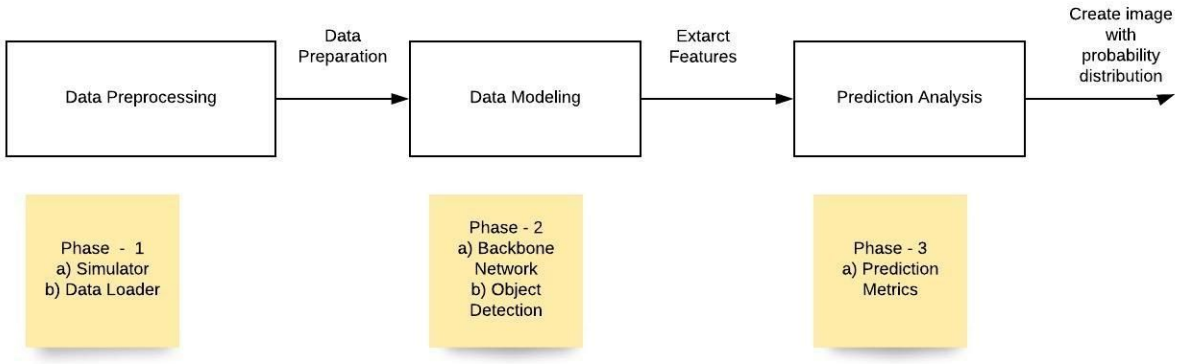
PnP Prototyping

Project Phase:



Data Loader:

DataLoader Requirements

A) Generated Data Properties: [This is a Gaussian data generated with mean & std]

1) Start velocity
2) 1st deceleration
3) Stop time (min time varies)
4) Accl at stop
5) 2nd accl
6) Vehicle width, length
7) Stop distance
8) # of cars except reference car

Add some sim noise add to all the simulator data

B) Map information at t= 0

It should have only

a) 4 Stop line information
b) 4 Path information

C) Frame Generation

At t = 0,

Total frames:

Generate the frames from configurable start# to end# with Configurable Reference frame

D) Coordinates

It will start with the coordinates of camera view at t = 5 sec.
Which will be pre calculated origin based on the reference vehicles coordinates at future t = 5 sec

E) Header of csv file

Exp no, seq no, N, [10 parameters * N]

F) Parameters

X, y = position
X',y' = velocity
X'',y'' = acceleration
Theta = angle
Theta' = rotational angle
H, W = size

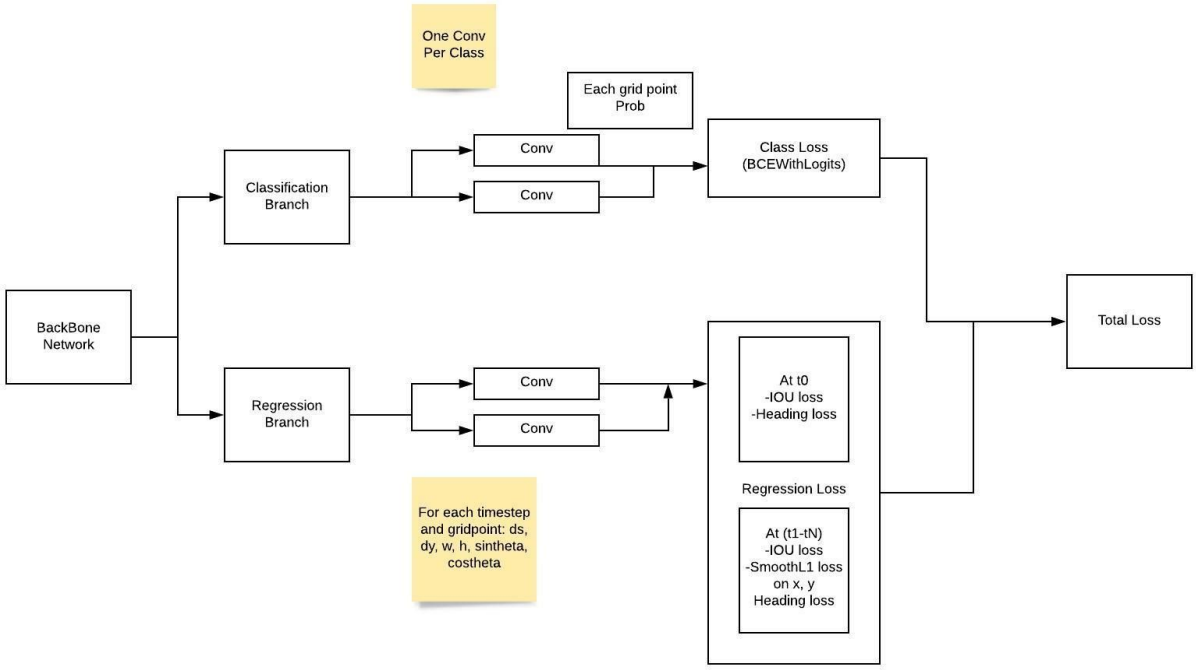
G) Data Structure to maintain the AV's state

```
struct vehicle_state {
  size_t id;
  size_t type;
  double width_p;
  double length_p;
  double loc_x_p;
  double loc_y_p;
  double heading_rad;
  double speed_pps;
  double acc_ppss;
};

struct frame_db {
  std::string sim_name;
  size_t seq_no;
  size_t frame_no;
  size_t ref_frame_no;
```

```
  size_t num_tracks;
  double pixels_per_meter;

  vehicle_state sdv_state;
  std::vector<vehicle_state> tracks;
};
```

Data Modeling

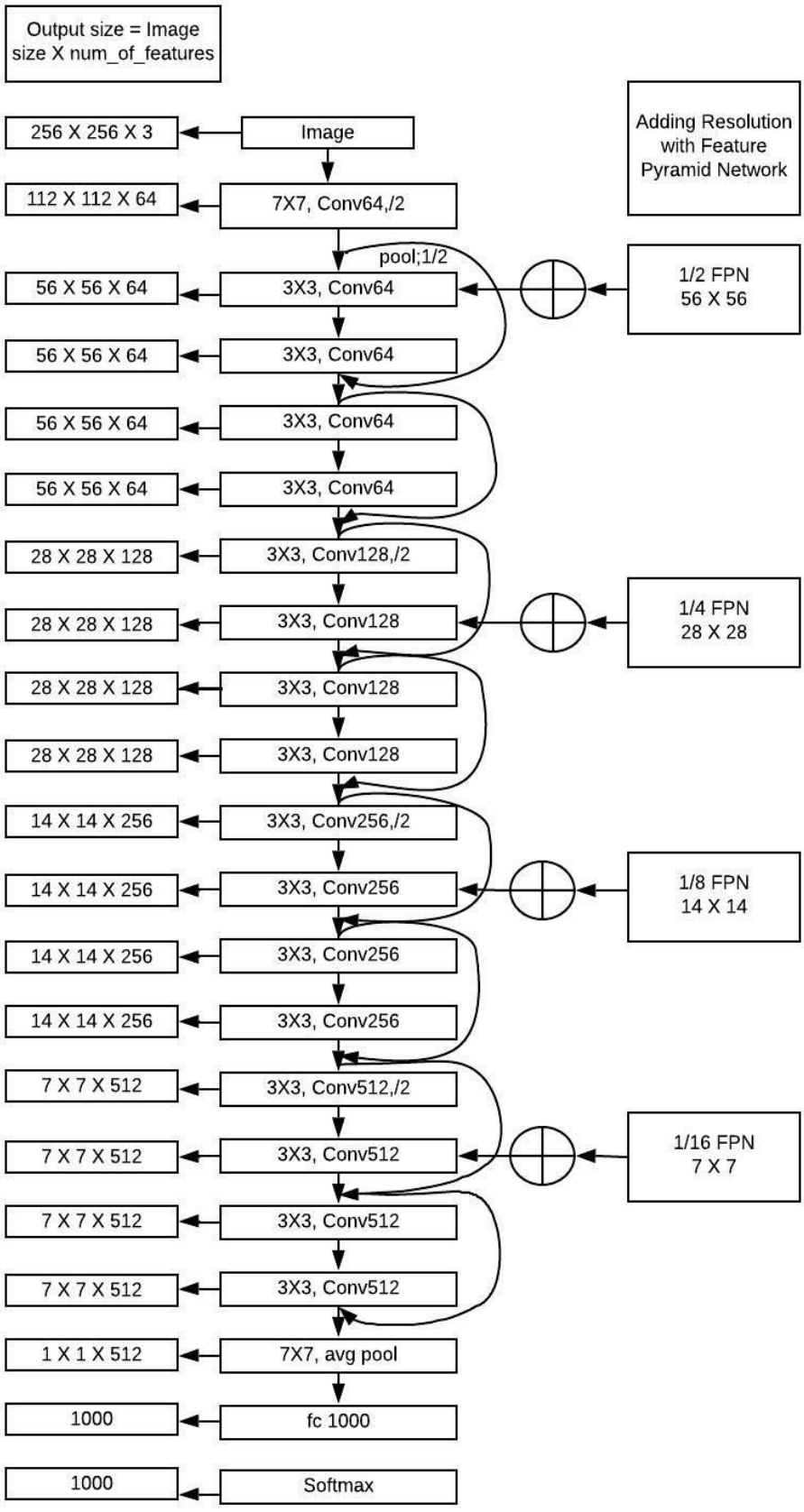
Model Overview:



Backbone Feature Pyramid Network:

1) **Backbone Network:** We use a deep residual network with 18 convolution layers (ResNet-18) to extract feature from the rasterization. From intermediate ResNet-18 layers, four feature maps at ½, ¼,⅛,1/116 of the resolution of the input are taken and aggregated with ¼ resolution with a feature pyramid network (FPN)

Fig: Resnet-18 [18-layer residual]



2) Regression Branch [Not implemented yet]

RROI :

RROI crops ->

RROI region of interest selection

a) Crop is a square crop of 60m by 60 m (with the actor in the center)
b) We should be able to take a rectangular crop and have some area in front of the actor over the back

Loss Calculation:

a) Smooth L1 Loss: Refinement loss is applied to all future timesteps t > 0 and time decay of 0.97 is applied

$$\text{loss}(x, y) = \frac{1}{n} \sum_i z_i$$

where z_i is given by:

$$z_i = \begin{cases} 0.5(x_i - y_i)^2, & \text{if } |x_i - y_i| < 1 \\ |x_i - y_i| - 0.5, & \text{otherwise} \end{cases}$$

x and y arbitrary shapes with a total of n elements each the sum operation still operates over all the elements, and divides by n .

Software Organization:

FCN.py : contains models VGGNet,
Pixor_model.py: Contains model pixor
Onehot.py : one hot encoded images
BagData_test.py : dataloader