German International University of Applied Sciences
Informatics and Computer Science
Dr. Caroline Sabty
Eng. Aya Abdalla

Advanced Machine Learning, Spring 2024
Assignment 2
Due date is May 9, 2024 at 11:59 PM

Video classification is a challenging task that aims to automatically categorize videos based on the content they depict. Videos information rely on mainly two aspects, their individual frames (spatial details) and the sequence of those frames (temporal flow).

Classifying videos using only CNNs, which excel at analyzing single images, wouldn't capture the essential temporal aspect. Here's where RNNs come in.

By combining CNNs for feature extraction from each frame and feeding those features into RNNs, we can create a model that understands both what's happening in each frame (objects, actions) and how those elements unfold over time (e.g., someone walking vs. running). This combined CNN-RNN approach provides a more comprehensive understanding of video content, leading to more accurate video classification.

This project investigates video classification using a combined Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) architecture. The goal is to identify actions within videos by capturing spatial features from frames with a CNN and modeling temporal relationships between these features using an RNN.

1 Dataset

We will utilize the UCF101 action recognition dataset from TensorFlow Datasets

`https://www.tensorflow.org/hub/tutorials/action_recognition_with_tf_hub`

UCF101 offers a diverse collection of videos showcasing various human activities.

a) Download the dataset or fetch using UCF101 url, you can use the following link to help you fetch dataset `https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/hub/tutorials/action_recognition_with_tf_hub.ipynb`

b) Create a dataframe with two columns, one for videos paths and one for the labels

2 Data Preprocessing

a) Define helper method to extract frames from videos, the method should take video path as input and return an array of video frames as output

b) Apply preprocessing steps on each frame ex: (resize to a fixed size to normalize frame dimensions)

c) Split the dataset into training, validation, and testing sets (optional, can be done during data loading).

3 Feature Extraction with CNN

Design a CNN architecture (e.g., VGG16, ResNet50, or a custom model) to extract spatial features from each video frame. Consider pre-trained models for faster training like InceptionV3

a) Implement a feature extractor that could extract features form Image

b) Apply this feature extractor for all frames per video then apply for all videos (this could take several minutes you can include at least 300 video)

4 Sequence Modeling with RNN

Implement an RNN sequence model (e.g., LSTM or GRU) to capture temporal relationships between the extracted frame features. RNNs excel at processing sequential data like video frames. The RNN will process the sequence of features, learning temporal dependencies between frames to provide context for action classification.

5 Model Training

a) Combine Features and RNN: Concatenate or feed the extracted features from the CNN into the first layer of the RNN.

b) Compilation: Compile the model using an appropriate optimizer (e.g., Adam) and loss function (e.g., categorical cross-entropy , sparse categorical crossentropy) suitable for multi-class classification.

c) Training Process: Train the model on the training set, monitoring validation loss to prevent overfitting (e.g., early stopping technique).

1. During training, the model iterates through video sequences, feeding extracted features from each frame into the RNN.
2. The RNN processes the sequence, capturing temporal dependencies and ultimately predicting the action class for the entire video.
3. The loss function calculates the difference between the predicted and actual class labels.
4. The optimizer updates the model's weights based on the calculated loss, gradually improving its classification accuracy.

6 Evaluation

a) Evaluate the model's performance on the testing set using metrics like accuracy, precision, recall, and F1-score

b) Compute a confusion matrix to visualize the model's classification performance across different action classes

7 Deliverables

:

a) Your code needs to be submitted as google colab notebook link to the following google form : `https://forms.gle/m9zcR8kBvGbBRfTs6`, **please make sure to open share access of your notebook**

b) Make sure to comment on every step while coding.

c) Please split the code in cells (Don’t write all your code in one cell)

PLAGIARISM IS NOT TOLERATED AND COPIED WORK WILL BE AWARDED 0 POINTS FOR ALL TEAM MEMBERS !