Dashcam footage Accident Prediction

Team 9
Tejas Arya
Amritha Venkataramana

Agenda

- Problem Statement
- Data
- Models
- Evaluation
- Questions

Problem Statement- Car Accident Prediction

1) Given a set of video data predict the time estimate when accidents are prone to happen.
2) Classification using time and space based information → closer to accident frame → more weight?

3) Consider each video as a set of frames, analyze and predict.

Additional Data

1) Number of objects involved
2) Type of objects of involved
3) Number of objects in frame
4) Type of objects in frame
5) Accident location (Traffic signal, on road, etc.)
6) Number of lanes
7) Road signs
8) Type of accidents (Head-on, T-Bone, etc.)
9) Speedy

Processing

1) Handling missing values (Not all clips have similar number of frames)

2) Conversion to a format that could be supplied to the model

Models

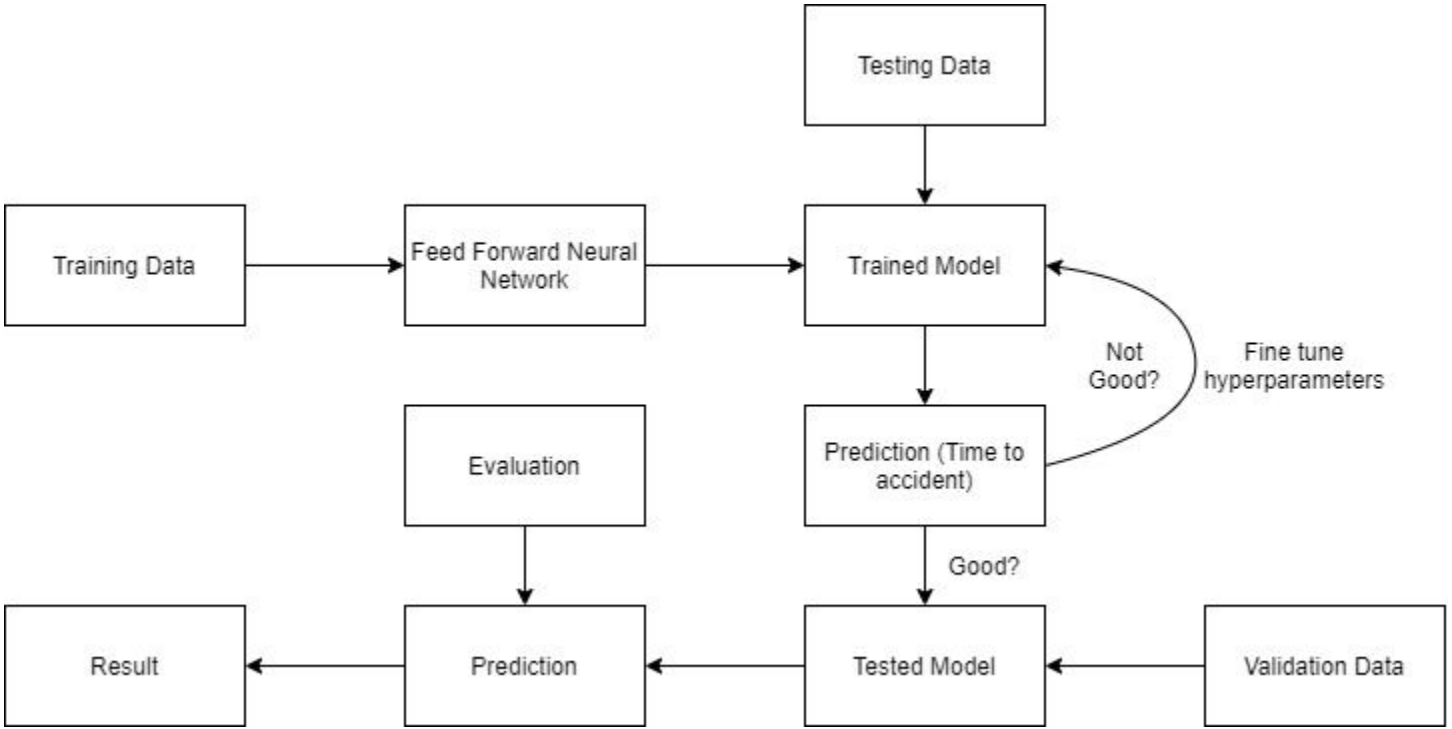
1) What we plan to use?
    a) Feedforward Neural Network

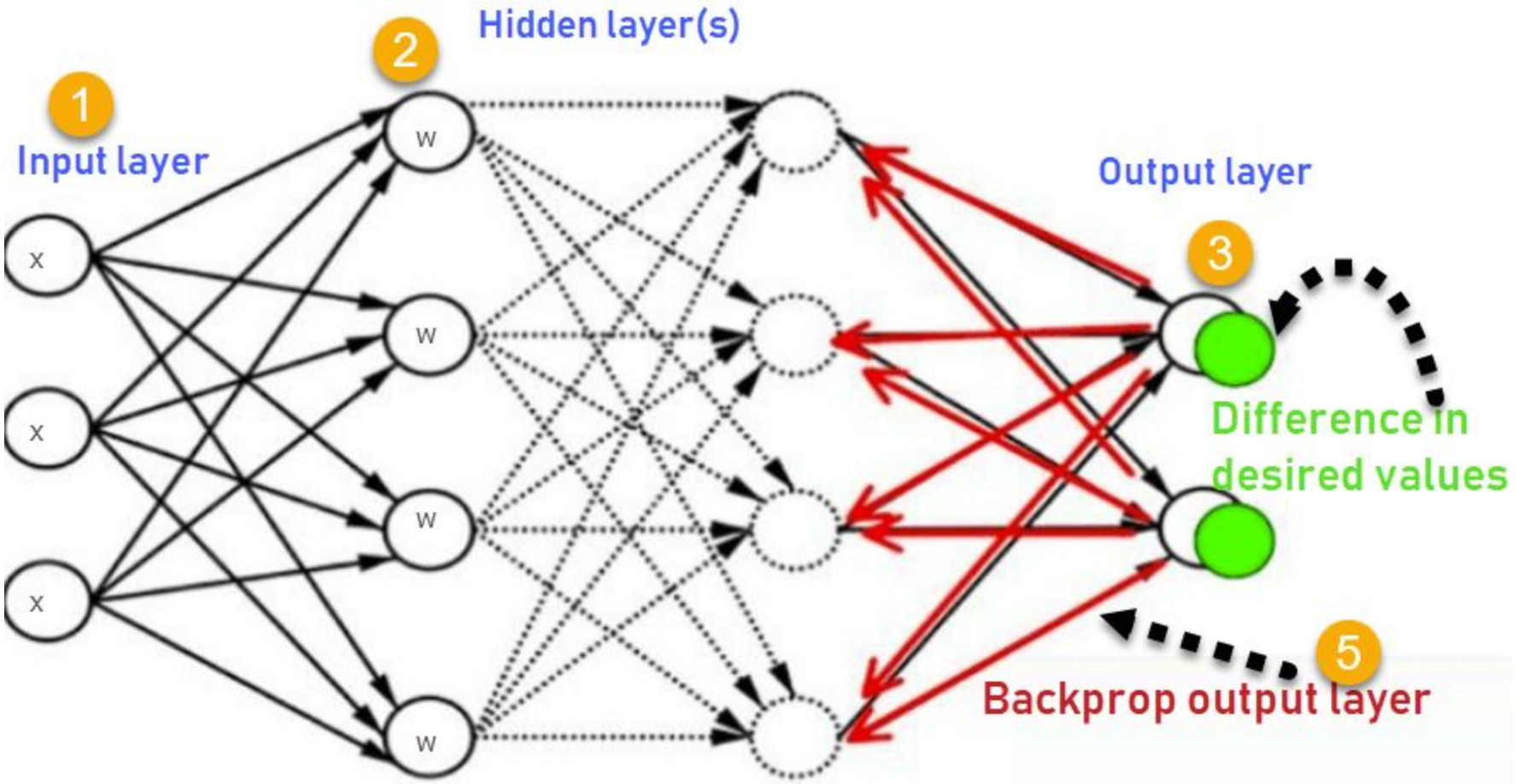
        Update weights at each training clip

        i) Activation Function- Leaky ReLU/ Swish
    b) Dynamic Spatial Attention RNN

2) Wrapping models with K-fold cross-validation

Feedforward Neural Network




Hidden layer(s)
2
1
Input layer
Output layer
3
x
x
x
w
w
w
w
Difference in
desired values
5
Backprop output layer

Dynamic Spatial Attention RNN

- DSA features
- Frame features
- RNN

DSA Features

- Space specific features for n objects in a frame and their locations
- Compute a weighted sum using above observation for each frame
- Weighted sum takes in account output from previous frame which is then again passed to the RNN (Attention weight)

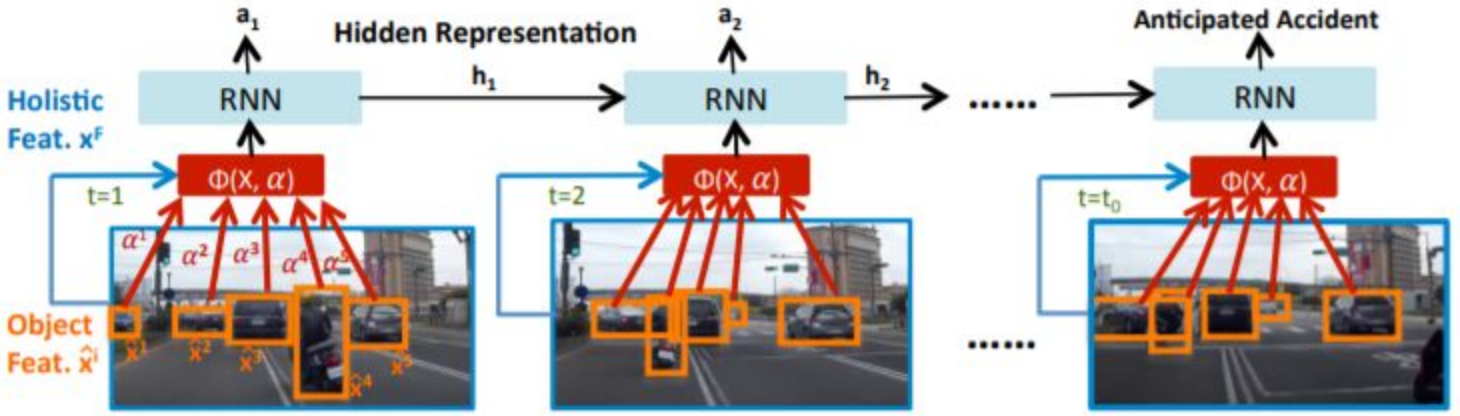
Frame features

- Features extracted from the frame from annotation task and some manually generated features discussed above
- Weighted sum of DSA features and frame features to ensure similar feature dimension
- Weights learned during training

RNN

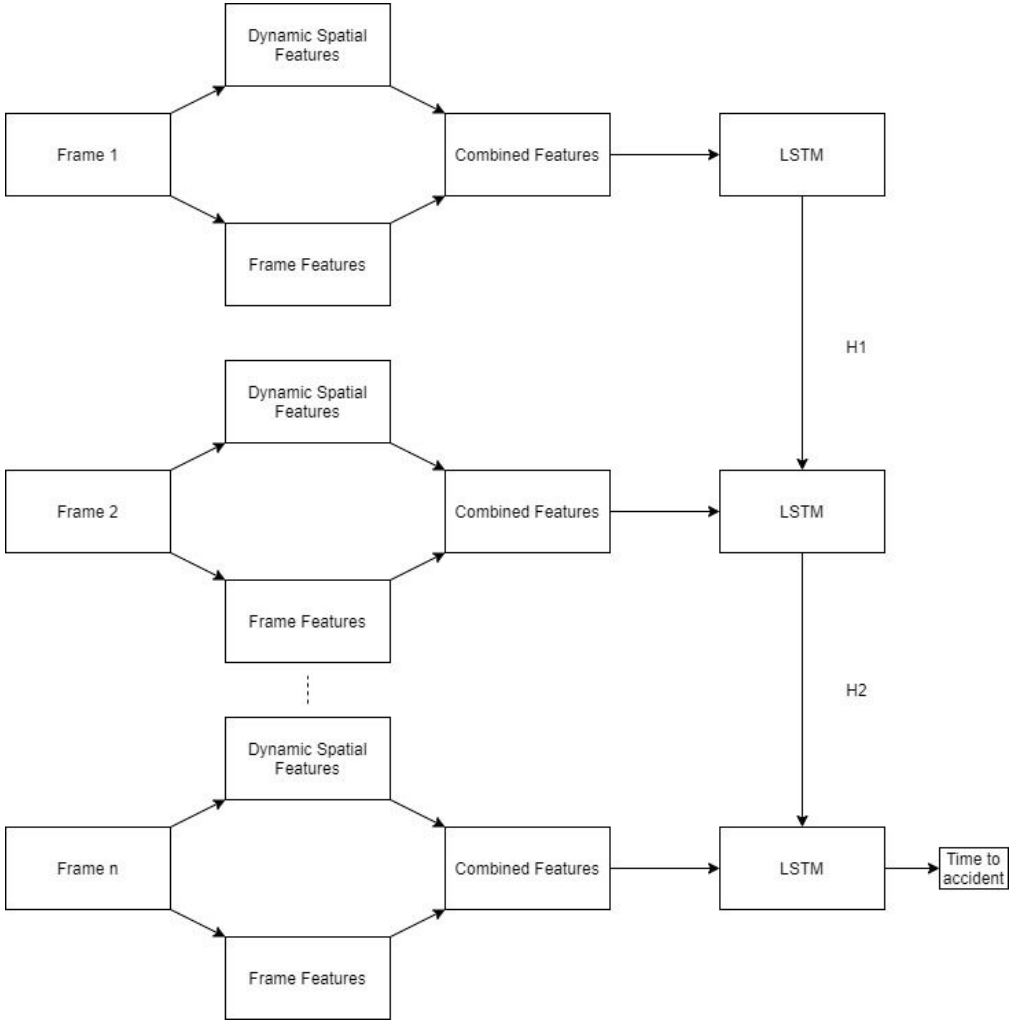
- RNN cannot capture long-term dependencies
- LSTM with memory cell (To store previous information)
- Three gates: Input gate, forget gate, output gate
- Input gate: Allows current observation and previous hidden representation
- Forget gate: Forget previously stored information
- Compute current memory and update it from previous
- Output gate: Compute current hidden representation using computed current memory

DSA + Feature RNN



Object features will contain space specific features and frame features from annotation tool and manually generated attributes

Image reference: https://yuxng.github.io/chan_accv16.pdf


Dynamic Spatial Features
Frame 1
Combined Features
LSTM
Frame Features
H1
Dynamic Spatial Features
Frame 2
Combined Features
LSTM
Frame Features
H2
Dynamic Spatial Features
Frame n
Combined Features
LSTM
Time to accident
Frame Features

Evaluation

- Classified frame later than actual accident frame: Penalized
- G_count: Classified frame before actual accident frame
- B_count: Classified frame after actual accident frame
- Misclassification rate: B_count / (G_count + B_count)
- G_count and B_count vs Number of trained clips
- RMSE: Square root of mean squared error of predicted time and actual accident time
- MAE: Absolute time difference between predicted time and actual accident time

Evaluation

1) We’ll penalize heavily if predicted crash frame is more than the actual crash frame.
2) Minimize the error between prediction and actual values (prediction << actual value)

Tools

1) Object Detection- LabelBox (Pretty good for larger labeling projects and has options for different types of labeling tasks)
2) VGG Image Annotator (VIA)
3) OpenCV
4) Tensor Flow, scikit-learn (Predicting accident frame)

Thanks!