

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ



AUTO PARKING OF VEHICLE BY USING SUPERVISED LEARNING APPROACH

SUPERVISOR: DR. MUWAHIDA LIAQUAT

GROUP MEMBERS:

MAZAHIR HUSSAIN

MUSTAQEEM ASHRAF

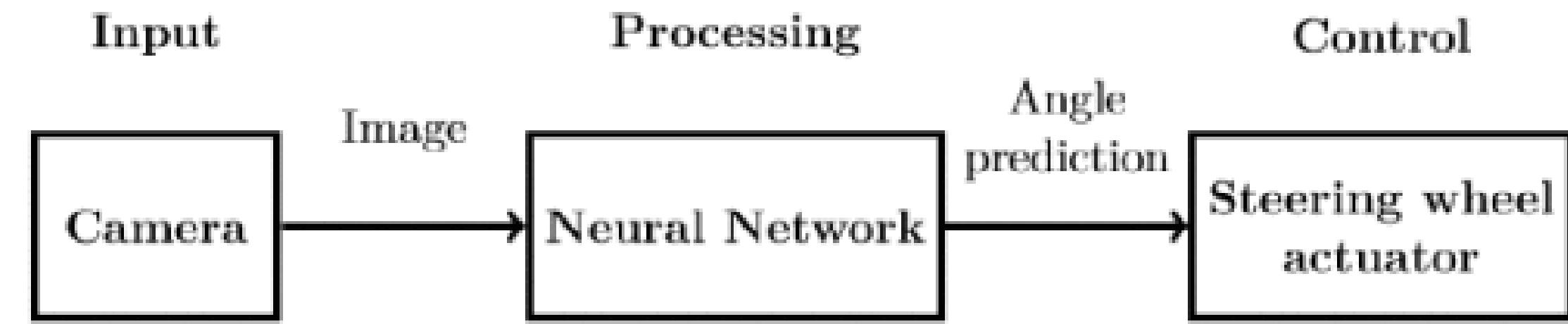
SHAMSA KANWAL

TOOBA ANWAR

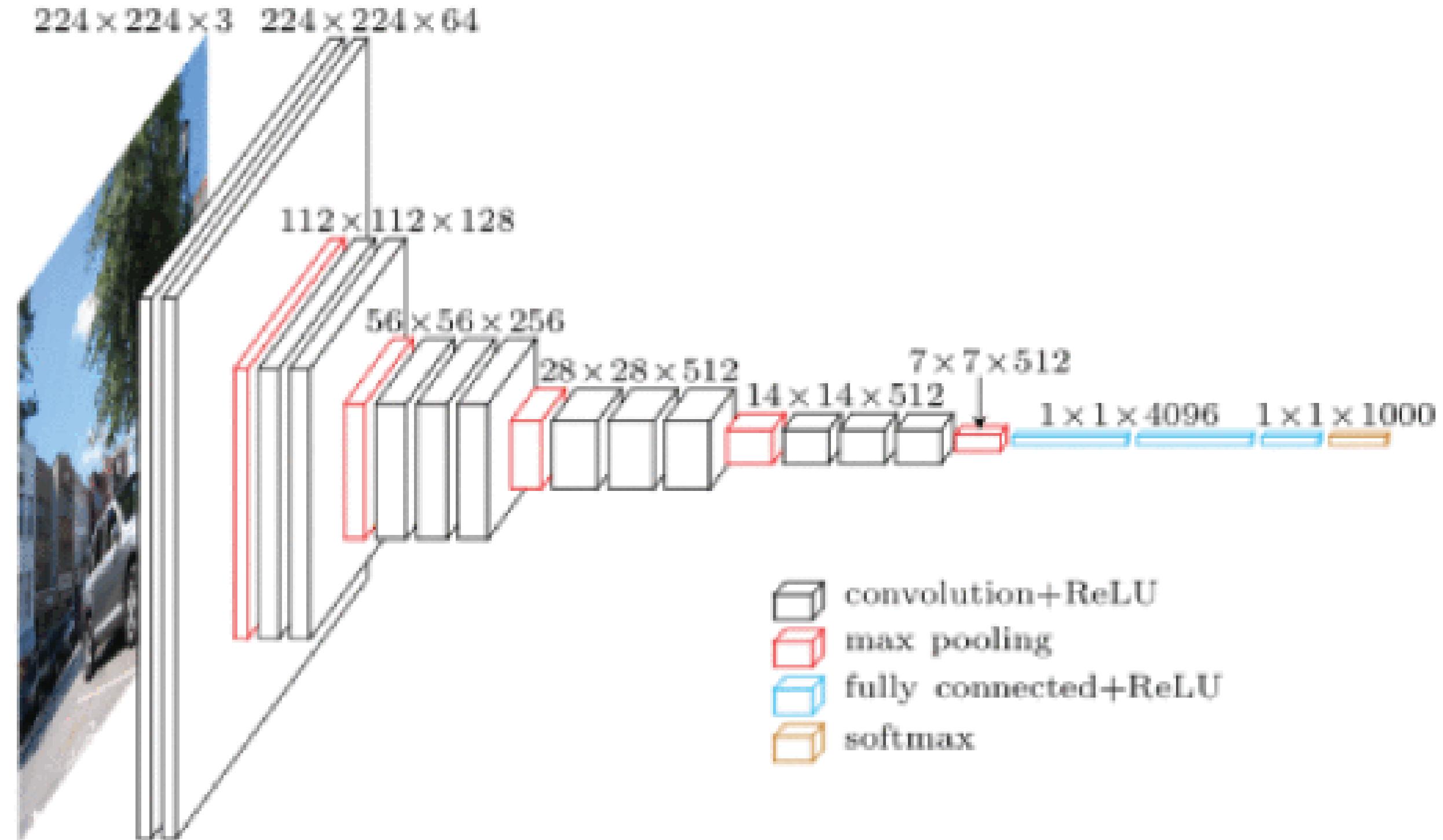
Introduction

1. Autonomous vehicles are designed to complete most driving tasks that human drivers can do
2. A CNN is trained to map raw pixels from a single front-facing camera directly to steering commands.
3. We can mainly eliminate human experience and learn parking strategies autonomously and quickly using this technology

Block Diagram



Block Diagram



Tasks Assigned

Mustaqeem Ashraf (Leader) & Mazahir Hussain

- Creation of data set
- Learn TensorFlow Library
- Deep learning Algorithms (CNN)
- Training of Model

Tasks Assigned

Shamsa Kanwal & Tooba Anwar

- Processing of created data
- Learn OpenCV Library
- Controller Design and implementation
- Hardware Implementation
- Documentation and styling

Tasks Achieved

- *Learn TensorFlow library*
- *Learn OpenCV library*
- *Deep Learning Algorithms (CNN)*
- *Carla simulator for data Set*
- *Preprocessing of input data*



Tasks Remaining

- Training of Model
- Controller Design & Implementation
- Hardware implementation



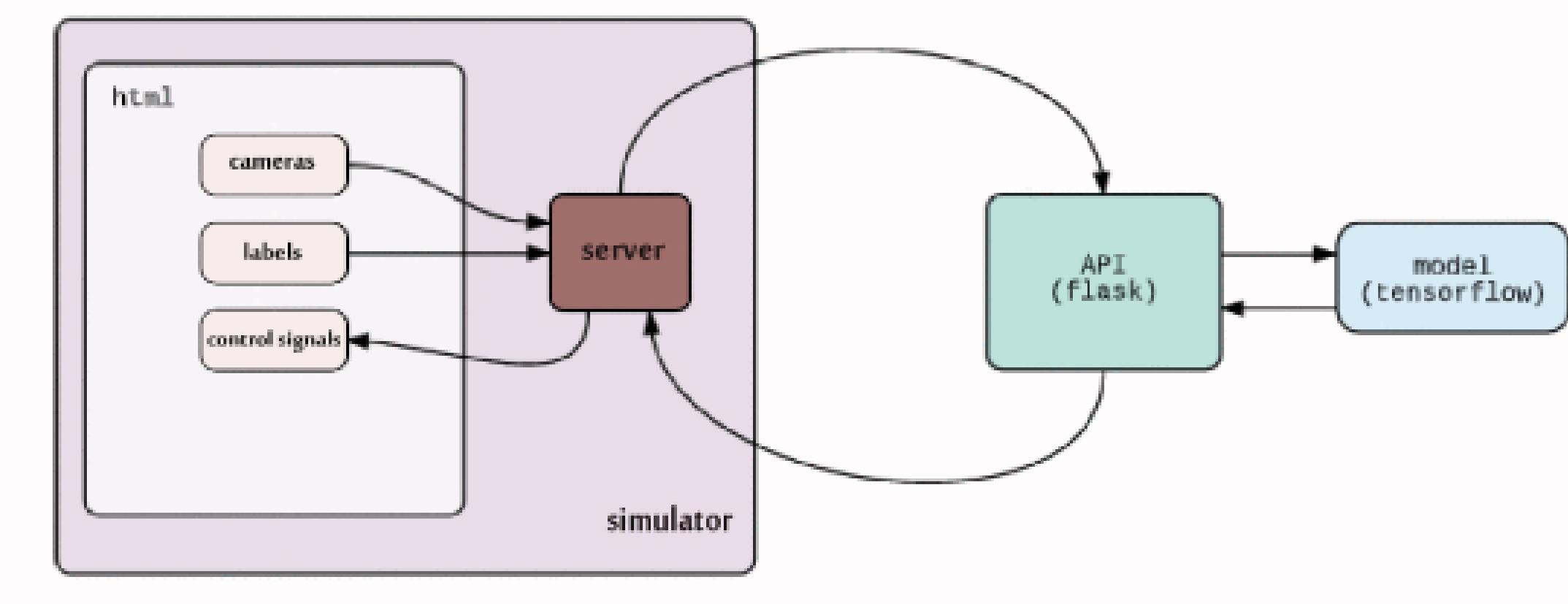
Software/Hardware

```
for a in images:  
    try:  
        image = Image.open(path + '\\\\' + a)  
        # resize image  
        image = cv2.resize(image, (160, 320, 3), interpolation=cv2.INTER_AREA)  
        image = image/255  
  
        image = np.array(image)  
        # sim = Image.from array(image)  
        data.append(image)  
        labels.append(angle)  
    except:  
        print("Error loading image")
```

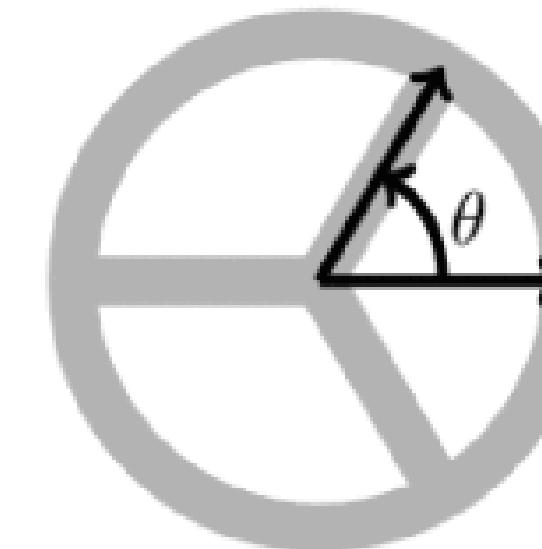
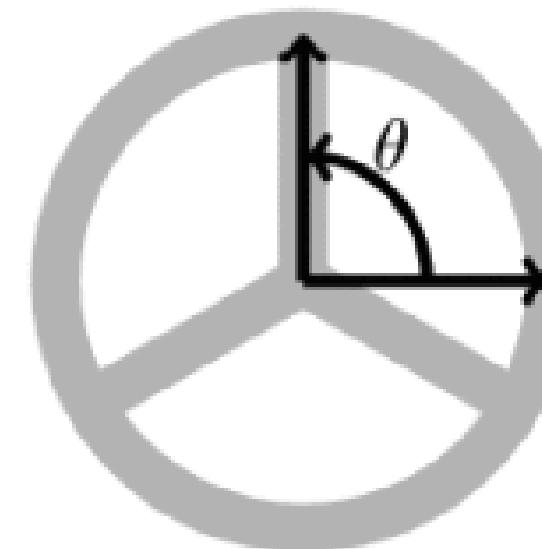
Software/Hardware

```
model.add(Conv2D(filters=24, kernel_size=(5, 5), strides=(2, 2), input_shape=X_train.shape[1:]))  
model.add(Conv2D(filters=36, kernel_size=(5, 5), strides=(2, 2)))  
model.add(Conv2D(filters=48, kernel_size=(5, 5), strides=(2, 2)))  
model.add(Conv2D(filters=64, kernel_size=(3, 3)))  
model.add(Conv2D(filters=64, kernel_size=(5, 5)))  
model.add(Conv2D(filters=128, kernel_size=(5, 5)))  
  
model.add(Dropout(rate=0.5))  
  
model.add(Flatten())  
  
model.add(Dense(100, activation='elu'))  
model.add(Dense(50, activation='elu'))  
model.add(Dense(10, activation='elu'))  
model.add(Dense(3, activation='linear'))
```

Software/Hardware



Software/Hardware



Software/Hardware



Software/Hardware



Software/Hardware

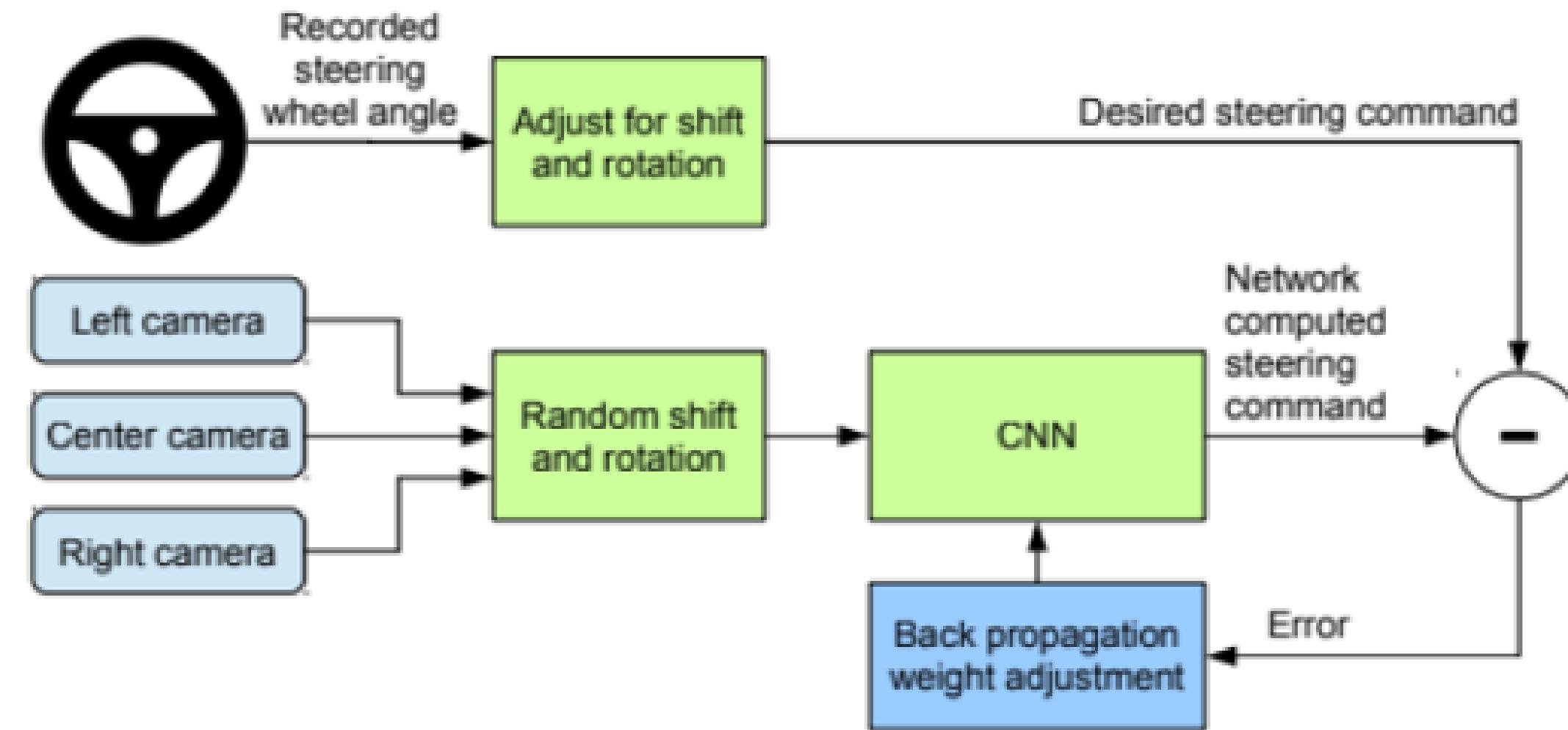
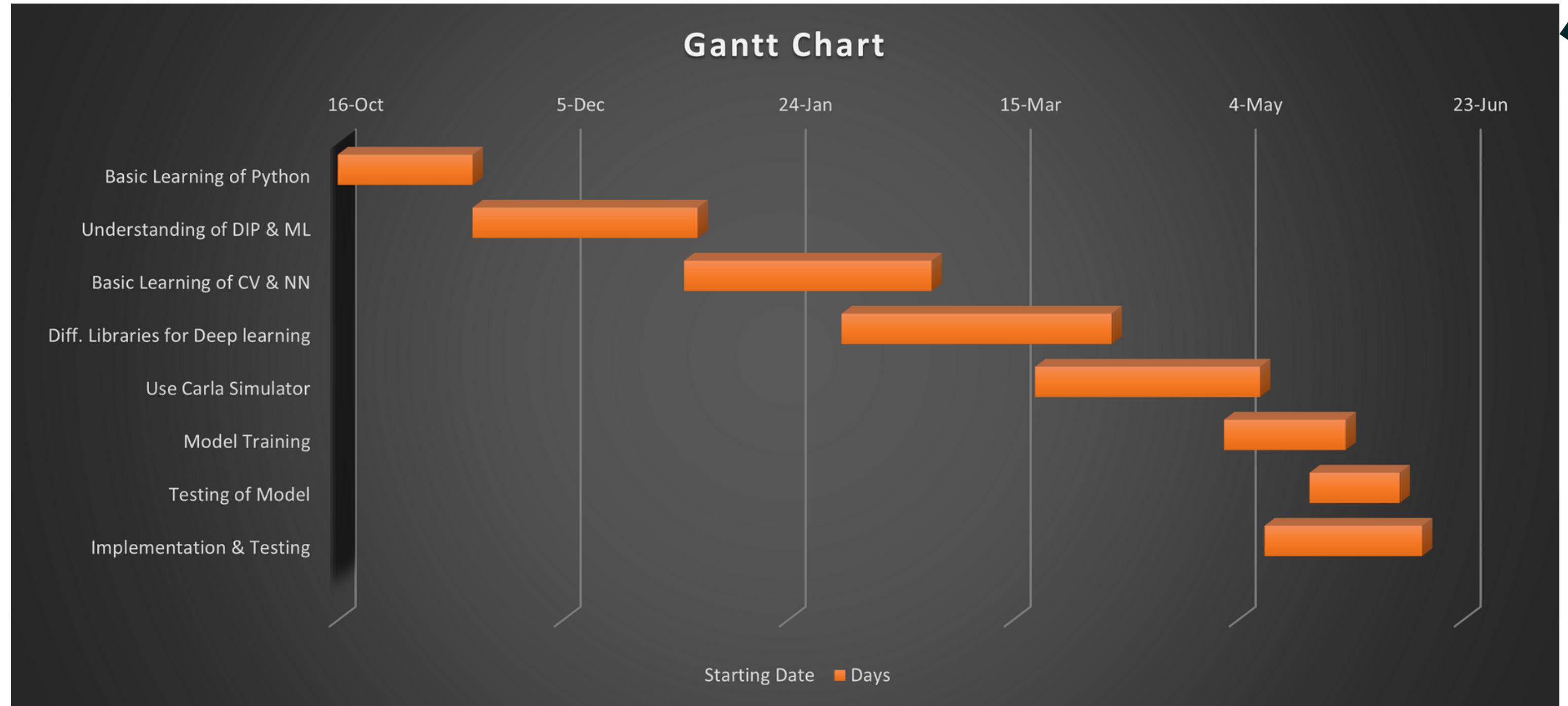


Figure 3: Training the neural network

Timeline



Conclusion

- Collected Data from CARLA SIMULATOR by using camera sensor
- Preprocessing on collected Data
- Built CNN Model
- Implementation of Model

Thank you

