**Project summary**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Question** | **Answer** | **Details (if any)** |
| 1 | Briefly state the highlights of your project | - Identify music notes from pictures of monophonic scores  - Implement convolutional recurrent neural network for the project  - Use connectionist temporal classification loss to regulate the training process | End-to-end image recognition project that reads a line of music sheet and returns the sequence of musical notes in it |
| 2 | The newness, creativity of the project (Is this a new idea? If it is an old idea, what makes your work different from other things that people have done?...) | - While there’s nothing new about the project, few research have been carried out to solve this problem.  - This project is an improvement on one of the most popular papers in this field. |  |
| 3 | Did you collect and annotate new dataset? | NO |  |
| 4 | Which benchmark datasets did you use? | PrIMuS | There are 87,678 samples available in the original project, each placed in a folder containing two types of input image (normal and distorted) and other files for labels |
| 5 | Did you perform any pre-processing steps? | YES | - The image is augmented  - The label sequence is converted to a tensor of numerical values |
| 6 | Did you use any image processing techniques and/or low-level and mid-level CV algorithms? | YES | - Scaling  - Normalization  - Padding |
| 7 | Did you use different approaches to solve the problem? | Multitask learning is implemented | Using the same input, the new model with give two outputs for both kinds of label at once and backpropagate from the combined loss of the two outputs |
| 8 | Did you propose a new method or modify/improve any method to solve your problem? | YES | - The project is rewritten in Pytorch for faster runtime  - Multitask learning is implemented to improve result |
| 9 | Did you train the model or just use a pre-trained model for inference? | We trained the model ourselves | Since the original model was written in a deprecated framework, we didn’t implement it for cross reference |
| 10 | Did you train the model from scratch or using pre-trained weights? | We trained the models from scratch |  |
| 11 | Did you implement the method? Or just clone from GitHub? | We implemented the method using a new framework with reference from another work | We referenced the CRNN Pytorch model for text recognition in this project |
| 12 | How many percentages of the project’s code did you implement by yourself (without copying from other sources)? | 60% |  |
| 13 | Did you deploy the trained model on any platform (such as web, mobile app…)? | NO |  |
| 14 | Did you implement any backend, DB? | Cudnn backend is used for faster training time |  |
| 15 | Please describe the task list completed by each member | Vũ Công Duy:  - Implementing models  - CTC loss  Đào Hồng Quân:  - Data preprocessing  - CRNN  Trần Công Minh:  - Output conversion  - Optimization  Dương Thị Huê:  - Evaluation metrics  - CNN  Trần Thị Hằng:  - Evaluation metrics  - RNN |  |
| 16 | Please estimate the contribution of each member | Vũ Công Duy: 34%  Đào Hồng Quân: 17%  Trần Công Minh: 17%  Dương Thị Huê: 16%  Trần Thị Hằng:16% |  |
| 17 | How do you estimate your contribution? | Vũ Công Duy: 10  Đào Hồng Quân: 9  Trần Công Minh: 9  Dương Thị Huê: 8  Trần Thị Hằng:8 |  |