CAPSTONE Project - Machine Learning Engineer Nanodegree

Diego Amicabile - August 2020

AUTOMATIC SPLITTING OF VIDEO GAMES RECORDINGS

Domain background

During the last few years it has become more and more common to stream on platforms such as Youtube and Twitch while playing video games, or to upload recorded sessions. The volume of videos produced is overwhelming. In many of the videos games being streamed there are different types of scenes. Both for content producers and consumers it would be useful to be able to automatically split videos, to find out in what time intervals different types of scenes run. For instance, having as an input the video recording of a Minecraft speedrun, we could be able to produce the time intervals when the game is taking place in the Overworld surface, in caves, in the Nether and the End respectively - the four main settings of this game.

Problem statement

For each game it would be necessary to train a separate model, to be able to recognize the various settings. For this purpose I will use videos I recorded while playing the game "Mount and Blade - Warband" (2010), which also has different types of scenes. We would be interested in identifying in a video from this game the parts where the player skills can be effectively shown, that is, mainly Battles and Tournaments. An uncut walkthrough of this game contains a lot of possibly boring parts, such as walking on the strategic map and shuffling around troops and inventory. A similar separation in scenes could also be applied for instance to games from the Total War series, whose walkthroughs contain both battles and less eventful parts.

If we have a video containing battles and other interesting scenes (such as an attack on a bandit hideout) we would like to produce the periods when the scenes are taking place during the video.

4:38-6:00 Battle
16:17-17:31 Battle
19:28-21:09 Battle

24:52-25:43 Battle
33:51-35:42 Battle
37:59-41:39 Hideout
42:46-44:14 Battle
56:40-58:09 Battle

In a battle

Tournament

Hideout

Other Scenes

Datasets and Inputs

Youtube contains a huge amount of walkthroughs for any game, including the one under analysis. For the purpose of this project, I will use the videos from playlists (1, 2) of mine that I have uploaded into a S3 bucket into the folder *wendy-cnn/videos/*, previously retrieved from youtube using the **youtube-dl** library. Some of these videos also have a companion text file (same bucket, folder *wendy-warband/metadata/*, extension *.prd*) containing the time intervals and categories of scenes.
I have extracted single images from videos at specific time offsets (every two seconds) using the **opencv** library**,** and labelled the resulting images with a scene type using the information in the *.prd* files. In this process, all images have been transformed to have a standard size, color palette and orientation.
The amount of images I have generated in this way, using the current set of videos broken down in scenes, is now 45718. They are contained in this zip file on S3 (3.4 GB).

These are the categories I would like to identify in scenes. The set is not balanced, some weighting or merging of categories may be necessary.

- BATTLE: any battle taking place in an open field or in a village (~13%)
- TOURNAMENT: Tournaments in arena (~14.8%)
- HIDEOUT: the warband assaults a bandit hideout (~2.5%)
- TRAINING: the hero trains in a field or in a village (~1.7%)
- SIEGE: a town is sieged (~0.4%)
- TRAP: hero falls into a trap and must fight their way out (~0.2%)
- TOWN (escape): escape from the town or castle prison (~0.2%)
- OTHER : everything else (ca 67.6%)

Solution Statement

The full solution should be able to recognize scenes in videos and write down the time intervals of scenes that have been recognized, along with the probability for each category.

To allow for that, I will create a model based on CNNs identifying the category probabilities of static images extracted from videos. There is enough information on most images to identify the scene type.

As the likelihood of an image to belong to a certain scene depends also on the images before and after it, introducing a RNN might be interesting, but as this would make the model too complex and would not allow us to work directly on images.

As images in video games are not a subset of images from the real world, an approach using transfer learning with pre-trained generic models does not seem promising.

When training the model, I will use images from each of the categories of interest to build a stratified train and validation set. An approach that I have found out to work well is to pick one out of five consecutive images in the video (randomly), so that from each scene there will be images available both in the training and validation set.

Benchmark Model

I could not find a benchmark for someone trying to solve a similar problem.
The closest problem I know of is categorizing sports videos, as from this link, which gets very high accuracy and uses a combination of CNN, RNN and transfer learning. However, there are two main differences, compared with the probleTm I am trying to solve:

- I am trying to split videos in parts and categorize those parts, and not trying to categorize full videos
- The images that need to be categorized do not come from the “wild”, but are limited by the way they are generated from games graphical engines

Evaluation Metrics

To verify how well the model categorizes images in videos, as this is a problem of categorization with multiple and unbalanced labels, I will use the log loss as a metric for optimizing the model.

This is because I am interested in a model which returns the probability of categories for each image.

We are also interested in a good average F1 and especially on a good recall when identifying images categorized in a different way as “OTHER”.

Project Design

Assumption: all videos have been uploaded to S3. For some videos, which I will use for training, a companion text file including the time interval of identified scenes and their category is provided.

Image dataset creation: Extraction of single images from videos at specific time offsets and labeling of the extracted images with a scene type. Processing of images to make sure that they are all the same size and format. The dataset will be split into training and validation subsets, ensuring that subsets contains similar samples - with balanced categories and retrieved from similar scenes.

Training Job: a neural network will be trained to identify the probability for each category of an image. Our goal is to reach a good measurement on the identified metrics (log loss, f1, recall)

Deployment of Service: a model will be deployed, in form of a service, that will be able to categorize a set of images, returning a probability for each category.
This model will be also able to accept a video or the URL of a video and the seconds interval at which images will be retrieved - say, like, every two seconds.

Client: a client will call the service passing a video and a second interval as an argument, and then execute a visualization of the result - identifying video segments and the probability of what category segments belong to.