FINAL PRESENTATION

• Topic •

“Car Image Classification Using neural-ODE”

2017311725 김도열 2016312632 김진우
2018310238 정채원 2017312379 주재현

Agenda

0 Executive Summary

1 Project Background

2 Goal & New Perspective of the project

3 Dataset

4 Data Preprocessing

5 Model Designing

6 Training & Optimization

7 Evaluation

8 Implementation Issues & Interests

Executive Summary

: For **minimizing memory usage** & **Improving model**

: Cropping object using **YOLOv5** & Resizing for model input shape

: Our Target Hyperparameters -> **Epoch & Learning Rate**

A. Ideation ~4/11

B. Data Collecting ~4/18

: **64,879 pictures** of Korean car collected by KIST

C. Data Preprocessing ~5/2

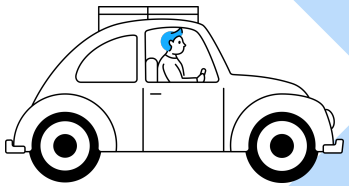
D. Model Designing ~5/9

: **ResNet-18** for baseline model & Model1,2 related to **ODE**

E. Training & Optimization ~5/13

F. Evaluation ~5/16

: Using **Cross-Validation** & Calculating **F1 Score**



We made "Car Image Classifier" using neural ODE for 2 months.

Project Background

Perspectives

1 BUSINESS PERSPECTIVE

- It is planned based on **the lack of information on preferred cars**.
- Especially, the fact that it is difficult to remember specific inform ation such as the year and model by looking at your preferred car suggests raises **the need for ML** in the end.
- Therefore, we implemented **a detector system** that provides. **more accurate vehicle information** based on the characteristics of vehicle images and use different types of techniques to **Improve performance**.

2 TECHNICAL PERSPECTIVE

- Can be classified into automotive detection system and image classification model.
- Analyze existing **CNN-based classification models** in terms of classification models.
- After applying various applicable new techniques, the optimal model will be implemented by comparing them.
- For detection systems, we used the **YOLOv5 model**.

We've thought largely from two perspectives: **business and technology**.

Goal & new perspective of the project

Goal & New Perspective

1 GOAL OF OUR PROJECT

- The aim of the project is to **make the car classifier model** which is utilized by ODE block, ResNet and YOLOv5.
- At that time, Input data -> Photo of the car
  Output data -> what kind of the car it is.
- Also, we minimize the memory usage and improve the model.

2 NEW PERSPECTIVE OF OUR PROJECT

<DATA>

- Existing research data : Focus on the front or back of a car
  Data that we used : Photos from different directions
  (Also, it's government data, it has a huge amount of data.)
  -> Provides more accurate automotive recognition

<TECHNUQUES>

- Technically, we will use **yolo5 model + ODE**
  -> Two advantages
  1) Lower memory cost by Adjoint sensitivity method
  2) Fitting appropriate with irregular time points data

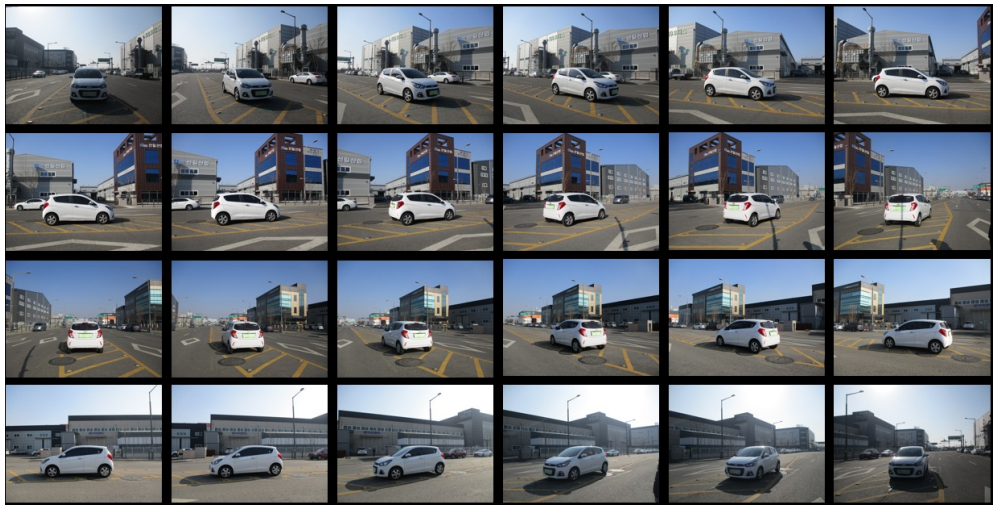
We aimed to improve the problems of existing research,

creating an advanced model over existing automotive classifier models.

Dataset

Dataset

SPEC> Illuminance: at 10AM, 2PM, 4PM, 6PM and inside
angles: 0 degree, 15 degree... 345 degree



Chevrolet Next Spark 2016 at 10AM

- The original dataset is the big data of Korean car, which consists of 64879 pictures of cars.
- The data is collected by KIST(Korea Institute of Science and Technology).
- The dataset is about 100 different vehicles. There are an average 650 photos of each car.
- The data includes 100 kinds of car that is taken in various illuminance and angles.

Why did we select this dataset?

Data Preprocessing

Process

Data Preprocessing is done in two steps.

1) Cropping main object using YOLOv5

YOLOv5 is a new approach to object detection. It can be optimized end-to-end directly on detection purpose. It is much faster algorithm for detecting objects.

We deleted the non-detected car picture by deleting the small sized picture because most of the car picture that is processed normally has big size.

2) Resizing for model input shape

Preprocess models are resized into square shape with 224px length.

The shape fixing is motivated from MNIST dataset, which has square image of one channel. For our project, 3 channels with RGB are used.

Data Preprocessing

Q&A

Q. Why not using single Network?

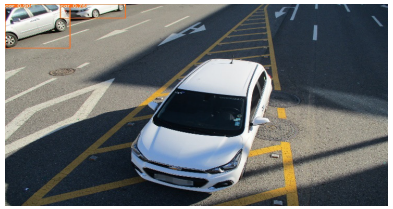
A. There are three reasons.

1. Model's Novelty
2. Hardness of Making trainset
3. Problem of dataset

[Errors in the acutal dataset]

YOLO is wrong

Not whole car is taken by photo



There are too many cars

Model Designing

Models

1 Baseline Model

layer name	output size	18-layer	34-layer
conv1	112×112		
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$
	1×1		av
FLOPs		1.8×10^9	3.6×10^9

Used ResNet-18

2 Model 1

Down Sample
Same Size (ODE-Block, Euler)
Down Sample
Same Size (ODE-Block, Euler)
Down Sample
Same Size (ODE-Block, Euler)
Down Sample
Same Size (ODE-Block, Euler)

For analyzing the performance of neural-ODE's weight update algorithm

3 Model 2

Down Sample
Down Sample
Down Sample
Down Sample
Same Size
(ODE-Block, Dopri5)

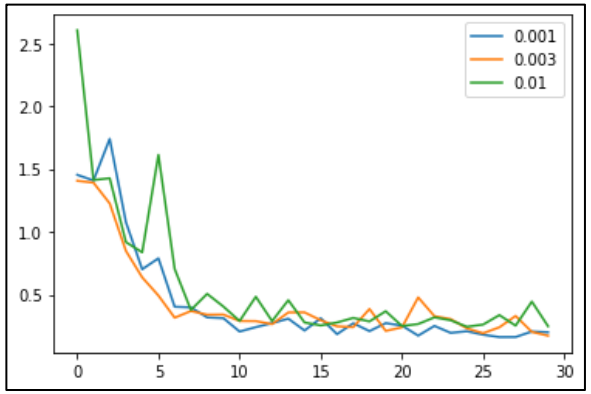
Piled all down-sampling residual blocks at front, then piled non-down-sampling residual blocks at backward.

Training & Optimization

Our Target Hyperparameters

1 Learning Rate

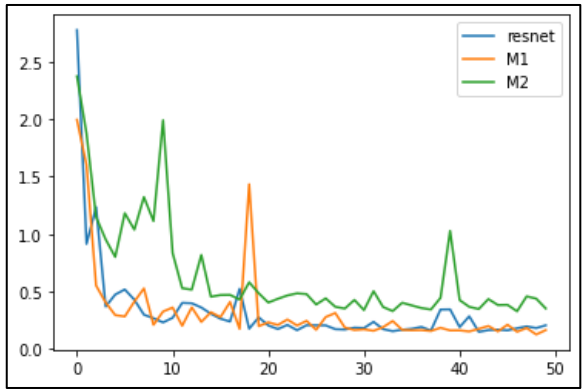
<K-fold Validation, Model 1>



For selecting best epoch, we rejected K-fold idea, and just **split dataset** into train and test dataset.

2 Epoch

<Test loss of all models>



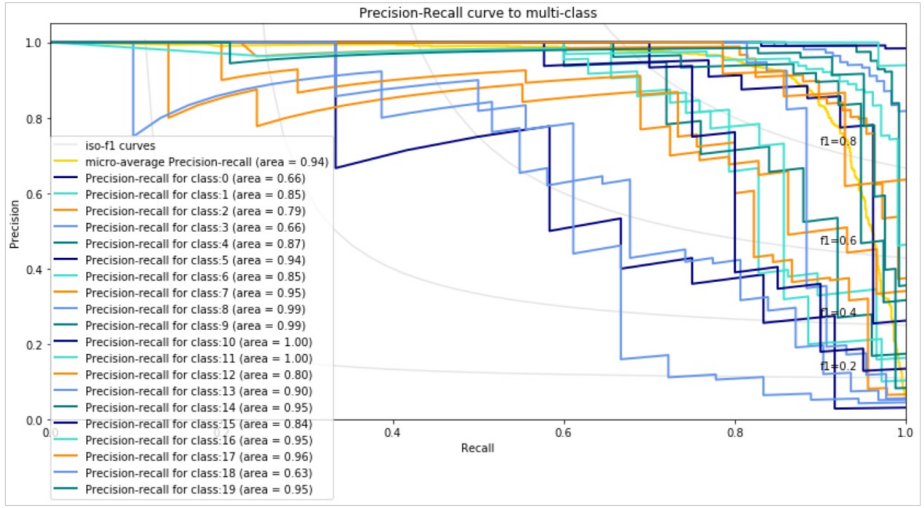
<Best Result>

	Epoch	Best loss	Best Acc
ResNet-18	43	0.1493	96.77%
Model 1	49	0.1265	97.11%
Model 2	47	0.3285	91.88%

Evaluation

Evaluation Methods

$$F_{1,macro} = 2\frac{\text{recall}_{macro} \times \text{precision}_{macro}}{\text{recall}_{macro} + \text{precision}_{macro}}$$

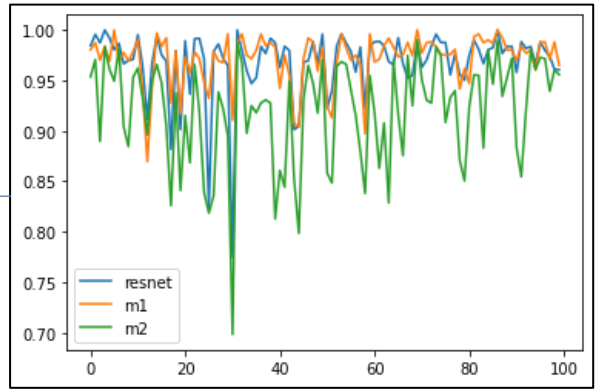


- Evaluation can be done in two perspectives, **accuracy and unknown class detection**.
- For accuracy, AUC area and f1 score are mainly used for classification task evaluations.
- Variances of the f1 score can be used to handle unbalanced predictions, and covariances of the vehicle model portion and the F1 score can be used for services.
- For unknown class detection, it also can be evaluated by f1 score, treating unknown class as another new class.

Evaluation

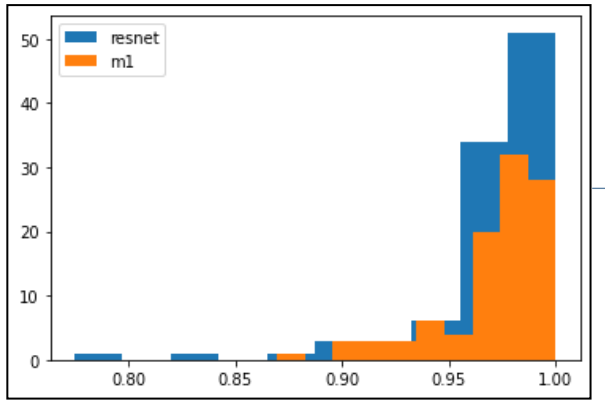
Evaluation Metric

<F1-score by classes and models>



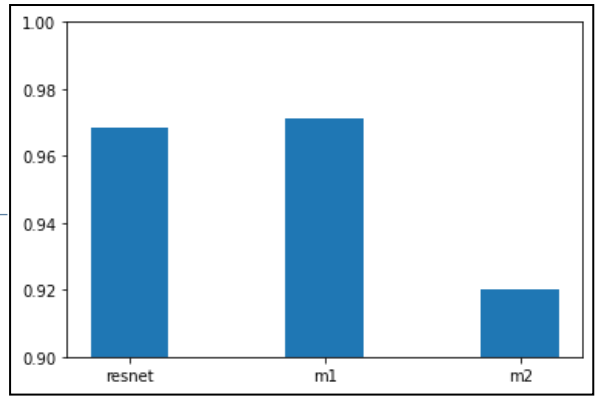
Each class will have f1-score

<F1-score, histogram>



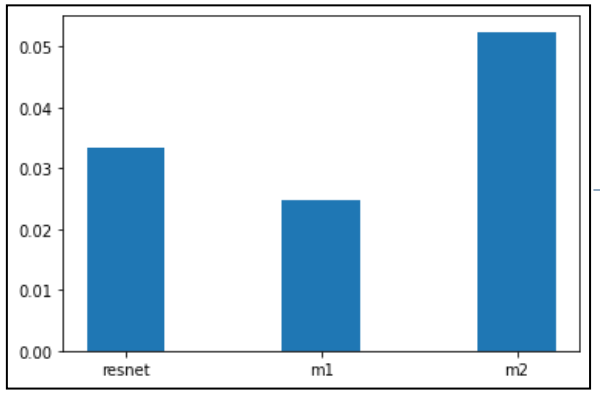
The distribution Model 1 is slightly better than baseline model

<F1-score, mean>



Model 1 showed slightly better result than baseline, just like accuracy

<F1-score, standard deviation>



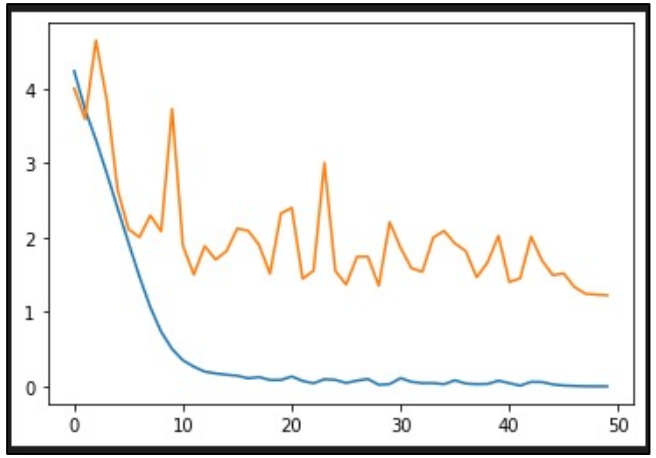
The standard deviation of F1-score for each model are Shown at right figure

Implementation Issues & Interests

Issues & Interests

1 Data Partitioning

<Training with Data Partitioning Problem>



- As we used random split for dataset, there was **some unfair distribution** between classes.
- After noticing the inefficiency problem, we changed data partitioning method with **using split method of sklearn library.**

2 Neural-ODE Function

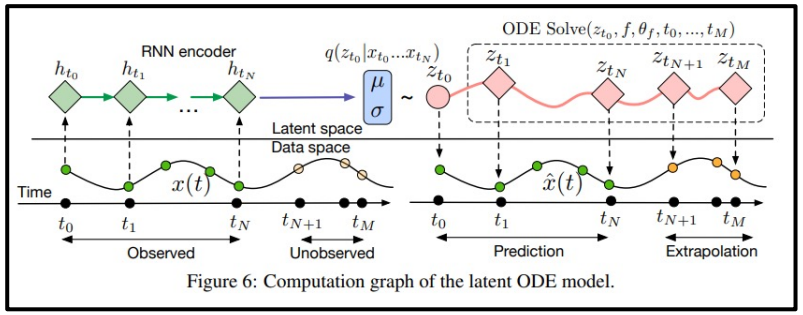


Figure 6: Computation graph of the latent ODE model.

- Due to lack understanding about neural ODE, we made a mistake at model implementation. For Model 1, we replaced the residual block into ODE block, which uses default ODE solver method, **Dopri5.**
- Interesting point is that it took a very long time for each training, but it was able to train in the environment.

End of Document

Thank You ☺