Pattern Recognition
Convolutional Neural Networks
Project

Course Instructor: Dr. Morteza Analoui
Teaching Assistant: Morteza Hajiabadi
Iran University of Science and Technology, School of Computer Engineering
First Semester of the Academic Year 1402-1403

---

1. Abstract

Convolutional Neural Networks (CNNs) are a category of deep learning models specifically designed for processing network-like data, such as images. They have been highly successful in various computer vision tasks, including image classification, object detection, and semantic segmentation. In this project, you will implement a model based on Convolutional Neural Networks for the classification of traffic signs and driving guidance. You will learn about the key components of CNNs, experiment with different architectures and techniques, and gain practical experience in applying them to a real-world problem.

2. Dataset

Use the dataset of traffic signs and driving guidance, which includes images

of traffic signs and driving guidance. You can download the dataset from here: Dataset Link The Excel file 6 provided in this dataset has some issues. You can address these issues yourself. To save time, it is recommended to use the file provided alongside the project description (this file).

3. Project description

In this section, an attempt has been made to explain the project steps in order. Please proceed according to the instructions provided.

3.1. Data preprocessing

- Start by exploring the dataset and understanding the different classes of traffic signs and driving guidance.

Be sure to write explanations about the dataset at the beginning of your report file.

- Optionally, display a few examples of images from the training dataset along with their labels. For example, it could be as follows:

Figure 1: sample dataset pictures

- Divide the dataset into training subsets, validation, and test. For this dataset, you can split the data in the 'train' folder into two sections: 'training' and 'validation' with an 80-20 ratio. For testing, use the data available in the 'test' folder.
- To increase the diversity of the dataset, use data augmentation 10. Be sure to use methods suitable for the given dataset. Mention the rationale for using each of the methods in the report.

- Instead of using all the classes available in the dataset, use only the first 10 classes (Class 0 to 9).

3.2. Model architure

- Design a convolutional neural network architecture comprising convolutional layers 11, pooling layers, and fully connected layers .
- When designing the CNN architecture, start with a simple model and gradually increase its complexity as needed.
- To find the best combination for the task, perform the learning process with various hyperparameters , such as the number of layers, filter size, and learning rate .
- Utilize normalization techniques such as batch normalization.
- Initialize the weights using appropriate methods (e.g., Xavier or He initialization).
- Explain the reason for using each of the layers in the report.

(Optional) In the useful links section, two links titled "Model Visualization" have been provided. If you are interested and have implemented your model using PyTorch, you can use these links to visualize your model and become more familiar with its structure.

3.3. Training and Optimization

- Choose an optimization algorithm (e.g., SGD, Adam) and a learning rate scheduler for training the model. Write down the reason for your choice of the optimization algorithm.
- Train the model using the training dataset and validate its performance using the validation dataset.
- Monitor training and validation for loss and accuracy to prevent overfitting. Employ early stopping or Dropout if necessary. Feel free to use any other ideas for this purpose. The model should achieve a minimum accuracy of 80%.

3.4. Transfer Learning

Experiment with transfer learning using pretrained models such as DenseNet, ResNet, VGG, and fine-tune them for classification. Modify your initial model to suit the new dataset for classification. Finally, train your model by freezing an appropriate number of layers and report the results.

3.5. Evaluation and Visualization

- Evaluate the model performance on the test set using metrics such as accuracy, precision, recall, and F1 score 1 in sections 3-3 and 4-3. Specify which metric is more suitable for this problem.
- Calculate the confusion matrix and provide your analysis of it.
- Display the predicted class for a limited number of samples from the test dataset. For example, it could be as follows:

Figure 2: Sample Images Predicted by the Trained Model

4

- Visualize the features 30 of the network to understand what the model has learned. Perform this for a selected sample as desired.
- A few points for visualizing the features of convolutional neural networks:

Visualize the output feature maps for the convolutional layers for a specific input image. This can help you see how the model processes input data and which features become active in different layers. Feature maps result from applying learned filters to the input data. For example, for the given image, feature maps could be as follows:

Table 1: Output Feature Maps for Some of the Layers

Layer 1 - Filter 72	Layer 1 - Filter 72	Layer 1 - Filter 72	Input picture
			

In a CNN, each Conv layer has multiple pattern-applying filters that maximize their output when a similar pattern is found in the input image. Identify the input images or regions that create the maximum activation for a specific filter. This process can help you understand what types of patterns the filter is sensitive to and how it contributes to the decision-making process of the model. Then compare the results with the confusion matrix and write down your observations. For example, for the image in Table 1, we arrive at the following results:

5

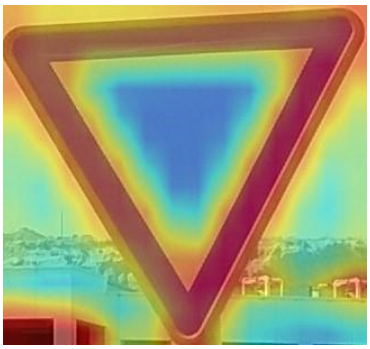

Figure 3: Maximum Activation for a Sample Image. As the color tends towards red, it indicates the network's higher attention to that area. Conversely, as it approaches blue, it means the network is less sensitive to those regions.

- Analyze the challenges of convolutional neural networks and if you have any solutions for improvement, please write them down.

3.6. Optional Section

- Differences in Class Distribution: A common challenge in machine learning is the potential imbalance in the number of samples for some classes. Examine the impact of differences in class distribution on the performance of your model.
- Calculate the distribution of samples in different classes of traffic signs in the dataset.
- Implement a class balancing strategy such as oversampling, undersampling, or class weighting to address the issue of differences in class distribution. Choose a strategy and apply it to the training set.
- Train your model based on the balanced training set and evaluate its performance on the test set.

- In your report, address the following points and provide your opinion:

  The impact of differences in class distribution on the performance of the initial model.

  The class balancing strategy you have chosen and its impact on the model's performance.

  Any observations or insights derived from comparing the results of the initial and balanced models.

Research which area of image processing and computer vision is more prone to the issue of class imbalance. What solutions have been used to address this issue?

4. Additional Points

- Prepare a comprehensive report on the implementation and the questions posed. A significant portion of your grade depends on your report, so pay careful attention to its writing. The report should be typed and submitted in PDF format.
- Please compress the report file and the written code into a single compressed file named YourStudentID.zip and upload it.
- The evaluation of the project is based on the correctness of the solutions, a suitable report, the optimization of the code, and avoiding plagiarism.
- Collaboration on the project is not allowed, but you can discuss and exchange ideas with each other.
- Please submit the relevant code in the .ipynb format, and implement each of the requested items in a new cell. Additionally, share your trained

model with the email provided at the end of this section. The model name should be in the format 'YourStudentID'.

- "Different sections of the code should be well-commented and understandable for another person who sees your code; incomplete or incorrect commenting and implementation explanations in the report will result in a deduction of points."
- In case of using a piece of ready-made code, its source must be mentioned. In the report, you should fully explain and be proficient in any code that you present. This section is to ensure that if your code is similar to another source, you won't be penalized. Failure to cite the source and similarity to another code will be considered as plagiarism, resulting in a zero score for your assignment.

- To run the project, you can use PyTorch.
- You can ask your questions in the Telegram group or reach out to the respective assistant quickly by messaging the following ID on Telegram.
  @MortezaHajiabadi

If you do not have access to Telegram, you can communicate through the following email address.

*hajiabadi*1377*@gmail.com*

5. Useful links

- model visualization 1
- model visualization 2