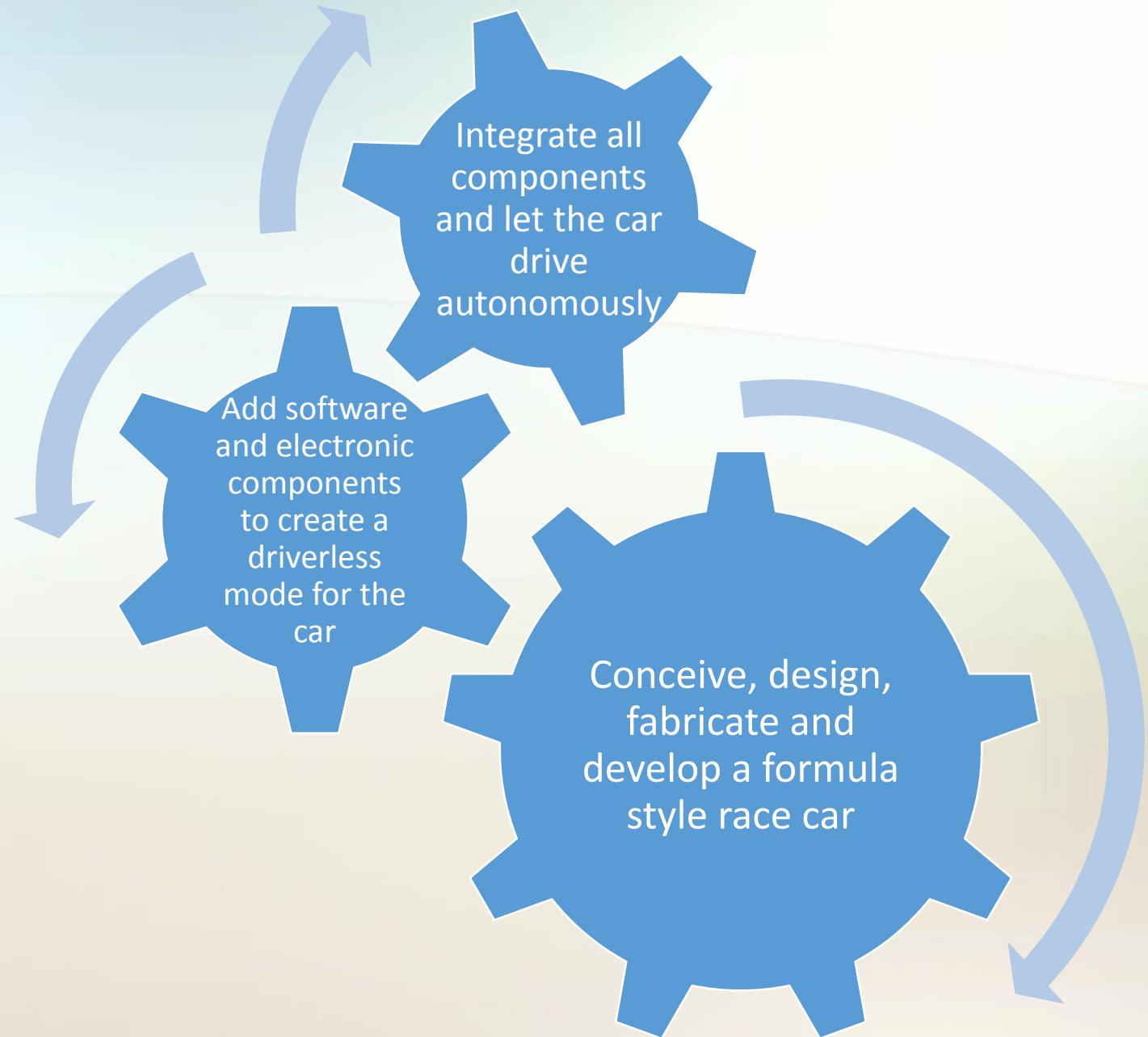


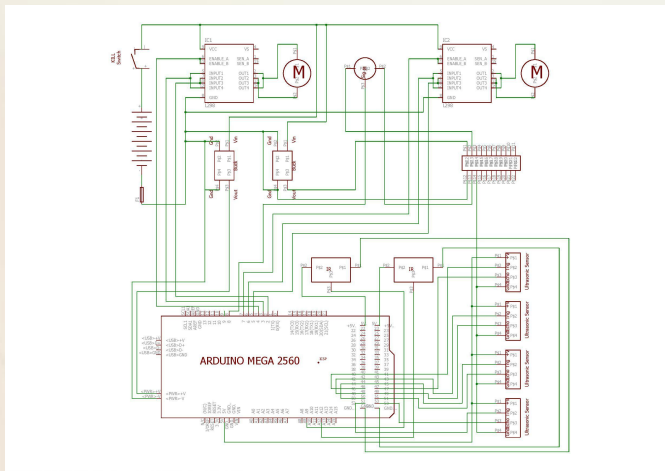
INTENT OF DRIVERLESS EVENT



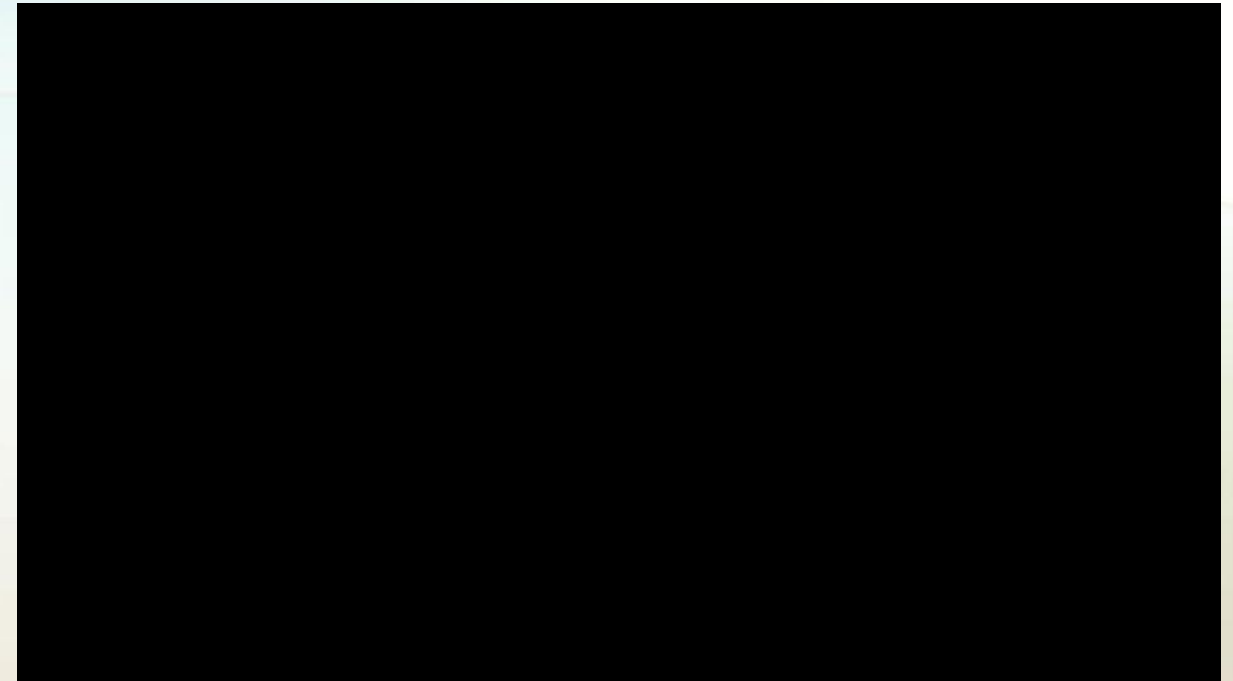
DRIVERLESS PROTOTYPE

Components-

1. One servo
-MG90
2. Two DC motors
3. Motor drivers
- L298N
4. 12V Lead acid
battery



Initial schematic design – including IR sensors



Prototype – Design and testing

DESIGN CONSIDERATIONS

Reliability

- Optimal position of sensors
- Choice of motors and actuators with respect to weight

Cost

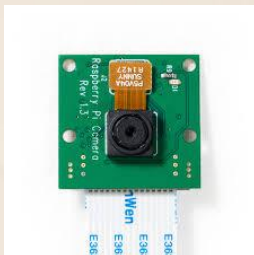
- Selection of sensors and other components
- Minimal data training cost

Performance - Hardware and software

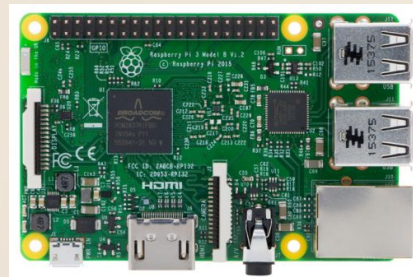
- Manageable wiring
- Efficient integration
- Efficient algorithms
- No delays, real time prediction

Prototype 1 – Driverless prototype using CV via hand-engineering

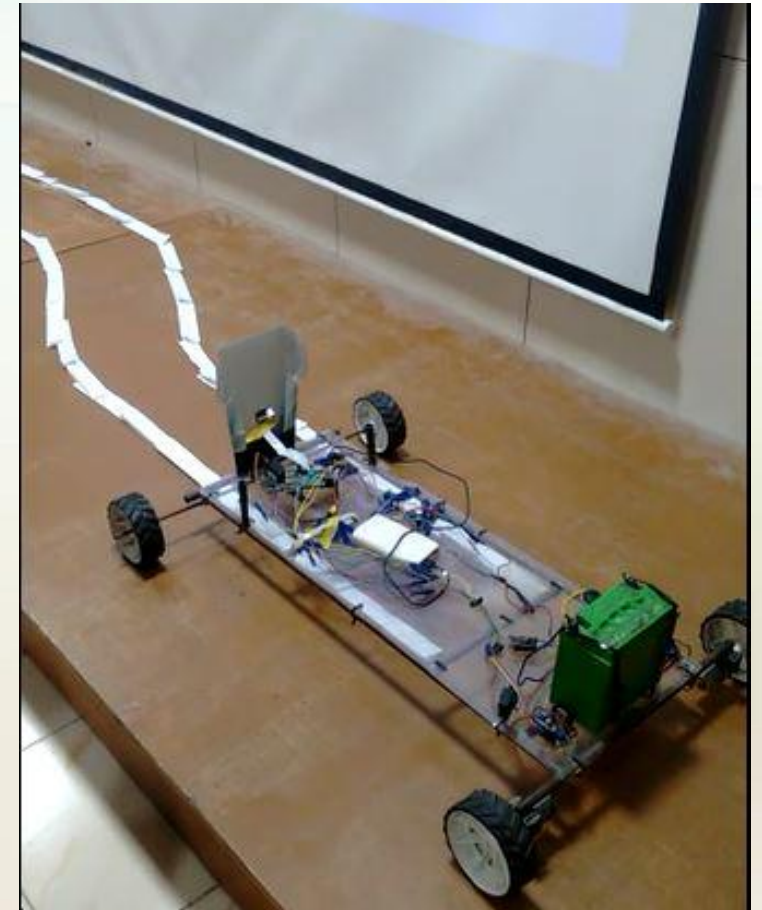
- Components
 - Raspberry Pi 3
 - Pi Camera
- Aim
 - Prototype to run between two white lanes
- Method
 - Implementation of Computer Vision techniques
 - Hough lines
 - Filters
 - Edge detection
 - Region of interest



Pi camera



Raspberry Pi 3



Prototype



Prototype 1 - Video



Test image



Smoothing



Region of interest



Gray scaling



Edge detection



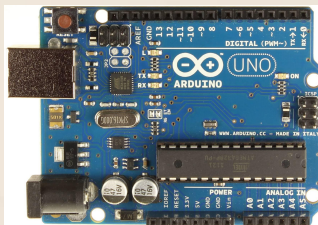
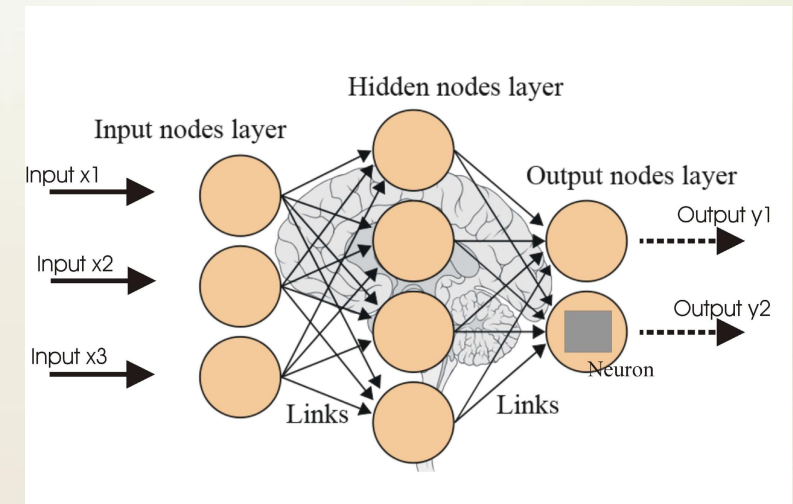
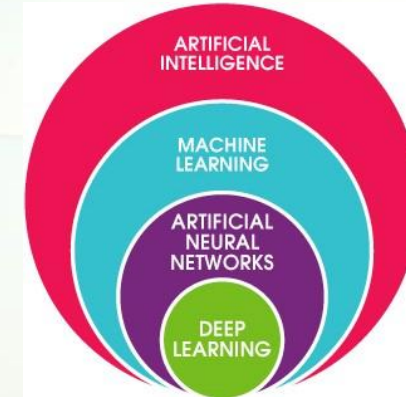
Incorrect lane detection



Correct lane detection after multiple iterations

Prototype 2 – Driverless prototype using CV and ML

- Components
 - Arduino Uno
 - Logitech Camera
- Aim
 - Prototype to run between two white lanes
 - Prediction of steering angle
- Dataset
 - Input: images labeled with steering angle
 - Output: predicted steering angle
- Machine Learning
 - Artificial neural network with artificial neurons
 - Learning achieved by giving weight to neurons



Arduino Uno



Logitech Webcam C270



Prototype 2 – data collection video

```

jupyter
CNN steering model 1 - Testing Modules - Preprocess data, Data

File Edit View Insert Cell Kernel Help
+ - * % < > H C Code

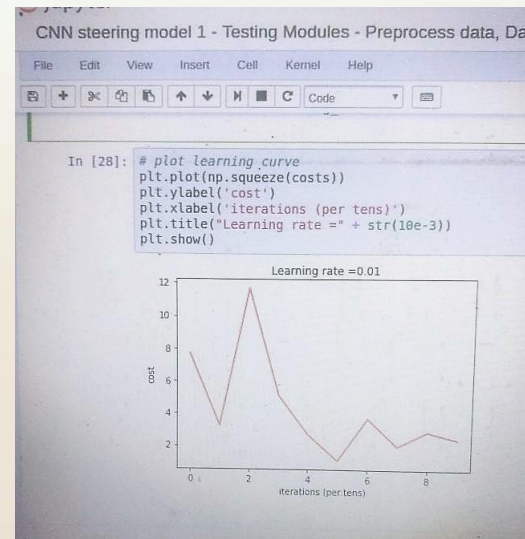
ys = np.array(train_batch[1]).reshape([len(
#print("ys length: ", len(ys))
#print("ys passed")
#print("image: {}, x shape: {}, y shape: {}".format(x, x.shape, y, y.shape))
#print(xs)
sess.run(train_step, feed_dict={model.x: xs,
#print("Train passed:")
train_error = loss.eval(session=sess, feed_dict={
    model.x: xs, model.y: ys})

if epoch % 10 == 0:
    print("Step: %d, train loss: %g" % (epoch, train_error))
    costs.append(train_error)

#checkpoint_path = os.path.join(CKPT_DIR, CKPT_FILE)
#filename = saver.save(sess, checkpoint_path)
#print('Model saved in file: %s' % filename)

Step: 0, train loss: 7.69963
Step: 10, train loss: 3.23576
Step: 20, train loss: 11.6841
Step: 30, train loss: 5.10057
Step: 40, train loss: 2.67917
Step: 50, train loss: 1.06684
Step: 60, train loss: 3.69885
Step: 70, train loss: 1.97931
Step: 80, train loss: 2.89156
Step: 90, train loss: 2.41979

```



```

Testing the accuracy

In [29]: xs, ys = data_reader.test_set()

In [33]: test_error = loss.eval(session=sess, feed_dict={model.x: np.array(xs),
print("Test Error: {}".format(test_error))

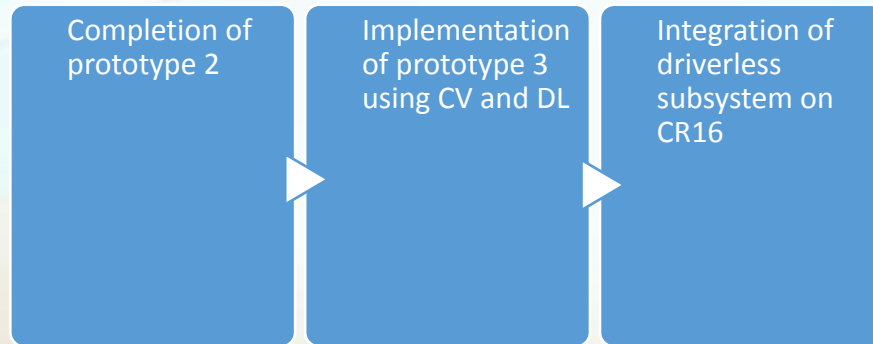
Test Error: 3.563483238220215

In [ ]:

```

Machine learning algorithm – training and testing the accuracy of the model

FUTURE PLANS



Prototype 3 – Driverless prototype on a scale version of chassis using CV and DL

- Components
 - Arduino Uno
 - Logitech Camera
- Aim
 - Prototype to run between two lanes marked by cones
 - Prediction of steering angle
- Dataset
 - Input: images labeled with steering angle
 - Output: predicted steering angle
- Deep Learning
 - Supervised learning
 - Model (x,y)
 - x : collection of features of data point
 - y : label of data point
 - Gradient of cost