Real-Time Fashion Classification Using CNN, ONNX, and OpenCV

Manuel Mateo Delgado-Gambino López
Advance Automatic Learning
UIE
Email: manuel.delgado_gambino.01@uie.edu

***Abstract*—This document presents a real-time fashion classification project that uses a Convolutional Neural Network (CNN) to classify images from the Fashion-MNIST dataset. The system is built using TensorFlow for model training, tf2onnx for converting the trained model into the ONNX format, and OpenCV for real-time video capture and prediction.**

I. Introduction

The rapid evolution of computer vision techniques has facilitated the development of real-time applications. In this project, a CNN is utilized to automatically classify images of clothing items. After training with TensorFlow, the model is converted to ONNX, allowing deployment in diverse environments. OpenCV is then used for capturing video and applying real-time inferences.

II. Data Loading and Preprocessing

Data are loaded from a CSV file containing the Fashion-MNIST test set. In the preprocessing step, images are reshaped to a 28x28 pixel size with an additional singleton dimension, normalized, and then split into training and testing sets. The preprocessing pipeline is implemented in a dedicated module (`data_loader.py`).

III. Model Architecture and Training

The CNN model is designed using TensorFlow's Keras API. The architecture is comprised of the following layers:

- A convolutional layer with 32 filters, followed by a max-pooling layer.
- A convolutional layer with 64 filters, followed by a max-pooling layer.
- A flattening layer that converts the 2D feature maps into a 1D feature vector.
- A dense layer with 128 neurons and ReLU activation.
- A dropout layer for regularization.
- A final dense layer with 10 neurons (for 10 classes) and softmax activation.

The model is compiled using the adam optimizer and the categorical cross-entropy loss function. It is subsequently trained over 10 epochs.

IV. Model Conversion and Real-Time Inference

After training, the model is saved in HDF5 format and converted to the ONNX format using `tf2onnx` (see `onnx_export.py` for details). For real-time inference, OpenCV captures live video frames. Each frame is preprocessed (converted to grayscale, resized, and normalized) to suit the CNN's input requirements. The preprocessed frame is then passed to the ONNX runtime, and the resulting prediction (class label) is rendered on the video stream itself (see `opencv.py`).

V. Results and Discussion

The integration of training, model conversion, and real-time prediction demonstrated high feasibility for streaming applications. The use of ONNX runtime guarantees platform independence and flexibility, while OpenCV provides robust video handling. The project shows promising performance in classifying fashion items under real-time constraints.

VI. Conclusion

This work demonstrates an end-to-end pipeline for real-time fashion classification. Future work could involve optimizing the CNN model, exploring alternative preprocessing strategies, and implementing advanced post-processing techniques to further improve real-time performance.

Acknowledgments

The authors would like to acknowledge the open source communities behind TensorFlow, ONNX, and OpenCV for their invaluable contributions.