Unsupervised Test-Time Domain adaptation

Ismail Nejjar

March 2022
Unsupervised Test-Time Domain adaptation

Different from the traditional domain adaptation.

Test-time adaptation aims to adapt a model to **changing conditions** by updating to new and different data during testing without altering training or requiring more supervision.
The Purpose

The Need

Use case Road crack detection
Adaptation to Different Shifts: How to reduce generalization error on simulation-to-real discrepancies
Test-Time Adaptation: The Purpose (2)

Adaptation to Different Shifts: How to reduce generalization error on simulation-to-real discrepancies, different operating conditions
Adaptation to Different Shifts: How to reduce generalization error on simulation-to-real discrepancies, different operating conditions and other shifts?

Test-Time Adaptation: How to improve during testing without relying on training data, using solely the pre-trained model and target data.
The Need

Unsupervised Test-Time Domain adaptation

OUTLINE

The Purpose

The Need

Use case Road crack detection
Test-Time Adaptation: The Need (1)

How to reduce generalization error on new and different example at test time?
How to reduce generalization error on new and different example at test time?

**Model Performance**: A model could perform poorly without adaption

**Unavailability**: The source data are unavailable for privacy reasons or limited bandwidth.

**Efficiency**: It may not be practical to process source data during testing.
OUTLINE

The Purpose

The Need

Use case Road Crack detection

• Road Damage Dataset
• Previous Method
• Current Approach
• Results
Road Damage Dataset 2020 (RDD2020) is a real-world dataset of road damage detection collected using a smartphone under different conditions in 3 countries:

- Japan
- India
- Czech

State Highways  Local Road  Urban Road
There are 4 types of Road Damage Types denoted as:

- D00: Longitudinal linear crack
- D10: Lateral linear crack
- D20: Alligator crack
- D40: Other corruption (e.g., bump, pothole)
Road Damage Dataset 2020 (3)

Japan | India | Czech Republic
Unsupervised Test-Time Domain adaptation

OUTLINE

The Purpose

The Need

Use case Road Crack detection

- Road Damage Dataset
- Previous Method
- Current Approach
- Results
The classical Resnet architecture is structured as follows:

1. Input → Weight
2. Weight → BatchNorm
3. BatchNorm → ReLU
4. ReLU → Weight
5. Weight → BatchNorm
6. BatchNorm → ReLU
7. ReLU → Weight
8. Weight → BatchNorm
9. BatchNorm → Output

The batch normalization layers are illustrated below:

IN $\mu$ $\sigma$ $\gamma$ $\beta$ OUT

Defined by:
- Standardization part: $\mu \leftarrow \mathbb{E}[x], \sigma^2 \leftarrow \mathbb{E}[(\mu - x)^2]$
- Linear transformation: $\gamma$ and $\beta$ are learnable parameters
Adaptive batch normalization (AdaBN)

AdaBN assumes that model performance **deterioration** on the **target distribution** is caused by a **distribution gap** on the **intermediate layers**.

It proposes to replace the batch normalization statistics of the source data with those of the target data:

\[
\mu \leftarrow \mathbb{E}[x_t], \quad \sigma^2 \leftarrow \mathbb{E}[(\mu - x_t)^2]
\]

The distribution gap should be reduced in each layer:

![Graphs showing distribution gap reduction](image)

**Advantages**: AdaBN doesn’t require any additional parameters or further training. Only the batch normalization statistics are refined.
Test entropy minimization (TENT)

Tent assumes:

- Model performance **deterioration** on the target distribution is caused by a distribution gap.

- Entropy can serve as an estimate of the degree of shift.

It proposes to replace the batch normalization statistics of the source data with those of the target data, and minimize entropy of model predictions on the target dataset:

\[
\text{normalization } \mu \leftarrow \mathbb{E}[x_t], \quad \sigma^2 \leftarrow \mathbb{E}[(\mu - x_t)^2] \\
\text{transformation } \gamma \leftarrow \gamma + \frac{\partial H}{\partial \gamma}, \quad \beta \leftarrow \beta + \frac{\partial H}{\partial \beta}
\]

**Advantages:** The number of updated parameters remains relatively small compared to the total number of parameters.
**Source HypOthesis Transfer (SHOT)**

SHOT aims to learn a **domain specific feature encoding**, with respect to a fixed source classifier.

The intuition behind is the neural network is adapted to a **source-like representations** for **target data** that the classification outputs from the source classifier (hypothesis) for target data should be similar to those of source data.

The learning objective aims to minimize the entropy while maximizing the diversity of the predictions:

\[
L_{cnt}(x_t) = - \sum_c p(\hat{y}_c) \log p(y_c)
\]

\[
L_{div}(x_t) = \sum_c \bar{p}(\hat{y}_c) \log \bar{p}(y_c)
\]

where \(\bar{p}(\hat{y}_c) = \frac{1}{B} \sum_b p(\hat{y}_c)\) and \(B\) the batch-size.

**Disadvantages:** The number of trained parameters is drastically bigger than the previous methods requiring more computational power at test-time.
Use case Road Crack detection

- Road Damage Dataset
- Previous Method
- Current Approach
- Results
Proposed Method: Test-time augmentation (TTA)

Given a test image from the target domain, test-time augmentation first creates an augmented set of images for the given test image.
Proposed Method: Test-time augmentation (TTA)

The final test-time augmented predictions can be utilized in 2 different ways:

1. In the inference stage, directly use the prediction for evaluation on a given pre-trained model

2. Given the predictions and their corresponding scores

   - Define a threshold to create a confident list of pseudo-labels:
     \[ y_t = \arg\min_{c \in C} \mathbb{1}[p(\hat{y}_c) > \theta] \]

   - The pseudo labels can be used to fine-tune the pre-trained model using the cross-entropy loss

     \[ L_{\text{pseudo}}(x_t, y_t) = -\sum_c q_c \log \delta_c(f(x_t)) \]

     where \( \delta_c(x) = \frac{\exp(x_c)}{\sum_i \exp(x_i)} \) denotes the \( c \)-th element of the softmax function
Use case Road Crack detection

- Road Damage Dataset
- Previous Method
- Current Approach
- Results
We report the Average Precision (AP) for each of the four damage types as well as the Mean Average Precision (mAP) in %. 

Japan → India
We report the Average Precision (AP) for each of the four damage types as well as the Mean Average Precision (mAP) in %.
Possible improvement

Input

Prediction
Possible improvement

Input

Prediction

Segmented image of the road
Possible improvement

Input

Prediction

Segmented image of the road
Conclusion

• Test-time adaptation is a special setting of unsupervised domain adaptation where a trained model on the source domain has to adapt to the target domain without accessing source data.

• We studied the behavior of recent test-time adaptation algorithms in the presence of several domain shifts for crack detection.

• Data augmentation can be used both for boosting the results of your model at training time but also at testing time.

• There is still room for improvement!
Merci

Ismail Nejjar