Deep Learning Based Image Classification and CNN Model Training using PaddlePaddle

Alex Shi, David Liu, Emily Luo
Stevenson High School, 1 Stevenson Dr. Lincolnshire, IL 60089

ABSTRACT

Image recognition and classification can be achieved using recent deep learning techniques to distinguishing images in different categories based on their semantic meaning. It has applications in many areas such as face recognition, intelligent video analysis in security systems, traffic scene recognition in transportation systems, content-based image retrieval and automatic photo indexing in web services, image classification in medicine, etc.

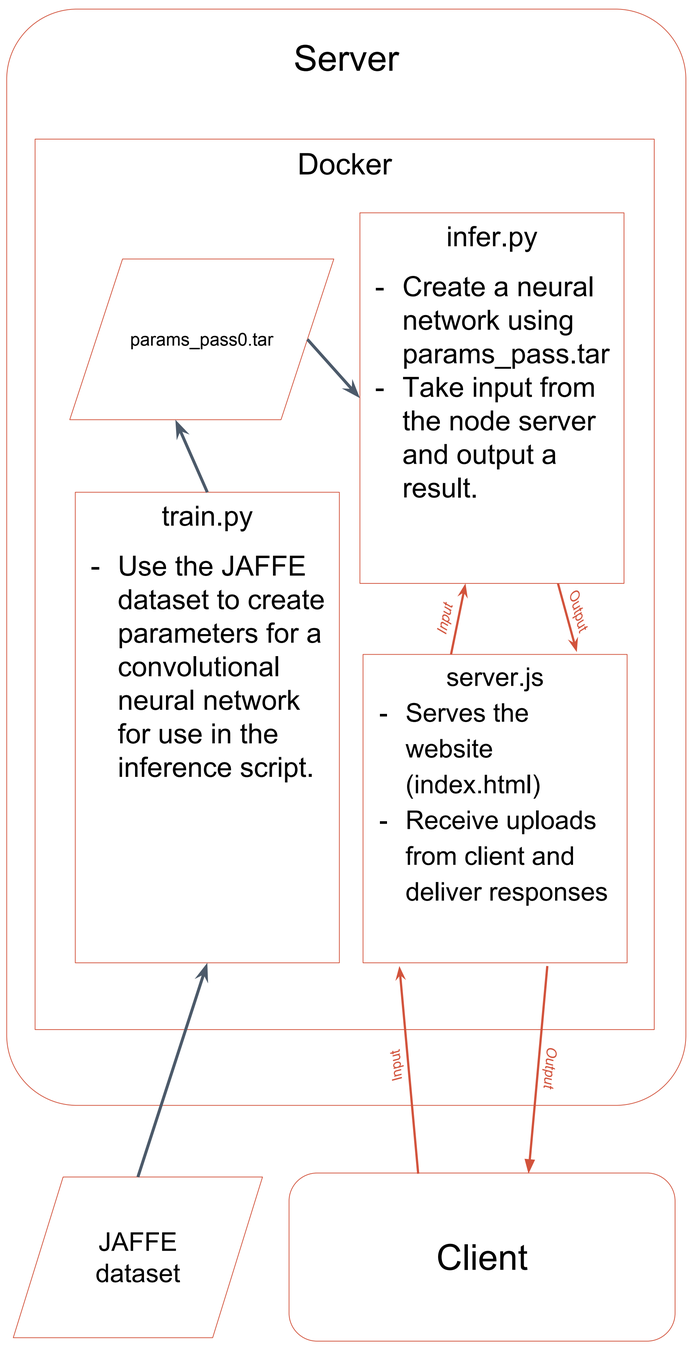
Our focus is to use Convolutional Neural Networks (CNNs) model to use raw image pixels as input, extract low-level and high-level abstract features through convolution operations, and directly output the classification results from the model utilizing a free dataset. Our results are an end-to-end Image Recognition application with a webcam or file upload to read in images, input those to a trained CNN model running in a open source Docker container to be analyzed and output results to be displayed on a web application and recognition accuracy to be determined afterwards.

INTRODUCTION

A good image classification model should recognize objects of different categories correctly. The results of such a model should not vary due to **viewpoint variation, illumination conditions, object distortion or occlusion** and it should classify these images correctly like humans. Our goal was to create a model that could be used to output an emotion from an image input.

To train a model, a decent scale of image dataset needs to be used for fine-grained image classification. The Japanese Female Facial Expression (JAFFE) Database dataset contains 213 images of 7 facial expressions. The goal of this project is to create a model which best utilizes these 213 images to create a program that can categorize any inputted image into one of the 7 facial expressions. Baidu's PaddlePaddle AI Framework was used to implement the machine learning features, while Node.js and HTML5 were used to implement the web service.

SOFTWARE ARCHITECT



- train.py creates the parameters for the neural network, which are stored in params_pass.tar
- server.js serves the webpage. When the client accesses the web page, the server waits for an image to be uploaded. When the image (the input) is uploaded, it is passed to infer.py, which loads a neural network using params_pass.tar. infer.py then generates a result which is received by server.js and passed back to the client, where it is then displayed on the web page.

MODEL TRAINING

Since our model was only trained with 213 images, the model could train fairly quickly and we did not have to use GPU training. Training was done through train.py, which parsed the images in the dataset and loaded them to be used for training the CNN with PaddlePaddle. A larger dataset would have required more time to train, which was not available in the single week we had. Ideally, a larger dataset would be used at the cost of a longer training time.

RESULTS

Given the size of the dataset (213 images), excellent results were not expected. With an accuracy level of ~20%, it is clear that the model does do a certain extent of inference, but obviously is not very accurate. The application would be expected to function much better with larger datasets. This model created with the smaller dataset functions satisfactorily given the small amount of training data.

APPLICATION

This concept could be used to provide automated user feedback (with permission of course), and improve any application's user interactivity or accessibility. The ability for a program to recognize a user's facial emotion could be harnessed in many ways.

CONCLUSIONS

- A web service is effective to let users interact with machine learning applications like this one.
- A larger data set is crucial to train a more accurate CNN model; a small dataset of only 213 images is only sufficient for the ~20% accuracy we obtained, especially for image recognition.

REFERENCES

- PaddlePaddle - Deep Learning 101
- PaddlePaddle - Python API Documentation
- Docker command reference

ACKNOWLEDGEMENTS

Baidu Inc.