the same code
* Car circling at one once after completing random rotations in actor’s zone
* Tried by increasing car random time stamps
* Car circling at one once after completing random rotations in actor’s zone
* As car was never reaching destination, added done when reached borders or after reaching certain time steps values so that it could minimum start train
* Tried with different parameters of TD3
* Car circling at one once after completing random rotations in actor’s zone
* Added orientation and distance as another state variable along with image for training
* xx = goal\_x - self.car.x
* yy = goal\_y - self.car.y
* new\_obs\_ori = Vector(\*self.car.velocity).angle((xx,yy))/180.0
* new\_obs\_dis = distance
* current\_state2 = [new\_obs\_ori,-new\_obs\_ori,new\_obs\_dis]
* Modified conv network to take care of one more state ( state1 : image, state 2: orientation and distance)
* Car circling at one once after completing random rotations in actor’s zone
* Padding and cropping the image with different dimensions
* crop\_size =40
* sand = np.asarray(PILImage.open("./images/MASK1.png").convert('L'))/255
* pad = crop\_size\*2
* #pad for safety
* crop1 = np.pad(sand, pad\_width=pad, mode='constant', constant\_values = 1)
* # centerx = car\_x + pad
* # centery = car\_y + pad
* centerx = self.car.x + pad
* centery = self.car.y + pad
* #first small crop
* startx, starty = int(centerx-(crop\_size)), int(centery-(crop\_size))
* crop1 = crop1[starty:starty+crop\_size\*2, startx:startx+crop\_size\*2]
* #rotate
* crop1 = scipy.ndimage.rotate(crop1, -self.car.angle, mode='constant', cval=1.0, reshape=False, prefilter=False)
* #again final crop
* startx, starty = int(crop1.shape[0]//2-crop\_size//2), int(crop1.shape[0]//2-crop\_size//2)

 image = crop1[starty:starty+crop\_size, startx:startx+crop\_size].reshape(crop\_size, crop\_size)



* Car circling at one once after completing random rotations in actor’s zone
* Tried with different learning rate
* And with different rewards
* By trying with more random rotations
* Car circling at one once after completing random rotations in actor’s zone
* Removed dropout and added batch normalization so that action values are normalized assuming and added 1\*1 convolution
* Car circling at one once after completing random rotations in actor’s zone

Challenges

* Conversion from numpy array to tensor and going through different out of bonds issues
* Adding TD3 into Kivy environment
* Struggled with different rewards