REAL-TIME HUMAN ACTIVITY CLASSIFICATION USING DEEP LEARNING TECHNIQUES

[DATA695 Research capstone project in data science and analytics]

Hao Su, An Yan

Problem Description

The proliferation of video data in various sectors, including security, healthcare, and entertainment, has increased the demand for automated video analysis systems. One such analysis is human activity recognition (HAR), which refers to the process of classifying a sequence of observed actions performed by a human subject in videos. While there has been significant progress in this field, real-time and accurate recognition of complex human activities from video data remains a challenging problem due to the variations in perspective, illumination, speed, and scale of human actions in different video sequences. The objective of this capstone project is to design and implement a model that can classify human activities in real-time, leveraging deep learning techniques and the UCF101 dataset. By classifying these activities in real-time, we aim to contribute to the fields that can greatly benefit from this technology, such as surveillance systems, human-computer interaction, and sports analysis.

Data Source

The dataset we used in this project is the UCF101 dataset (https://www.crcv.ucf.edu/data/UCF101.php) from the University of Central Florida Center for Research in Computer Vision (CRCV). The UCF101 dataset consists of 13,320 videos labeled across 101 action categories, grouped into five types: Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments, and Sports. This dataset provides a diverse collection of actions, making it ideal for training and evaluating our model's performance. (UCF Center for Research in Computer Vision)

Method and Tools

The method we used for this project is a combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks and a 3D Convolutional Neural Network (Conv3D). The CNN layers were utilized to extract spatial features from each frame image, capturing object appearance and structural information. On the other hand, LSTM layers were employed to model temporal relationships, learning dynamic patterns and contextual information across different frames. By integrating these two network structures, the CNN+LSTM model effectively captures both spatial and temporal features within videos, leading to more accurate classification and prediction outcomes. For the second approach, the 3D convolutional neural network (Conv3D), we tried to add time as the third dimension. Python was the main programming language used due to its extensive support for scientific computing and machine learning. The deep learning package, TensorFlow, was utilized for defining and training neural networks. OpenCV was used for real-time video reading, processing, and activity prediction. Also, AWS was used as the cloud server for our project.

Steps and Analysis

- **Data preparation**
  When employed for real-time human activity classification using deep learning techniques, owing to constraints such as personal device performance, we did a sequence of preprocessing steps during the feature extraction process to ensure that the video data is suitably prepared for subsequent deep learning model handling. Initially, the video's resolution is adjusted to 80x60 dimensions. This resizing not only aids in reducing computational complexity but also guarantees uniform sizing across all input images, facilitating a consistent input data flow. Subsequently, the video's frame rate is reduced to 10 frames per second. After this adjustment, the computational workload could be reduced, considering that processing high frame rate video may require more computational resources. Moreover, lower frame rates may still capture crucial information in scenarios where human activity changes are not frequent. Lastly, based on the decreased frame rate and overall frame count, the code employs a predetermined interval to select frames for extraction.

- **Video Loading**
  We iterated through different folders of various categories of datasets for video loading. For each video file within each folder, it constructs the path of the video and subsequently employs the function in previous step to preprocess the frames of the video. These preprocessed frames are then appended to a list, which will eventually form our input data. Simultaneously, the corresponding label index (representing the activity category) is added to a separate list of labels, facilitating subsequent classification tasks.

- **Model Building**
  In our study, we only chose the first 30 categories from the UCF101 dataset. While the UCF101 dataset covers numerous activity categories, we narrowed our attention to these specific 30 categories to delve deeper into the recognition challenges associated with these activities.

1. *Simple CNN+LSTM Model*
   First, we built a sequential model for video classification using a combination of convolutional and LSTM layers. The model architecture comprises multiple time-distributed convolutional layers followed by LSTM and dense layers. Convolutional layers process frames within each video sequence, extracting features, which are then fed into the LSTM layer to capture temporal patterns. The model outputs a classification result through a dense layer.
   Next, we trained a deep learning model for video classification. The model is configured using categorical cross-entropy as the loss function and the Adam optimizer with a

learning rate of e^{-3}. During training, an early stopping mechanism is employed to monitor validation loss and halt training if no improvement is observed for a defined number of epochs. This helps prevent overfitting by restoring the best model weights. The model is trained using specified batch size and epoch settings, ensuring effective learning while avoiding overfitting.

Finally, we plotted a graph to check training and testing accuracy over the period.

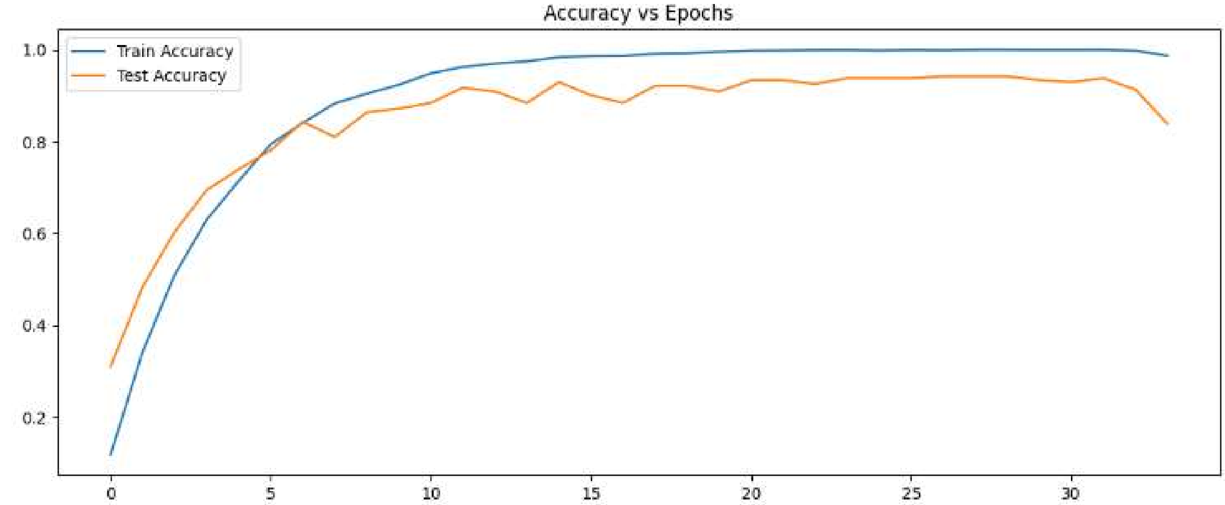


Figure 1. Accuracy plot for CNN+LSTM model

The training set accuracy gradually increased and later leveled off, while the test set accuracy showed an overall upward trend but sharply declined after epoch 30. The gradual increase in training set accuracy followed by a plateau suggests that the model was effectively learning and fitting to the training data. However, the overall upward trend in the test set accuracy, followed by a rapid decline after epoch 30, indicates a potential issue of overfitting. The model's ability to generalize to unseen data diminishes after a certain point, leading to a drop in test accuracy. Additionally, we calculated the model's accuracy to be 94%, which can serve as a benchmark for comparing with subsequent models.

2. *Conv3D Model*

The second model we used for activity classification is a Conv3D model. The Conv3D model is a three-dimensional convolutional neural network (CNN) designed specifically for processing data with a temporal dimension, such as videos. The "3D" in Conv3D stands for three-dimensional, indicating that it operates on three dimensions: width, height, and time. Unlike traditional image recognition models, the Conv3D model considers the temporal dimension, allowing it to capture both spatial and temporal features present in video data. This enables better understanding and classification of dynamic content. In video classification tasks, the Conv3D model efficiently captures the correlations between video frames, leading to improved classification performance.

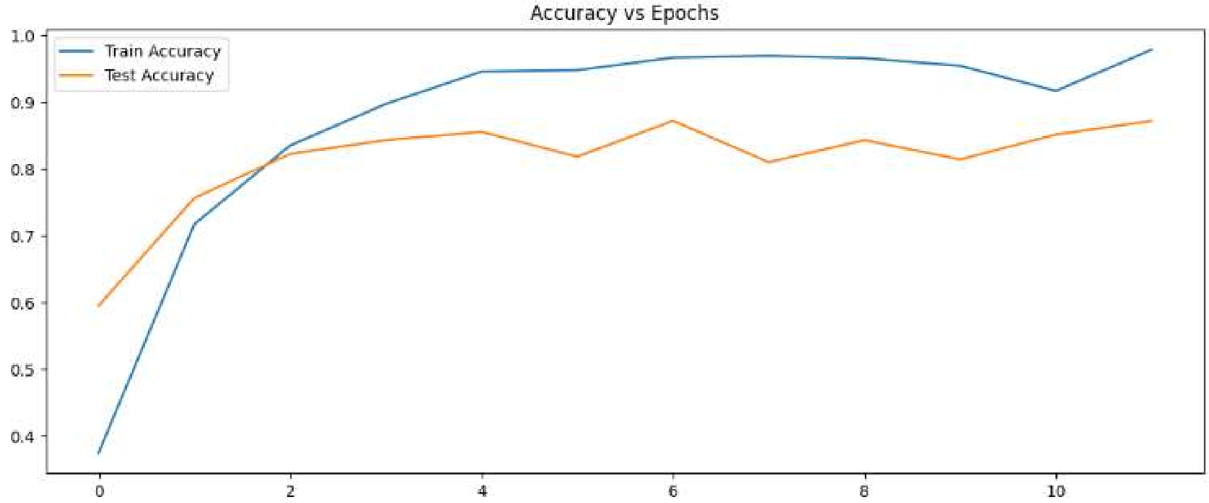


Figure 2. Accuracy plot for Conv3D model

It can be observed that prior to epoch 2, the accuracy of the test set is higher than that of the training set, but thereafter, the training set consistently exhibits higher accuracy. Despite fluctuations, both dataset's accuracies show an overall increasing trend. This suggests the likelihood of the model experiencing overfitting. The increase in training accuracy signifies the model's efforts to fit the training data, while the decrease in test accuracy indicates poorer performance on unseen data. The accuracy of this model stands at 87%, which is lower than the previous model.

- **Video Classification and Predictions**

  To observe the model's performance in classifying videos, we employed two prediction methods for comparative analysis. The first one used the first frame of a randomly selected video for prediction and visualize the frame along with the prediction result.

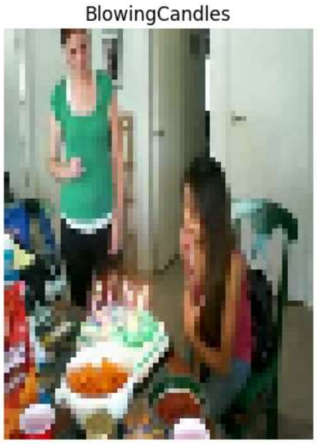


Figure 3. Second classification output

The successful prediction of the "blowing candles" class indicates that the model was able to accurately classify the given video frame as an action of blowing candles. This suggests that the model has learned to recognize relevant visual features and patterns associated with the "blowing candles" action. The correct prediction aligns with the model's ability to generalize and make accurate predictions on unseen data, highlighting its potential effectiveness in identifying specific activities or actions in videos.

The second method used a randomly selected complete video for prediction and subsequently displaying each frame alongside final prediction class label. The output contains ten images since each video was compressed as 10-frame.

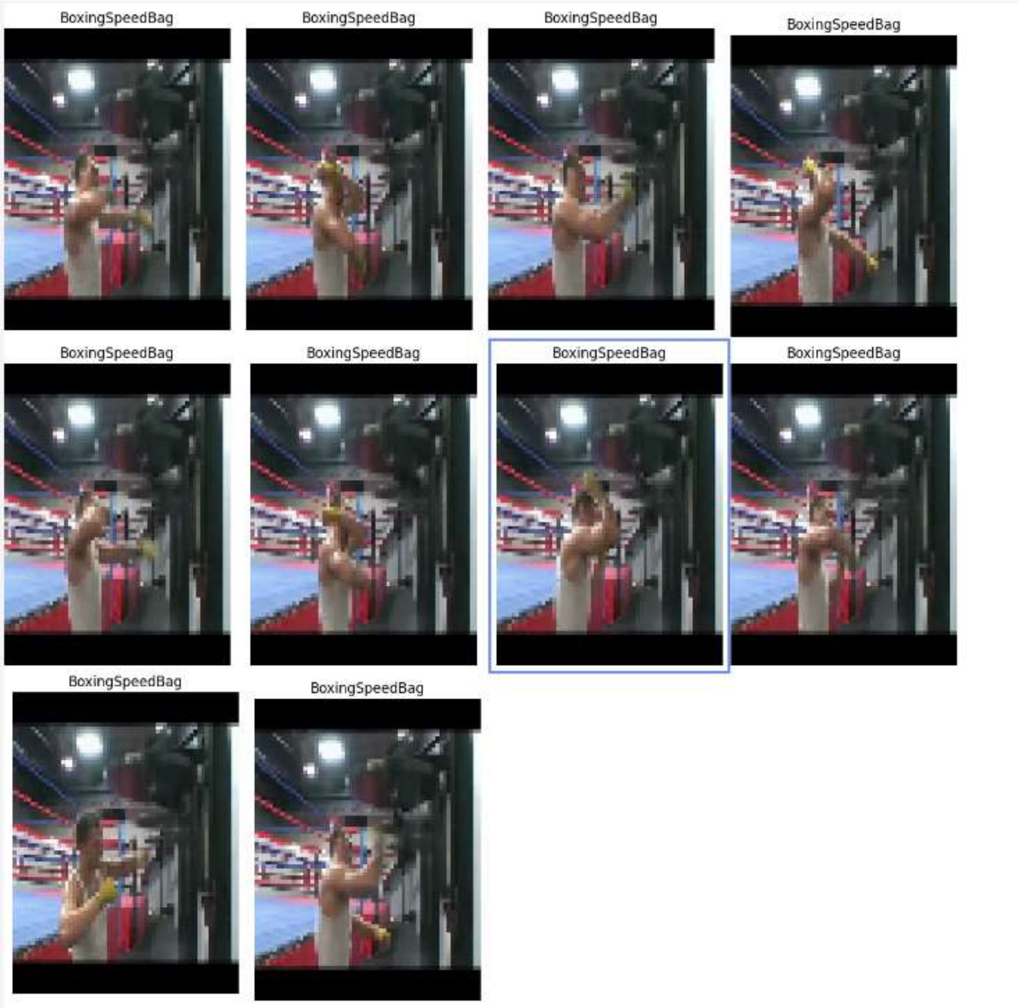


Figure 4. Second classification output

The output of the second method effectively demonstrates the performance and capability of the trained model. By visualizing prediction labels at the top of video frames, it provides an intuitive way to measure the accuracy of models in real-time scenarios. In this instance, the randomly selected video was classified as "BoxingSpeedBag", suggesting that the content of the video may involve scenes of boxing training using speed bags. It demonstrated that the model has a high level of accuracy in classifying the video as that class and this level of accuracy can be attributed to the model's ability to capture fine-grained details and temporal patterns in the video data.

- **Potential Model Improvements**
  By examining the performance of the model, it becomes evident that there exists significant potential for improvement and enhancement. Therefore, distinct improvement strategies can be applied to different models, aiming to further enhance their performance in video classification tasks. By delving into the models' limitations, optimization potential, and applicability to various model structures, we can formulate more accurate and robust classifiers that adapt to diverse video scenes and intricate data features.

  1. *Improvements due to Limited Data Diversity:*

     <u>Enhancing Data Diversity:</u> Due to our decision to focus on only 30 categories from the UCF101 dataset, it's important to acknowledge that this choice could potentially introduce limitations to our model's performance. In comparison to models trained on the entire set of categories, our model is unable to make predictions when faced with activities not covered by our chosen subset.
     To address the limitation, instead of limiting ourselves to a subset of 30 categories, we can endeavor to include as many categories as our computing capability allow. This entails selecting a diverse range of categories that encompass various activities, settings, and scenarios.

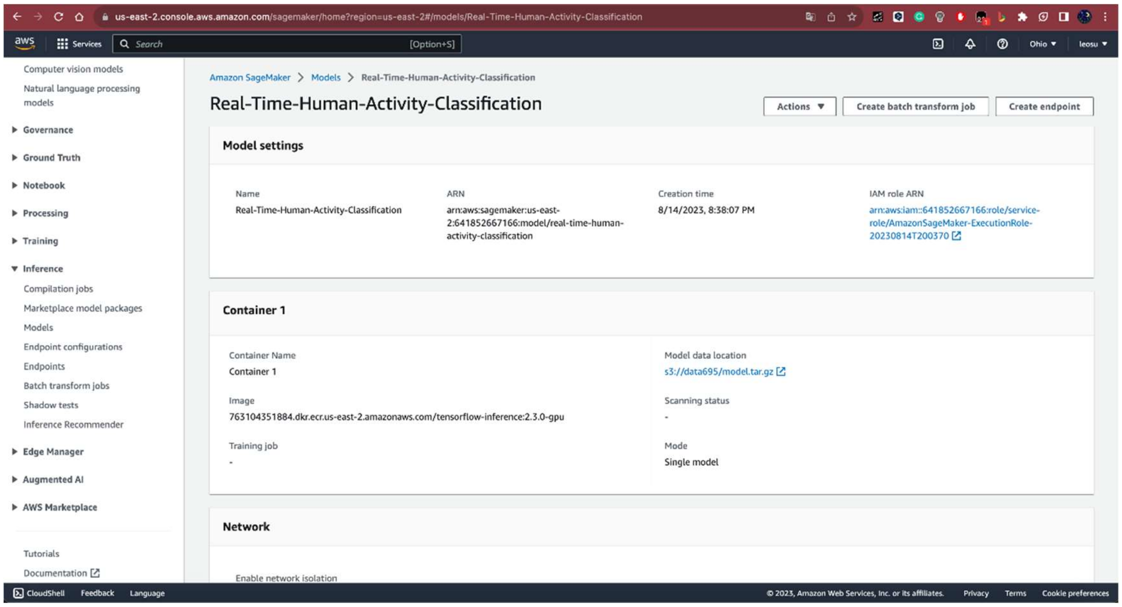
  2. *Improvements for the CNN + LSTM Model:*

     <u>Data augmentation</u>: It is a technique that enhances model performance by applying diverse transformations and expansions to the training data. Specifically, by applying various transformations to the training data, data diversity is increased, enabling the model to better learn different features present in the data. Among the mentioned data augmentation techniques, random cropping, rotation, and flipping help improve the model's robustness and generalization. (Karani)

  3. *Improvements for the Conv3D Model:*

     <u>Regularization and Normalization</u>: Since we can see the trend of overfitting in this model we can Introduce techniques like Dropout, L2 regularization, or batch normalization to reduce overfitting and enhance the model's generalization ability.

- AWS Cloud
  The model has been successfully deployed to AWS, enabling future implementation of online prediction capabilities, and allowing anyone to access and utilize our model for real-time classification predictions through API endpoints. This deployment approach provides our model with high scalability and availability, capable of handling a large volume of prediction requests while maintaining optimal performance. Our model can now be easily integrated into various applications, websites, or services, offering users real-time classification functionality.



Real-Time-Human-Activity-Classification

Model settings

Name: Real-Time-Human-Activity-Classification

ARN: arn:aws:sagemaker:us-east-2:641852667166:model/real-time-human-activity-classification

Creation time: 8/14/2023, 8:38:07 PM

IAM role ARN: arn:aws:iam::641852667166:role/service-role/AmazonSageMaker-ExecutionRole-20230814T200370

Container 1

Container Name: Container 1

Model data location: s3://data695/model.tar.gz

Image: 763104351884.dkr.ecr.us-east-2.amazonaws.com/tensorflow-inference:2.3.0-gpu

Scanning status: -

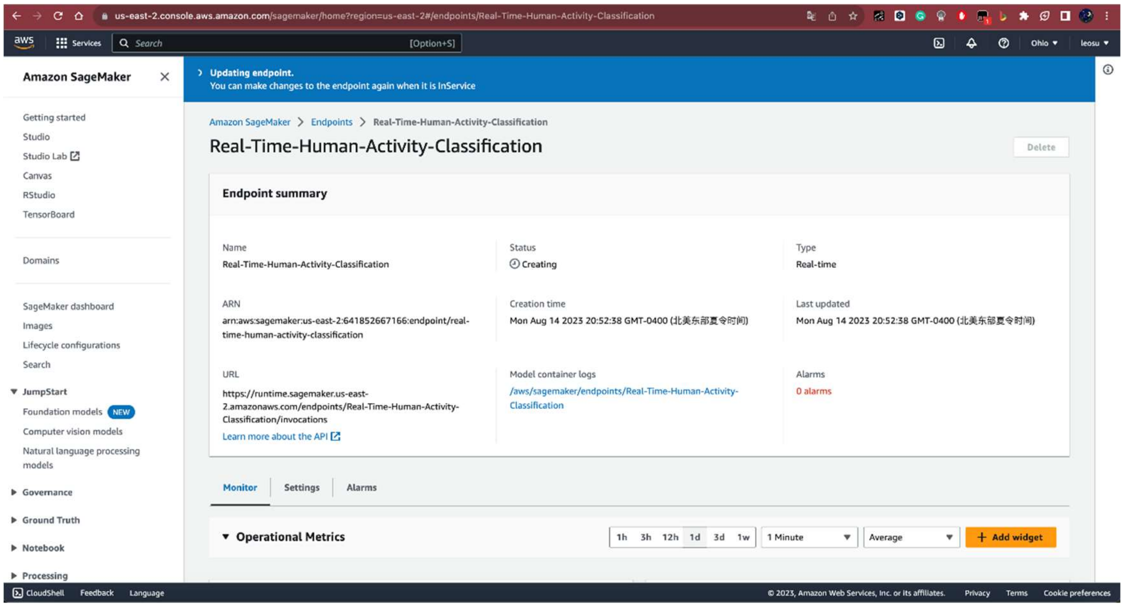
Training job: -

Mode: Single model

Network

Enable network isolation

Figure 5. Model on AWS



Updating endpoint.
You can make changes to the endpoint again when it is InService

Real-Time-Human-Activity-Classification

Endpoint summary

Name: Real-Time-Human-Activity-Classification

Status: Creating

Type: Real-time

ARN: arn:aws:sagemaker:us-east-2:641852667166:endpoint/real-time-human-activity-classification

Creation time: Mon Aug 14 2023 20:52:38 GMT-0400 (北美东部夏令时间)

Last updated: Mon Aug 14 2023 20:52:38 GMT-0400 (北美东部夏令时间)

URL: https://runtime.sagemaker.us-east-2.amazonaws.com/endpoints/Real-Time-Human-Activity-Classification/invocations

Learn more about the API

Model container logs: /aws/sagemaker/endpoints/Real-Time-Human-Activity-Classification

Alarms: 0 alarms

Operational Metrics

Figure 6. Endpoint

References

- UCF Center for Research in Computer Vision. (n.d.). UCF101 - Action Recognition Data Set. UCF CENTER FOR REDEARCH IN COMPUTER VISION. https://www.crcv.ucf.edu/data/UCF101.php

- Karani, Dhruvil. "How Data Augmentation Improves Your CNN Performance? - An Experiment in PYTORCH and Torchvision." Medium, 2 Sept. 2020, https://medium.com/swlh/how-data-augmentation-improves-your-cnn-performance-an-experiment-in-pytorch-and-torchvision-e5fb36d038fb