SAPIENZA
UNIVERSITÀ DI ROMA

Vision and Perception

Final Project - Phase B

Ekin M. Senler- Matricola: 1801499
Cecilia A. Peredo - Matricola: 1822225
Huriye Kayis - Matricola: 1802952

1 - Introduction to Classification

The first part of this project was to train a Mask R-CNN network that would learn how to recognize and detect certain objects. The detection of these objects now helps us classify videos that could contain these objects into their respective category (Ballet, Cumbia, Cheerleading, Tango, Belly Dancing, Breakdance, and Baton Twirling). In order to classify correctly the category of the activity, several actions were evaluated and performed. One of the most important of those was the pre-processing of our dataset to provide the network enough quality information to learn; this was followed by the extraction of the feature vector that describes the trend of the object detection per activity, and lastly this feature vector was inputted into an LSTM network to train and learn to infer the activity. The result of our methodology is described subsequently.

1.1 - Preparation of Dataset

To classify videos into their category through a network, it is important to feed a network with data that contain quality information and sufficient examples to learn. Our original dataset contained videos of every category. These were organized into two separate folders containing the same categories in their respective folders. The folder ‘test’ contained the first 10 videos of each category and the folder ‘train’ contained the remaining videos. This would later help us train and evaluate our network performance.

The quantity of videos in each category differed ranging from about 30 to 80 videos. After analyzing this data, it was determined that the quantity of data would not be sufficient to train a model. Therefore, we proceeded to divide each video into smaller portions, each containing different frames from the original video. Each video generated beame 10 seconds long each.

From each 10 second video, images were extracted that would then be analyzed by our previous trained network to provide the detected objects. An image was extracted for every 3 frames per second, resulting in 30 images (from 30 equally spaced frames) per video. For each image generated, the trained network analyzed and inferred the objects in the scene. The outputted information regarding each object was then used to create the feature vector that describes the pattern of each activity.

2 - Feature Extraction

Extracting features is the most challenging part of the second phase of the project. Since we are allowed to use any possible feature as a input to our lstm network we have been through wide research about past studies. After collecting enough information and reasoning with our own ideas we have found possible features as, keeping track of bounding boxes, evaluating the scores of each mask, counting the objects and output of max-pool layer from mask-rcnn.

Evaluating “bounding-boxes” can be characterized according to many parameters. Very basically it gives information about the four points of the coordinates which allows us to calculate center of the mask and also the area. Here the challenging part is to deciding the vector when there are multiple instances of one object. We have decided to use average, this way we were able to see how much the object moves through time. Another parameter we can calculate is the area. Area is a very useful feature since it gives information about if object can be highly close to camera or not. For example one object like “baton”, can not be very close to the camera, by calculating the area we make sure if system detects baton and if its bounding box is covering the most of the image, we can assume it is either not baton or baton twirling.

Keeping the values of coordinates of the bounding box is extremely important since it gives exact information about where the object in the image is detected mostly. For example a ballet shoe is at the bottom of the image while sneakers are supposed to be everywhere.

One other useful feature that maskrcnn provides is the “scores” of each mask detected. Having the information of scores teaches system if one object is detected with high scores and it is detected true during the evaluation, later system will remember this information.

During the time we developed the project we have tried each approach separately on a small subset of our actual dataset. Due to lack of computational power our computers were not able to produce enough data. As a result we decided to use a one-hot representation of objects detected as a feature vector. This was the most feasible way in our case. Resulting feature vector is [sequence_lenght, num_objects*num_objects]. Since we wanted to run on a small dataset 40 for sequence length seem to be a reasonable number. For the total number of objects including background, COCO Objects and our chosen objects for dance dataset is 32. Resulting vector is [40, 1024]. If we had more computational power we would be able to give more information with the outputs of maskrcnn as stated above.

2 - Lstm Architecture

Humans don’t start their thinking from scratch every second. They build up their knowledge on the previous experiences, and don’t throw everything away than start thinking from scratch again. Traditional neural networks can’t achieve that. It seems like a major shortcoming. Recurrent neural networks address this issue. They are networks with loop in them, allowing information to persist. And LSTM (Long short-term memory) is a special kind of RNN which almost all the existing results based on.
Knowing the success of LSTM power on learning sequential data, we try to construct LSTM model on top of our classification result from mask RCNN. With some research, we found out that complicated LSTM model doesn’t perform well on the video classification result. We keep

our model simple with 2048 LSTM unit and one dense layer with dropout value of 0.5 than connected to fully connected layer to yield the classification result.

3 - Future Improvements

To improve the accuracy of our classification, some changes can be implemented with the use of more computational power, memory, and time. One main change is the expansion of the dataset. Even though our approach accounted for such an issue, a greater diversity of videos would help the network. For this, videos would have to be downloaded from the web and classified manually to its correct category followed by the manipulation of the data for the extraction of features. The feature vector can also be improved by adding more significant features. For example, our initial vector contained more data but in order to save resources due to the limited computational power we had to reduce our vector to only the most essential and descriptive features for the objects. Our initial vector contained more elements, these were the count of each object encountered, the average confidence detection score of each object, and the average area of the masks (since our current area feature only contains the last object detected when more than one object of the same type is found). These features that were initially part of our feature vector would add more information help the LSTM train effectively.

A major addition that could be added to our feature vector is the addition of a descriptive layer of the previously trained Mask R-CNN. While we attempted this step and learned how to extract this feature from the rebuilt model, a critical part is the extraction of the correct layer that will identify the proper features. After many hours dedicated to the analysis of this, it was determined that the only way to find the optimal layer was to test each of the layer options that we had agreed upon. This unfortunately would take too long as our computational power is limited and does not yet include the training phase which could also be significant depending on the size of the extracted feature vector.

To further improve the feature vector, both vectors mentioned above with the full extracted feature from the detection and the trained network can be joined into one more complete vector. Of course, with added information the accuracy could improve but would take significant time to train. With added resources, any or all these options can be implemented.