1. Neural Nets
2. Deep Convolutional Neural Networks
3. Modern Deep Architectures
4. Beyond ImageNet
5. Generative models, GANs
6. Segmentation
7. Deep learning use-case: Autonomous Driving Systems
Outline

1. Self-driving systems: from modular to end-to-end learning systems
2. Need for explainability
3. Use-Case: talk to explain the decision
01 Self-driving cars
From historical modular pipelines...

- **European Eureka Prometheus**
- **DARPA Challenges**
- **Modular systems**

- **2005**
  - STANLEY
    - Race in the desert
- **2008**
  - BOSS
    - Urban driving

Stanley, Stanford 2015
From historical modular pipelines...

**Advantage:**
- Intrinsic interpretability and transparency

**Limits:**
- Rely on human heuristics
- Lack flexibility
- Hard to scale and maintain
- Prone to error propagation

Modular systems

- 2005
  - STANLEY
  - Race in the desert

- 2008
  - BOSS
  - Urban driving

DARPA Challenges

European Eureka Prometheus
From historical modular pipelines to end-to-end learning models

Modular systems

- 2005: STANLEY Race in the desert
- 2008: BOSS Urban driving

European Eureka Prometheus

DARPA Challenges

End-to-end models

- 2016: PilotNet

Deep learning revolution
Deep learning revolution

End-to-end models

PilotNet, Bojarski et al. 2016

... to end-to-end learning models
PilotNet

Neural driving system

Bojarski et al. 2016
Tesla system

Goal: stay in the correct lane safely, change lanes to follow the right trajectory
Sensors: cameras+radars
Obstacle detection: big part of the stack
Lane detection: the most important feature of self-driving cars
Pedestrian tracking => anticipate every scenario

To make it work:
Run at least 50 neural networks simultaneously

Specific deep architecture called HydraNets
• Share backbone = modified ResNet 50 with Dilated Convolutions
• Heads based on semantic segmentation — FPN/DeepLab/UNet architectures

3D views: Bird’s Eye View
Waymo (Google) system

Localization

In any location: detailed three-dimensional maps available with information such as road profiles, curbs and sidewalks, lane markers, crosswalks, traffic lights, stop signs, etc.

Waymo is leveraging the knowledge of Google Maps to do so. Rather than rely on GPS, the Waymo Driver cross-references our pre-built maps with real-time sensor data to precisely determine their location on the road.

Perception

Sensors: LiDAR, cameras, radar
The Waymo sensor suite has a 360° degree view around the vehicle and is designed to respond to objects up to up to 300 meters away (nearly three football fields).

Software: find obstacles, traffic lights, roads
Constantly scan for objects around the vehicle — pedestrians, cyclists, vehicles, road work, obstructions — and continuously read traffic controls, from traffic light color and railroad crossing gates to temporary stop signs

Machine Learning: Active learning to collect data, and AutoML to generate architectures and select the more efficient ones (accuracy and inference time)
Waymo (Google) system

Prediction
Prediction of future movements for every dynamic object on the road based on current speed and trajectory

*It understands that a vehicle will move differently than a cyclist or pedestrian. The software then uses that information to predict the many possible paths that other road users may take. Our software also takes into account how changing road conditions, such as a blocked lane, may impact the behavior of others around it.*

Model: recurrent neural networks
ML: reinforcement learning in simulators to train their agents to estimate all trajectories

Planning
Planning: generating trajectories based on feasibility, staying on the road, and avoiding collisions. The vehicles are also learn from human labelers to generate more realistic trajectories.

*Waymo's software considers all of this information as it finds an appropriate route for the vehicle to take, then selects the exact trajectory, speed, lane, and steering maneuvers needed to progress along this route safely. Because our Waymo Driver constantly monitors the environment, and predicts the future behavior of other road users in 360° degrees around our vehicles, our Driver can respond quickly and safely to any changes on the road.*
Challenges for Waymo / Tesla systems

- Waymo and Tesla are Direct competitors in the race to Level 5 Autonomy!

- Tesla’s limitation: no LiDAR

- Waymo’s limitations:
  - Waymo can’t drive without a map. They can map the whole world more precisely, but this is a huge challenge for scale.
  - Waymo’s principal vision system being made of LiDARs is also actually a problem—LiDARs are completely blind during snow, rain, or fog.
  - Consequently, Waymo drives a lot in places like Phoenix, Arizona, or San Francisco, California, where the conditions are perpetually dry and sunny. Waymo’s Notes—Recently, Waymo started to drive in a very humid Michigan, a story Miami, and a rainy Washington.

- Machine Learning, data:
  - Google cannot leverage the same fleet power, as Tesla does. Tesla put hundreds of thousands of cars in the hands of their customers and let them collect data for them. Waymo wasn’t able do that; they however have their own fleet of vehicles that could grow a lot in the recent years.
  - If you look at Tesla, they’ve already driven their autonomous vehicles in the middle of New York City and in Paris. They already have knowledge of these places thanks to their drivers. Scaling might be much easier.
To summarize: sensors, deep neural nets architectures, Learning

Neural network-based autonomous driving system framework
Challenges

Opened questions on:
- Safety,
- Certification,
- Robustness, etc

Hard to certify
Need for Understanding the net behavior: Explainability of the self-driving system decisions
Main claims of current systems:
  - Closed-loop in learning
  - Huge number of kms without pbs
=> x100 better than humans?
02 Explaining a deep driving model
Introduction: Post-hoc explanations

- Deep self-driving models (e.g. PilotNet) = black box

- How to get post-hoc explanations?

- Differs from explainable by design (modularity-inspired models exhibit some forms of interpretability, which can be enforced at different levels in the design of a driving system)
Post-hoc explanations

- Two approaches to explain such models

**Global explanations**
- Explain the behavior of a model in general, e.g. across an entire dataset

**Local explanations**
- Given a specific input, justify why the model specifically gives its prediction
Post-hoc global explanations

Aim = explain the behavior of a model in general, e.g. across an entire dataset

Ideas from the general XAI literature:

● Prototypes-based methods
  ○ Providing global explanations by selecting and aggregating multiple local explanations, i.e. find prototypes (specific data instances representing well the data) & criticisms (instances not well represented by the set of prototypes) and see the model predictions on these examples
    => Not used in AD

● Model translation techniques
  ○ Transforming an opaque neural net-work into a more interpretable model (e.g. Explanatory graph, soft decision trees, causal model)
    => Not used in AD

● Representation explanation
  ○ Analyzing the knowledge contained in the data structures of the model
Global explanation: Stratified evaluation and automatic finding of corner cases

Aim = fined-grain evaluation, finding corner cases, understand in which situations will the model fail

- Automatic testing tool to detect erroneous behaviors:

Figure 7: Sample images showing erroneous behaviors detected by DeepTest using synthetic images. For original images the arrows are marked in blue, while for the synthetic images they are marked in red. More such samples can be viewed at https://deeplearningtest.github.io/deepTest/.
Global explanation: Stratified evaluation and automatic finding of corner cases

- **Automatic testing tool to detect erroneous behaviors:**

<table>
<thead>
<tr>
<th>original</th>
<th>synthetic</th>
<th>transformations</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Original Image" /></td>
<td><img src="image2" alt="Synthetic Image" /></td>
<td>Scale:0.6; Contrast:0.4;</td>
</tr>
<tr>
<td><img src="image3" alt="Original Image" /></td>
<td><img src="image4" alt="Synthetic Image" /></td>
<td>Scale:0.6; Brightness:75;</td>
</tr>
</tbody>
</table>

Tian et al. 2018
Local explanations

Given this situation $x$, the output decision taken by the network is $y$. Why?

Three main approaches to get local explanations:

- **Input saliency visualization** = input attribution
- **Counterfactual interventions** = inferring the prediction of a model for imaginary inputs that have not been observed
- **Supervised external model** = explain by learning to provide additional information
Local explanations: Input saliency visualization

Aim = Generate a heat map (a.k.a. Saliency map) which highlights regions influencing the most the output of the model

<table>
<thead>
<tr>
<th>Aim</th>
<th>Generate a heat map (a.k.a. Saliency map) which highlights regions influencing the most the output of the model</th>
</tr>
</thead>
<tbody>
<tr>
<td>orig img + gt bb</td>
<td>mask</td>
</tr>
<tr>
<td>gradient</td>
<td>guided</td>
</tr>
<tr>
<td>contrast excitation</td>
<td>grad-CAM</td>
</tr>
<tr>
<td>occlusion</td>
<td></td>
</tr>
</tbody>
</table>

| chocolate sauce      |                                                                                                               |
| Pekinese             |                                                                                                               |
| cliff                |                                                                                                               |
| street sign          |                                                                                                               |

Fong et al. 2017
Local explanations: Input saliency visualization

Bojarski et al. 2017

Stop for Traffic Light
Stop for Pedestrian
Stop for Stop Sign
Stop for Congestion

3DResNet-TRB

Liu et al. 2019
Local explanations: Counterfactual analysis

Aim = inferring the prediction of a model for imaginary inputs that have not been observed. e.g. “If this pedestrian was not crossing the road, what decision would have been taken by the self-driving model?”
Post-hoc explanations

Local explanations == input saliency visualization only?
Driving Behavior Explanation by textual info/NLP

Goal

Human-friendly explanations for the decisions of a neural self driving system

Motivation

Increase trust

**Car:** I’m slowing down because there is a car in front of me

Enable interactions

**Car:** I’m slowing down because there is a car in front of me.
**User:** This car seems to be stopped.
**Car:** Understood, I am passing this stopped vehicle.

NeurIPS 2020 ConfidNet
03 Natural Language explanation for AD
Introduction to textual-based explanations

Supervised explanation model

Example of textual descriptions + explanations:

Ours: “The car is driving forward because there are no other cars in its lane”

Human annotator: “The car heads down the street because the street is clear.”
Overview of BEEF: BEhavior Explanation with multi-level Fusion

Goal: Explain the decisions of a self-driving system to a human user

Paper: NeurIPS Workshop on Machine Learning for Autonomous Driving ML4AD 2020

bit.ly/2VUSpGc
BEEF: Self-driving 3D-conv backbone

**Visual encoder**

\[ r = 3DCNN(\chi) \]

**Video input**

**Trajectory prediction**

\[ (\hat{x}_k, \hat{y}_k, h_k) = \text{GRU}(r \oplus g, h_{k-1}) \]

**Visual features**

**Blinker signal**

**Imitation loss**

\[ L_{\text{drive}} = \sum_{k=1}^{K} \sqrt{(x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2} \]

**Ground-truth**

**Trajectory prediction**

3DCNN = R(2+1)D (Tran et al. 2018)

5 residual blocks
BEEF: Explanation module overview

**Problem**
Different causes collapse to a same driving decision

**Intuition**
- **Mid-level features** contain perceptual information about the scene
- **High-level features** contain information of the decision
BEEF: Explanation module overview

Fusion

\[ \hat{c} = f(v^L, m) \]

Classification

\[ p = \text{softmax}(\hat{c}) \]

Fusion function (details later)

High-level decision

Intermediate perceptual features

Stop for a red light

Probability over classes
BEEF: Multi-level fusion with BLOCK

Intermediate perceptual features
E.g. “There is a red light in the image”

High-level decision
E.g. “The vehicle chooses to make a stop”

\[ \hat{c} = f(v^L, m) \]
BEEF: Multi-level fusion with BLOCK

Intermediate perceptual features
E.g. “There is a red light in the image”

High-level decision
E.g. “The vehicle chooses to make a stop”

BLOCK (Ben-Younes et al. 2019)
BEEF: parallel with Visual Question Answering

Question: Is the mustache real?

Image: [Image of a person with a mustache]

Answer: no

(Spatio-temporal activation)

Predicted trajectory: \((x_k, \hat{y}_k)_{k=1}^K\)

Explanation: Stop for a red light

(Ben-Younes et al. 2017)
BEEF: Learning

Hypothesis
Mimicking driving behavior conserves explanations → Imitation learning for explanations

Explanation loss
\[ \mathcal{L}^{\text{explain}} = - \log p[c] \]

Global loss
\[ \mathcal{L} = \mathcal{L}^{\text{drive}} + \alpha \mathcal{L}^{\text{explain}} \]
Experiments: Visualizations on HDD

Honda Deep Drive (HDD) dataset: 104 hours, with frontal camera video + blinker + GPS

Annotations: Each frame is labeled with one of the 6 causes (+background class)
### Experiments: quantitative results on HDD

<table>
<thead>
<tr>
<th>System</th>
<th>Online/Offline</th>
<th>Individual causes</th>
<th>Overall mAP</th>
<th>Driver MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Congest.</td>
<td>Sign</td>
<td>Red light</td>
</tr>
</tbody>
</table>


## Experiments: quantitative results on HDD

<table>
<thead>
<tr>
<th>System</th>
<th>Online/Offline</th>
<th>Congest.</th>
<th>Sign</th>
<th>Red light</th>
<th>Crossing vehicle</th>
<th>Parked vehicle</th>
<th>Crossing pedestrian</th>
<th>mAP</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN+Sens. (Ramanishka et al. 2018)</td>
<td>On.</td>
<td>39.72</td>
<td>46.83</td>
<td>45.31</td>
<td>—</td>
<td>7.24</td>
<td>2.15</td>
<td>28.25</td>
<td>×</td>
</tr>
<tr>
<td>I3D (Li et al. 2020)</td>
<td>Off.</td>
<td>64.8</td>
<td>71.7</td>
<td>63.6</td>
<td>21.5</td>
<td>15.8</td>
<td>26.2</td>
<td>43.9</td>
<td>×</td>
</tr>
<tr>
<td>I3D+GCN (Li et al. 2020)</td>
<td>Off.</td>
<td>74.1</td>
<td>72.4</td>
<td>76.3</td>
<td>26.9</td>
<td>20.4</td>
<td>29.0</td>
<td>49.9</td>
<td>×</td>
</tr>
</tbody>
</table>

### Action recognition (no driver)

<table>
<thead>
<tr>
<th>System</th>
<th>Overall mAP</th>
<th>Driver MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN+Sens. (Ramanishka et al. 2018)</td>
<td>28.25</td>
<td>×</td>
</tr>
<tr>
<td>I3D (Li et al. 2020)</td>
<td>43.9</td>
<td>×</td>
</tr>
<tr>
<td>I3D+GCN (Li et al. 2020)</td>
<td>49.9</td>
<td>×</td>
</tr>
</tbody>
</table>

### Driver only (no explanation)

<table>
<thead>
<tr>
<th>System</th>
<th>Overall mAP</th>
<th>Driver MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver</td>
<td>1.33</td>
<td></td>
</tr>
</tbody>
</table>

### Introspective explanation

<table>
<thead>
<tr>
<th>System</th>
<th>Overall mAP</th>
<th>Driver MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-head</td>
<td>48.39</td>
<td>1.36</td>
</tr>
<tr>
<td>BEEF</td>
<td>50.96</td>
<td>1.33</td>
</tr>
</tbody>
</table>

### Key Points

- **SOTA results**
  - Outperforming both online and offline models

- **Slight drop on some classes**
  - Advantage of accessing future frames

- **Complementarity of features**
  - Comparison to multi-head
  - Does not degrade driver MSE
Experiments: Ablations

Choice of layer L

- Too low-level
- Rich perceptual signal
- Less complementary to the decision

Other fusion methods

<table>
<thead>
<tr>
<th>Fusion model</th>
<th>mAP</th>
<th>Driver MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 3</td>
<td>45.39</td>
<td>1.38</td>
</tr>
<tr>
<td>Cat+MLP</td>
<td>46.64</td>
<td>1.38</td>
</tr>
<tr>
<td>MFB</td>
<td>39.10</td>
<td>1.43</td>
</tr>
<tr>
<td>MLB</td>
<td>38.59</td>
<td>1.37</td>
</tr>
<tr>
<td>MUTAN</td>
<td>45.80</td>
<td>1.40</td>
</tr>
<tr>
<td>Bilinear</td>
<td>49.82</td>
<td>1.42</td>
</tr>
<tr>
<td>BLOCK</td>
<td><strong>50.96</strong></td>
<td><strong>1.33</strong></td>
</tr>
</tbody>
</table>

→ Importance of modeling fine-grain correlations
→ BLOCK is the only fusion not degrading driving performances
**Extension: natural language justifications**

E.g. “the car stops as traffic ahead is stopped at a red light”

**Motivations:**
- Open-domain sentences convey finer and richer semantics than predefined classes
- Going towards human-machine dialogs

Replace the classification layer by an auto-regressive LSTM language model.

Predict driving commands (throttle and steering angle) instead of mid-level future trajectory.

Produce justifications for temporal subsequences instead of the frame-by-frame basis.

Sampling more diverse justification with the softmax temperature.
Extension: qualitative results on BDD-X (77 driving hours)

Action description: the car is stopping
Human justification: since it is approaching traffic stopped at a red light
BEEF justification: because the traffic in front of it is stopped

Other samples from BEEF (with higher decoding temperature):
- because traffic has stopped in front
- because the car in front of it is stopped
- because the traffic in front of it is stopped
- because the car in front of it has stopped
- because the traffic ahead is stopped at a red light
Extension: qualitative results on BDD-X (77 driving hours)

Action description : the cars slows down
Human justification : since the light is red
BEEF justification : because the light ahead is red

Other samples from BEEF (with higher decoding temperature):
- because the light ahead is red
- because the light ahead is red
- because the light is red
- because the light ahead is red
- because the light ahead is red
### Extension: results on BDD-X

<table>
<thead>
<tr>
<th>Task</th>
<th>Model</th>
<th>BLEU-4</th>
<th>METEOR</th>
<th>CIDEr-D</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Action recognition</strong></td>
<td>“because the light is red”</td>
<td>5.85</td>
<td>10.74</td>
<td>59.50</td>
</tr>
<tr>
<td>(no driver)</td>
<td>Vid-to-Text (Venugopalan et al. 2015)</td>
<td>6.33</td>
<td>11.19</td>
<td>53.35</td>
</tr>
<tr>
<td><strong>Rationalization</strong></td>
<td>Rationalization (Kim et al. 2018)</td>
<td>6.52</td>
<td>12.04</td>
<td>61.99</td>
</tr>
<tr>
<td>(offline)</td>
<td>Decision Features</td>
<td>9.15±.37</td>
<td>14.34±.23</td>
<td>92.08±1.3</td>
</tr>
<tr>
<td><strong>Introspective explanation</strong></td>
<td>SAA (Kim et al. 2018)</td>
<td>7.07</td>
<td>12.23</td>
<td>66.09</td>
</tr>
<tr>
<td>(offline)</td>
<td>WAA (Kim et al. 2018)</td>
<td>7.28</td>
<td>12.24</td>
<td>69.52</td>
</tr>
<tr>
<td></td>
<td>Layer 2</td>
<td>7.96</td>
<td>13.51</td>
<td>83.46</td>
</tr>
<tr>
<td></td>
<td><strong>BEEF</strong></td>
<td><strong>9.81±.32</strong></td>
<td><strong>14.79±.22</strong></td>
<td><strong>97.31±2.3</strong></td>
</tr>
</tbody>
</table>

### Extracted frame

- **GT**: because traffic is moving now
- **T=0**: because the light is green and traffic is moving
- **T=0.3**: as the light turns green and traffic is moving
- **T=0.3**: because the light is green and traffic is moving
- **T=0.3**: because traffic is moving forward
- **T=0.3**: because the light turns green
- **T=0.3**: because the light turned green and traffic is moving

- since the cars in front aren’t moving
- because the car in front has stopped
- because the light is red
- as traffic ahead is stopped at a red light
- because the car in front has stopped
- because traffic is stopped at a red light
- because traffic is stopped
- since the car is free to move right
- because the car is turning to the right
- because the car is entering another street
- to enter another road
- because it’s making a right turn
- because the road is clear of traffic
- to make a right turn