Emotion Classification using LSTM Based on Driving Behavior

Hanyu Gao, Riya Sharma

Technical University of Munich
Department of Informatics

12.07.2019
Outline

1. Introduction
2. Workflow and Data illustration
3. Long-Short-Term-Memory
4. Experiments and Result
5. Robustness and Proposal
6. Conclusion
7. Reference
1. Introduction
Introduction
Autonomous Driving & Deep Learning

There are in general four questions a car needs to be able to answer to achieve the final goal of autonomy.

1) Where am I? → Localisation and Mapping

2) Where is everybody else? → Scene Understanding

3) How do I get from A to B? → Movement Planning

4) What’s the driver up to? → Driver State
Introduction

Autonomous Driving & Deep Learning

• **via semantic abstraction**
  - where each task is executed in a separate network and afterwards combined with classical control & decision-making algorithms.

• **end-to-end approach**
  - where a single DNN takes all the car’s inputs and computes a final output in a single step.
Classification Standard

- Emotion classification

- Driving behavior classification
Classification Standard

- Emotion classification
  - Basic (primary) emotions: Ekman’s Big 6[1]

- Disgust
- Fear
- Anger
- Happiness
- Sadness
- Surprise
Classification Standard

- Emotion classification
  - Basic (primary) emotions: Ekman’s B
  - Plutchik’s wheel of emotions
Classification Standard

- Emotion classification
  - Basic (primary) emotions: Ekman’s Big 6[1]
  - Plutchik’s wheel of emotions
  - PAD emotion representation model
Classification Standard

- Emotion classification
  - Basic (primary) emotions: Ekman’s Big 6[1]
  - Plutchik's wheel of emotions
  - PAD emotion representation model

- Driving States Classification
  - normal driving, aggressive driving or drowsy driving
Classification Standard

- Emotion classification
  - Basic (primary) emotions: Ekman’s Big 6 [1]
  - Plutchik’s wheel of emotions
  - PAD emotion representation

- Driving States Classification
  - normal driving, aggressive driving
  - driving style: dissociative, anxious, risky, angry, high-velocity, distress reduction, patient, and careful...... [2]

<table>
<thead>
<tr>
<th>MDSI factors</th>
<th>Self-esteem</th>
<th>Need for control</th>
<th>Sensation seeking</th>
<th>Extraversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dissociative</td>
<td>−0.38**</td>
<td>−0.04</td>
<td>0.10</td>
<td>−0.23**</td>
</tr>
<tr>
<td>Anxious</td>
<td>−0.05</td>
<td>−0.08</td>
<td>−0.11</td>
<td>−0.22*</td>
</tr>
<tr>
<td>Risky</td>
<td>−0.19*</td>
<td>0.09</td>
<td>0.40**</td>
<td>−0.02</td>
</tr>
<tr>
<td>Angry</td>
<td>−0.10</td>
<td>0.22*</td>
<td>0.13</td>
<td>0.14</td>
</tr>
<tr>
<td>High-velocity</td>
<td>−0.11</td>
<td>0.13</td>
<td>0.18*</td>
<td>0.01</td>
</tr>
<tr>
<td>Distress reduction</td>
<td>0.04</td>
<td>0.01</td>
<td>0.01</td>
<td>0.06</td>
</tr>
<tr>
<td>Patient</td>
<td>0.23**</td>
<td>−0.04</td>
<td>−0.09</td>
<td>−0.16</td>
</tr>
<tr>
<td>Careful</td>
<td>0.27**</td>
<td>0.17*</td>
<td>−0.31**</td>
<td>0.02</td>
</tr>
</tbody>
</table>
2. Workflow
Workflow

- Traditional Machine Learning Workflow
Workflow —— Data

• Vehicle dynamics-based

• Driver dynamics-based
Workflow —— Data

- Vehicle dynamics-based technique
  - Internal data collectors
    e.g: Controller Area Network bus (CAN Bus)
Workflow —— Data

- Vehicle dynamics-based technique
  - Internal data collectors
e.g.: Controller Area Network bus (CAN Bus)
Workflow —— Data

• Vehicle dynamics-based technique
  - Internal data collectors, e.g: Controller Area Network bus (CAN Bus)
  - External data collectors, e.g: Accelerometer, Gyroscope, Smartphone
Workflow —— Data

- Vehicle dynamics-based data
  - vehicle orientation
  - speed
  - acceleration
  - braking events
  - throttle
  - altitude
  - engine and fuel consumption
  - ......
Workflow —— Data

- Driver dynamics-based technique
  - Video based

Car camera
## Workflow —— Data

- Driver dynamics-based technique
  - Video based
  - Bio-signal based[3]

<table>
<thead>
<tr>
<th>EMG</th>
<th>ECG</th>
<th>respiration</th>
<th>EDA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean Value</strong></td>
<td>Mean amp</td>
<td>Mean amp</td>
<td>Mean amp of Skin</td>
</tr>
<tr>
<td>Root Mean Square</td>
<td>Heart Rate</td>
<td>Respiration Rate</td>
<td>Conductance Responses</td>
</tr>
<tr>
<td></td>
<td>Mean_abs_first_difference</td>
<td>Mean_abs_first_difference</td>
<td>Rate of Skin Conductance Responses</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mean_abs_first_difference</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mean rise duration of Skin</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Conductance Response</td>
</tr>
</tbody>
</table>

- EMG: muscle activity
- ECG: manifestation of contractile activity of the heart
- Respiration: breathing depth
- EDA: skin conductance activity
Workflow —— Data

- Driver dynamics-based Data
  - electrocardiogram(EMG)
  - Electrocardiography(ECG)
  - Respiration
  - electrodermal activity(EDA)
  - eye gaze
  - EEG activities
  - head and body pose
  - ......
Workflow —— Data

- Problem
  - In-cab lighting changes
  - Camera position
  - Statistical data extraction (mean, median, mode ....)
  - Acquisition difficulty
  - ....
Workflow —— Data

- Problem
  - In-cab lighting changes
  - Camera position
  - Statistical data extraction (mean, median, mode ......)
  - Acquisition difficulty
  - ......

Feature Selection!
Workflow —— Model

• Machine Learning Model
Workflow —— Model

- Machine Learning Model
  - Support Vector Machine (SVM)
  - Bayesian Logistic Regression (BLR)
  - Hidden Markov Model (HMM)[5]
Workflow —— Model

- Machine Learning Model
  - Support Vector Machine (SVM)
  - Bayesian Logistic Regression (BLR)
  - Hidden Markov Model (HMM)[5]

<table>
<thead>
<tr>
<th></th>
<th>Unweighted recall (UA)</th>
<th>Weighted recall (WA)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bayesian Logistic Regression (BLR)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM baseline</td>
<td>38.2%</td>
<td>39.2%</td>
</tr>
<tr>
<td>HMM baseline</td>
<td>35.9%</td>
<td>37.2%</td>
</tr>
<tr>
<td>BLR</td>
<td>41.57%</td>
<td>39.87%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Angry</th>
<th>Emphatic</th>
<th>Neutral</th>
<th>Positive</th>
<th>Rest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angry</td>
<td>290</td>
<td>171</td>
<td>65</td>
<td>63</td>
<td>22</td>
</tr>
<tr>
<td>Emphatic</td>
<td>210</td>
<td>752</td>
<td>325</td>
<td>136</td>
<td>85</td>
</tr>
<tr>
<td>Neutral</td>
<td>748</td>
<td>1094</td>
<td>2057</td>
<td>1109</td>
<td>369</td>
</tr>
<tr>
<td>Positive</td>
<td>23</td>
<td>13</td>
<td>39</td>
<td>131</td>
<td>9</td>
</tr>
<tr>
<td>Rest</td>
<td>95</td>
<td>58</td>
<td>134</td>
<td>197</td>
<td>62</td>
</tr>
</tbody>
</table>
Workflow —— Model

• Machine Learning Model
  - Support Vector Machine (SVM)
  - Bayesian Logistic Regression (BLR)
  - Hidden Markov Model (HMM)[5]
  - K-means
  - Symbolic Aggregate Approximation (SAX)
  - Gaussian Mixture Model (GMM)
Workflow —— Model

- Deep Learning Model VS Machine Learning Model
  - Advantages: No more Feature selection, a holistic data-driven approach
  - Disadvantages: More Computation, huge amount data
Workflow —— Model

- Deep Learning Model
  - Convolutional Neural Network (CNN)
Workflow —— Model

- Deep Learning Model
  - Convolutional Neural Network (CNN)
  - Recurrent Neural Network (RNN)
Workflow — Model

- Deep Learning Model
  - Convolutional Neural Network (CNN)
  - Recurrent Neural Network (RNN)
  - Long Short-Term Memory (LSTM)
Workflow —— Model

• Deep Learning Model
  - Convolutional Neural Network (CNN)
  - Recurrent Neural Network (RNN)
  - Long Short-Term Memory (LSTM)
  - Gated Recurrent Unit (GRU)
3. Long Short-Term Memory
Long Short-Term Memory

• RNN

\[ s_t = f(Ux_t + Ws_{t-1}) \]
\[ y = g(Vs_t) \]

\( x_t \): input at time \( t \)

\( s_t \): hidden state at time \( t \) (memory of the network).

\( f \): is an activation function (e.g, \( \text{tanh}() \) and \( \text{ReLU}() \)).

\( U, V, W \): network parameters (unlike a feedforward neural network, an RNN shares the same parameters across all time steps).

\( g \): activation function for the output layer (typically a softmax function).

\( y \): the output of the network at time \( t \)
Long Short-Term Memory

• Vanishing Gradient Problem
  - Back-propagation through time
  - With multiple matrix multiplications, gradient values shrink exponentially
  - Gradient contributions from “far away” steps become zero

![Some Common Activation Functions](image1)
![Activation Function Derivatives](image2)
Long Short-Term Memory

- Long Short-Term Memory

long-short term memory modules used in an RNN
Long Short-Term Memory

- Long Short-Term Memory
  - forget gate layer

\[ f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f) \]
Long Short-Term Memory

- Long Short-Term Memory
  - forget gate layer
  - input gate layer

\[ i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \]
\[ \tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \]
Long Short-Term Memory

- Long Short-Term Memory
  - forget gate layer
  - input gate layer
  - current layer

\[ C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \]
Long Short-Term Memory

- Long Short-Term Memory
  - forget gate layer
  - input gate layer
  - current layer
  - output layer

\[
o_t = \sigma \left( W_o \left[ h_{t-1}, x_t \right] + b_o \right)
\]
\[
h_t = o_t \times \text{tanh} \left( C_t \right)
\]
• Whole architecture of one LSTM cell
Long Short-Term Memory

- Limitations
  - Increase the number greatly of weights compared with RNN
  - Still unbalanced weight in time series (better than RNN)
Long Short-Term Memory

- Advantages[6]
  - LSTM can handle noise, distributed representations and continuous values
Long Short-Term Memory

• Advantages[6]
  - LSTM can handle noise, distributed representations and continuous values
  - Parameter fine tuning not really necessary, lstm works well over a broad range of parameters
Long Short-Term Memory

• Advantages[6]
  - LSTM can handle noise, distributed representations and continuous values
  - Parameter fine tuning not really necessary, lstm works well over a broad range of parameters
  - Update complexity is $O(1)$
Long Short-Term Memory

• Advantages[6]
  - LSTM can handle noise, distributed representations and continuous values
  - Parameter fine tuning not really necessary, lstm works well over a broad range of parameters
  - Update complexity is $O(1)$
  - Able to deal with long time sequence
4. Experiment and Results
Experiment and Results: Vehicle Based

Classical Approaches

1) k-nearest neighbours classification algorithm:

- accelerometer sensor data
- classes (normal driving and aggressive driving)
- 177 features were extracted and fed to k-nearest classification model

Data with normalization

Hanyu Gao, Riya Sharma (TUM) | Emotion Classification using LSTM Based on Driving Behavior, July 12, 2019
Experiment and Results: Vehicle Based

2) k-mean clustering algorithm (unsupervised learning):

- accelerometer sensor data + vehicle dynamics data (braking and turning)
- classes (normal driving and aggressive driving)
- other statistical features were also included (mean, max, variance) in feature vector
3) Support Vector Machine

- accelerometer sensor data + vehicle dynamics data (braking and turning)
- classes (normal driving and aggressive driving)
- other statistical features were included (mean, max, variance) in feature vector
Experiment and Results: Vehicle Based

• Results

- For k-nearest neighbours, maximum (100%) classification precision can be reached by selecting certain features.
Experiment and Results: Vehicle Based

- Problems (Classical Approach)
  - relies on hand crafting a set of features.
  - requires a domain expert knowledge to determine feature selection.
  - the separation happens between the feature extraction stage and the
    learning algorithm stage which become a challenging task for deciding which
    learning algorithm could be the best fit for the extracted features set.
Experiment and Results: Vehicle Based

• Problems (Classical Approach)
  - relies on hand crafting a set of features.
  - requires a domain expert knowledge to determine feature selection.
  - the separation happens between the feature extraction stage and the learning algorithm stage which become a challenging task for deciding which learning algorithm could be the best fit for the extracted features set.

Solution: End to End Approach (RNN and LSTM)!
Experiment and Results: Vehicle Based

End to End Approaches

Dataset Used: UAH-DriveSet

- rich timestamped data with more than 500 minutes of driving sessions.
- two types of roads (motorway and secondary).
- 6 different drivers and vehicles.
- 3 types of driving behaviours (normal, aggressive and drowsy).
Experiment and Results: Vehicle Based

The 9 feature vectors of sensor data at each time step

Inertial measurement sensors:  
1. Acceleration along x-axis & y-axis & z-axis  
2. Roll angle  
3. Pitch angle  
4. Yaw angle

GPS sensors:  
1. Vehicle Speed.

Camera sensors:  
1. Distance of vehicle -ahead.  
2. Number of detected vehicles
Experiment and Results: Vehicle Based

1) RNNs

- time series classification (Many to one architecture)

- internal state $h$ can capture the temporal dynamics

- input - a time-series window $S$ of feature vectors

- outputs a classification scores vector $O_s$.

RNN Model with sigmoid nonlinearity and softmax output

$$h_t = \sigma(W^{hh}h_{t-1} + W^{hx}x_t)$$

$$y_t = \text{softmax}(W^sh_t)$$

Loss Function at a step $t = J(t(\theta)) = -\sum_{j=1}^{K} t_{t,j} \log y_{t,j}$
Experiment and Results: Vehicle Based

- RNNs

Problems?

- memorising long sequence.
- also known as the “vanishing gradient” problem.
Experiment and Results: Vehicle Based

- RNNs

  Problems?

  - memorising long sequence.
  - also known as the “vanishing gradient” problem.

  Solution: LSTM!
Experiment and Results: Vehicle Based

2) Long Short Term Memory (2 layer)

- two LSTM memory cell layers.
- each layer have 100 hidden neurons.
- first layer input is a time-series window (64 feature vectors).
- second layer to output hidden feature vector.
- Finally, the last layer is a softmax layer.
Experiment and Results: Vehicle Based

• Results

<table>
<thead>
<tr>
<th>ROAD TYPE</th>
<th>TRUE STATE</th>
<th>RNN</th>
<th>Stacked LSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NORMAL</td>
<td>7.3</td>
<td>8.4</td>
</tr>
<tr>
<td></td>
<td>AGGRESSIVE</td>
<td>1.5</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>DROWSY</td>
<td>1.2</td>
<td>0.7</td>
</tr>
<tr>
<td>MOTORWAY</td>
<td>NORMAL</td>
<td>3.5</td>
<td>1.1</td>
</tr>
<tr>
<td></td>
<td>AGGRESSIVE</td>
<td>5.2</td>
<td>8.0</td>
</tr>
<tr>
<td></td>
<td>DROWSY</td>
<td>1.3</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>NORMAL</td>
<td>3.0</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>AGGRESSIVE</td>
<td>1.6</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>DROWSY</td>
<td>5.4</td>
<td>9.2</td>
</tr>
<tr>
<td>SECONDARY</td>
<td>NORMAL</td>
<td>7.7</td>
<td>9.5</td>
</tr>
<tr>
<td></td>
<td>AGGRESSIVE</td>
<td>1.1</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>DROWSY</td>
<td>1.2</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>NORMAL</td>
<td>0.6</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>AGGRESSIVE</td>
<td>7.3</td>
<td>9.6</td>
</tr>
<tr>
<td></td>
<td>DROWSY</td>
<td>2.1</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>NORMAL</td>
<td>1.9</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>AGGRESSIVE</td>
<td>1.9</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>DROWSY</td>
<td>6.2</td>
<td>9.3</td>
</tr>
</tbody>
</table>

• The LSTM clearly outperformed simple RNNs.
• This may be because of the LSTM's greater ability to make use of long time context.
Experiment and Results: Driver based

- Video Data Driven[10]
- 1.4K trimmed video clips from movies
Experiment and Results: Driver based

- **Model**
  - LSTM
  - C3D - A Direct Spatio-Temporal Model
  - SVM

![Diagram showing the process of emotion classification using LSTM based on driving behavior](image)
### Experiment and Results: Driver based

- **Result**
- **Baseline —— 40.47%**

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Fc6</th>
<th>Fc7</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-LSTM-OneLayer</td>
<td>128</td>
<td>45.43</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>44.39</td>
</tr>
<tr>
<td>VGG-LSTM-Two-Layers</td>
<td>256-256</td>
<td>41.26</td>
</tr>
<tr>
<td>CaffeNet-LSTM-OneLayer</td>
<td>128</td>
<td>40.99</td>
</tr>
</tbody>
</table>
Experiment and Results: Driver based

- Demo
Experiment and Results: Driver based

- Experiment II: Distraction Driving detection
- 30 participants had driven at least 10,000 kilometres in 12 months
- Data
Experiment and Results: Driver based

- Experiment II: Distraction Driving detection [8]
- 30 participants had driven at least 10,000 kilometres in 12 months
- Data (After correlation-based feature subset selection):
  - speed (SP)
  - steering wheel angle (SA)
  - throttle position (TP)
  - heading angle (HA, angle between the longitudinal axis of the vehicle and the tangent on the center line of the street)
  - lateral deviation (LD, deviation of the center of the car from the middle of the traffic lane)
  - head rotation (HR, rotation around the vertical axis of the car)
Experiment and Results: Driver based

- Experiment II: Distraction Driving detection
- Model:
  - Data collection
  - Statistical Processing
  - LSTM
  - Softmax
### Experiment and Results: Driver based

- Experiment II: Distraction Driving detection [8]
- Result

<table>
<thead>
<tr>
<th>features</th>
<th>classes</th>
<th>LSTMRNN accuracy</th>
<th>recall</th>
<th>precision</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>low-level sig.</td>
<td>2</td>
<td>91.6 %</td>
<td>89.7 %</td>
<td>90.8 %</td>
<td>90.1 %</td>
</tr>
<tr>
<td>low-level sig.</td>
<td>3</td>
<td>54.4 %</td>
<td>62.1 %</td>
<td>63.0 %</td>
<td>62.0 %</td>
</tr>
<tr>
<td>low-level sig.</td>
<td>6</td>
<td>43.3 %</td>
<td>39.0 %</td>
<td>38.7 %</td>
<td>38.1 %</td>
</tr>
<tr>
<td>functionals</td>
<td>2</td>
<td>96.6 %</td>
<td>95.0 %</td>
<td>97.2 %</td>
<td>96.0 %</td>
</tr>
<tr>
<td>functionals</td>
<td>3</td>
<td>60.4 %</td>
<td>70.2 %</td>
<td>70.1 %</td>
<td>70.1 %</td>
</tr>
<tr>
<td>functionals</td>
<td>6</td>
<td>45.4 %</td>
<td>42.6 %</td>
<td>41.0 %</td>
<td>40.7 %</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>features</th>
<th>classes</th>
<th>RNN accuracy</th>
<th>recall</th>
<th>precision</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>low-level sig.</td>
<td>2</td>
<td>74.6 %</td>
<td>60.0 %</td>
<td>68.3 %</td>
<td>63.2 %</td>
</tr>
<tr>
<td>low-level sig.</td>
<td>3</td>
<td>42.1 %</td>
<td>46.6 %</td>
<td>46.4 %</td>
<td>45.6 %</td>
</tr>
<tr>
<td>low-level sig.</td>
<td>6</td>
<td>37.8 %</td>
<td>30.9 %</td>
<td>30.6 %</td>
<td>29.5 %</td>
</tr>
<tr>
<td>functionals</td>
<td>2</td>
<td>94.9 %</td>
<td>92.9 %</td>
<td>95.0 %</td>
<td>93.8 %</td>
</tr>
<tr>
<td>functionals</td>
<td>3</td>
<td>62.5 %</td>
<td>67.9 %</td>
<td>65.7 %</td>
<td>66.5 %</td>
</tr>
<tr>
<td>functionals</td>
<td>6</td>
<td>44.7 %</td>
<td>41.4 %</td>
<td>36.4 %</td>
<td>38.0 %</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>features</th>
<th>classes</th>
<th>SVM accuracy</th>
<th>recall</th>
<th>precision</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>functionals</td>
<td>2</td>
<td>91.8 %</td>
<td>88.0 %</td>
<td>90.6 %</td>
<td>89.1 %</td>
</tr>
<tr>
<td>functionals</td>
<td>3</td>
<td>61.6 %</td>
<td>65.8 %</td>
<td>64.6 %</td>
<td>64.9 %</td>
</tr>
<tr>
<td>functionals</td>
<td>6</td>
<td>43.5 %</td>
<td>39.2 %</td>
<td>35.2 %</td>
<td>36.7 %</td>
</tr>
</tbody>
</table>
Experiment and Results: Driver based

- Experiment II: Distraction Driving detection [8]
- Problems:
  - Specific training condition
  - Bidirectional Long Short-Term Memory (BLSTM)
  - examine hybrid fusion of the low-level data streams
5. Robustness and Proposal
Robustness and Proposal

- Fusion-RNN: Sensory Fusion RNN with LSTM units
  - LSTM to solve vanishing gradient
  - by concatenating the streams
  - Performs poorly as does not capture the rich context for modelling

Solution?
Sensory Fusion Layer
Robustness and Proposal

- sensory fusion layer combines the high-level representations of sensor data.
- passes two sensory streams \{(x_1, \ldots, x_T), (z_1, \ldots, z_T)\} through separate RNNs.

\[
(h_t^x, c_t^x) = \text{LSTM}_x(x_t, h_{t-1}^x, c_{t-1}^x) \\
(h_t^z, c_t^z) = \text{LSTM}_z(z_t, h_{t-1}^z, c_{t-1}^z)
\]

Sensory fusion:
\[
e_t = \tanh(W_f [h_t^x; h_t^z] + b_f) \\
y_t = \text{softmax}(W_y e_t + b_y)
\]
Robustness and Proposal

- Visual Feature Extraction
- baseline video feature extractor
  - Local Binary Patterns (LBP)
Robustness and Proposal

- **Visual Feature Extraction**
  - baseline video feature extractor
    - Local Binary Patterns (LBP)
  - Proposed video feature extractor [12]
    - Optical flow
Robustness and Proposal

- Visual Feature Extraction
  - baseline video feature extractor
    - Local Binary Patterns (LBP)
  - Proposed video feature extractor[12]
    - Optical flow

<table>
<thead>
<tr>
<th>classifier</th>
<th>features</th>
<th>AROUSAL</th>
<th>EXPECTATION</th>
<th>POWER</th>
<th>VALENCE</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>WA</td>
<td>UA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLSTM</td>
<td>V</td>
<td>59.8</td>
<td>58.8</td>
<td>66.2</td>
<td>50.1</td>
<td>64.1</td>
</tr>
<tr>
<td>LSTM</td>
<td>V</td>
<td>62.7</td>
<td>61.5</td>
<td>66.0</td>
<td>50.1</td>
<td>70.2</td>
</tr>
<tr>
<td>BLSTM</td>
<td>V</td>
<td>43.1</td>
<td>42.9</td>
<td>68.6</td>
<td>62.0</td>
<td>51.7</td>
</tr>
<tr>
<td>LSTM</td>
<td>V</td>
<td>48.6</td>
<td>48.7</td>
<td>65.6</td>
<td>60.2</td>
<td>60.8</td>
</tr>
<tr>
<td>SVM [52]</td>
<td>V</td>
<td>47.8</td>
<td>47.4</td>
<td>62.0</td>
<td>54.8</td>
<td>69.6</td>
</tr>
<tr>
<td>LDCRF [33]</td>
<td>V</td>
<td>53.2</td>
<td>53.1</td>
<td>46.8</td>
<td>43.2</td>
<td>59.3</td>
</tr>
<tr>
<td>Baseline</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Robustness and Proposal

- Multimodal
- RECOLA dataset [13]: audio, video, electro-cardiogram (ECG) and electro-dermal activity (EDA) modalities
- Model
Robustness and Proposal

- Multimodal
- RECOLA dataset [13]: audio, video, electro-cardiogram (ECG) and electro-dermal activity (EDA) modalities
- Model:
  - Fusion layer

\[ m_t = \tanh(W_m [a_t, s_t] + b_m) \]

where \( W_m \) and \( b_m \) is the weight and bias in this layer.
Robustness and Proposal

- Multimodal
- RECOLA dataset [13]: audio, video, electro-cardiogram (ECG) and electro-dermal activity (EDA) modalities
- Model:
  - Fusion layer

\[ m_t = \tanh(W_m [a_t, s_t] + b_m) \]

where \( W_m \) and \( b_m \) is the weight and bias in this layer.
Robustness and Proposal

- **Multimodal**
- RECOLA dataset [13]: audio, video, electro-cardiogram (ECG) and electro-dermal activity (EDA) modalities
- **Result**

Table 3. Performance comparisons with the proposed regression model and with different feature set fused in the feature level for the AVEC 2015 training set and development set. The Lgb, Geo, Aud, Lan, Fac are short for LGBP-TOP, geometric, audio, landmarks, face-CNN respectively.

<table>
<thead>
<tr>
<th></th>
<th>Training</th>
<th>Development</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Arousal</td>
<td>Valence</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>CCC</td>
</tr>
<tr>
<td>Lgb+Geo</td>
<td>0.146</td>
<td>0.584</td>
</tr>
<tr>
<td>Aud + Lgb + Geo</td>
<td>0.114</td>
<td>0.808</td>
</tr>
<tr>
<td>Aud + Lgb + Geo + Lan</td>
<td>0.109</td>
<td>0.821</td>
</tr>
<tr>
<td>Aud + Lgb + Geo + Lan + ECG</td>
<td>0.112</td>
<td>0.816</td>
</tr>
<tr>
<td>Aud + Lgb + Geo + Lan + ECG + EDA</td>
<td>0.122</td>
<td>0.798</td>
</tr>
<tr>
<td>Aud + Lgb + Geo + Lan + ECG + EDA + Fac</td>
<td>0.115</td>
<td>0.813</td>
</tr>
</tbody>
</table>
Robustness and Proposal

- **Attention-based Model**
- A neural attention mechanism equips a neural network with the ability to focus on a subset of its inputs
Robustness and Proposal

- **Attention-based Model**
  - A neural attention mechanism equips a neural network with the ability to focus on a subset of its inputs
  - Apply in NLP a lot
Robustness and Proposal

- **Attention-based Model**
- A neural attention mechanism equips a neural network with the ability to focus on a subset of its inputs
- Apply in NLP a lot

\[
\alpha_{ts} = \frac{\exp\left(\text{score}(h_t, \bar{h}_s)\right)}{\sum_{s'=1}^{S} \exp\left(\text{score}(h_t, \bar{h}_{s'})\right)}
\]

[Attention weights]
Robustness and Proposal

- **Attention-based Model**
- **Example in Driving behavior and LSTM[14]**
Robustness and Proposal

- Attention-based Model
- Example in Driving behavior and LSTM

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>AUC</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>Mean</td>
</tr>
<tr>
<td>KNN</td>
<td>0.9033</td>
<td>0</td>
<td>0.947</td>
</tr>
<tr>
<td>DecisionTree</td>
<td>0.8543</td>
<td>0.0062</td>
<td>0.9213</td>
</tr>
<tr>
<td>RandomForest</td>
<td>0.8739</td>
<td>0.0044</td>
<td>0.9359</td>
</tr>
<tr>
<td>DeepConvGRU-Attention</td>
<td>0.9836</td>
<td>0.0015</td>
<td>0.9978</td>
</tr>
<tr>
<td>DeepConvLSTM-Attention</td>
<td>0.9786</td>
<td>0.0068</td>
<td>0.9978</td>
</tr>
<tr>
<td>DeepConvGRU</td>
<td>0.9772</td>
<td>0.0062</td>
<td>0.9968</td>
</tr>
<tr>
<td>DeepConvLSTM</td>
<td>0.9519</td>
<td>0.0186</td>
<td>0.9944</td>
</tr>
<tr>
<td>CNN</td>
<td>0.9568</td>
<td>0.0072</td>
<td>0.9984</td>
</tr>
<tr>
<td>LSTM-15</td>
<td>0.993</td>
<td>0.0015</td>
<td>0.9996</td>
</tr>
<tr>
<td>DNN-45</td>
<td>0.9395</td>
<td>0.0358</td>
<td>0.9682</td>
</tr>
</tbody>
</table>

$T_x = 60$, $\Delta t = 6$
6. Conclusion
Conclusion

• Conclusion:
  - LSTM outperforms in the Driving behavior temporal sequence analysis
  - Driving emotion recognition rely on a lot of factors and deep learning make it possible to combine

• Future Work
  - Attention
  - Biosignal+video+vehicle data
  - Variants of LSTM, e.g Bilstm, conv-lstm......
Reference


[2] Orit Taubman-Ben-Ari a∗, Mario Mikulincer b, Omri Gillath "The multidimensional driving style inventory—scale construct and validation 


