BOĞAZİÇİ UNIVERSITY

CMPE 492

REPORT

WEEK 4

İNCI MELIHA BAYTAŞ

BARAN DENIZ KORKMAZ

DOĞUKAN KALKAN

FALL 2020

1 Weekly Summary

1.1 Work Done

1.1.1 Extraction of Video Frames

The extraction of video frames from the downloaded videos have been upgraded in order to enable multiprocessing. This way, the extraction time will diminish in a considerable amount of time.

1.1.2 Transfer Learning: Execution

The extracted frames have been used to form sample subsets. Each collaborator has formed a distinct subset. Deniz has formed a subset using FaceForensics++, whereas Doğukan has used DeepFakeDetection. The results of two distinct executions can be found in the following link:

`https://github.com/barandenizkorkmaz/DeepFake-Detection/tree/master/Deliverables/Week%204/Fake%20Image%20Classification%20using%20Transfer%20Learning`

Results:

1. Execution 1:
    - Dataset Description: Low-Quality(LQ) Extracted Frames from 15 videos in average, which are manipulated by Deepfakes, Face2Face, FaceSwap, FaceShifter, Neural Textures. The dataset is divided into two subsets of Training and Validation sets with the ratio of 80/20.
    - Comments: The dataset contains lots of similar frames since a video sequence is manipulated by 5 different techniques. We assembled frames from 15 videos that are generated by different manipulation techniques.
    - Results: The dataset contains lots of similar images, which makes it easier to classify.
        - Training Acc. = 100.0%
        - Validation Acc. = 100.0%
2. Execution 2:
    - Dataset Description: Low-Quality(LQ) Extracted Frames from 156 videos from DeepFakeDetection dataset.. The dataset is divided into two subsets of Training and Validation sets with the ratio of 80/20.
    - Comments: An original sequence might also be used in the derivation of distinct fake sequences again. However, this subset presents much variety than the first one.

- Results: We observe that the VGG-19 network provides some promising initial results.
  - Training Acc. = 100.0%
  - Validation Acc. = 100.0%

1.1.3 LSTM

Building upon our current CNN, we are trying to add an LSTM layer to our design. As we discussed in our meeting, each video contains roughly 400 frames, and for each video we are supposed to pick 50 frames. Each frame is picked after skipping 8 frames since we do not want to pick 50 consecutive frames which would fill the dataset with reduntant frames. For each video, which consists of 50 frames, we want our model to produce feature maps of those frames. And preferably, the dimension is 512. Once we have these feature maps, we feed them to the LSTM layer of our design. Since each video consists of 50 frames, the number of units in our LSTM layer should be 50. And, it is important to note that, as our supervisor warned us, the dimensionality of the hidden layers must not be less than the dimensonality of the feature maps of the frames, that is 512. One other important issue is that we do not want output from each unit, it is adequate to obtain the output of the last unit. The dimensionality of this output will be 512. And lastly, we add a "Dense(1)" layer and a "Sigmoid" activation function to our model.

1.2 Learning Outcomes

1.2.1 Extraction of Video Frames

Although, it is still time-consuming enough for our CPUs, we think that the script still can bu run in long hours to complete for an entire dataset.

1.2.2 Transfer Learning: Execution

The FaceForensics++ dataset consists of manipulated sequences originated from the same original sequence via 5 distinct face manipulation techniques. Therefore, forming a large subset using FaceForensics++ requires lots of extracted frames from different sequences, which requires a considerable amount of time. Therefore, the variety of frames in DeepFakeDetection has been utilized in order to carry out the toy execution of this week. In the following week, we are going to try to increase the size of datasets in a way that the training will remain possible in our local CPUs, before we start to train in GPU environment.

1.2.3 LSTM

From theoretic point of view, we have learned how LSTM works but we need to extend our knowledge about it to a point where we know exactly how each unit works. When it comes to practice, TensorFlow provides necessary functions to utilize LSTM layers. However, the convention is not the same as

the literature. What we call **Unit** in TensorFlow corresponds to the number of hidden layers in the theory.

2 Challenges

2.1 LSTM

The tensorflow interface is difficult to adapt at first stages. As explained above, there is a conflict between the interface and what we learned in theory. And also, figuring out parameters that are passed to the LSTM-related functions was quite a challenge.

The other problem was the most important and crucial one for our implementation. For each video, we have features of 50 frames. For now, we are still not sure how we should keep these features and feed them into our LSTM layer.

3 What's Next?: Upcoming Week

- LSTM: We need to finish the implementation of LSTM, add it to our current model and test our new model with our data set.