Project Proposal Submission - Gif Content Rating

•Team:

Name	ID	Email
Asaf Ben Shaul	208004127	Asafbenshaul@mail.tau.ac.il
Hanan Wullman	316594183	hananwullman@mail.tau.ac.il
Yotam Zvieli	209497023	yotamzvieli@mail.tau.ac.il

•End product (system from user's perspective):

A web application where users can upload a GIF and receive an automated content rating. This rating categorizes the GIF based on predefined levels of appropriateness, ensuring users understand the suitability of the content for different audiences. The system follows the Giphy content rating guidelines, which can be found here. Additionally, the web application will include an API for auto-rating GIFs, providing easy integration for other applications. This automated process, powered by machine learning, eliminates the need for manual checking of GIFs, accelerating the content moderation process.

•Training and inference schemes:

We will use a CNN + LSTM approach for training and inference. The Convolutional Neural Network (CNN) will extract spatial features from individual frames of the GIFs, while the Long Short-Term Memory (LSTM) network will process the sequence of these frames to capture temporal dependencies. This combination leverages the strengths of both architectures: the CNN for spatial data and the LSTM for sequential data.

During training, we will feed labeled GIFs into this hybrid model, optimizing the network parameters through backpropagation. The training will involve multiple epochs with regular validation checks to monitor accuracy and avoid overfitting.

For inference, the trained CNN + LSTM model will process new GIFs uploaded by users. The CNN will extract features from each frame, and the LSTM will analyze the sequence to generate a content rating. This method ensures both spatial and temporal aspects of the GIFs are considered, providing accurate ratings for real-time applications.

•Datasets

The GIF dataset will be constructed using the Giphy public API. The random API endpoint, which provides a random GIF along with its content rating, will be used for this purpose. Due to the API limit of 100 requests per hour, a scraping mechanism will be developed to collect data incrementally over time. The goal is to accumulate thousands of GIF samples to build a comprehensive database. This database will be used to train and refine the machine learning model,hopefully ensuring accurate and reliable content ratings for the web application.

•Compute and storage requirements and options:

- Typically, like other deep learning projects we will need to use GPU for training our data. The training will take time and with GPU it will go faster.
- For storage we will start with 10GB but maybe we will increase it.
  For good results we will need a lot of GIF samples because GIFs can vary widely in content, style, and length. Also, we will want to classify the GIFs to 4 classes and to ensure a good balance for each we will need a lot of samples.
  Average GIF size is 2MB and we will start with 5,000 GIFs and if we see we need more we will increase it.

•Third party tools and models to be used

1. Giphy Public API: This will be used to collect GIFs along with their content ratings. The random API endpoint will help gather a diverse set of GIFs to build our dataset over time.
2. TensorFlow and Keras:
    - TensorFlow: We will use TensorFlow as the primary framework for building and training our machine learning models.
    - Keras: Within TensorFlow, Keras will be used to define and train the CNN+LSTM model. Keras provides a high-level API that simplifies the creation of complex neural networks.
3. CNN + LSTM Model:
    - CNN (Convolutional Neural Network): Used to extract spatial features from the individual frames of GIFs.
    - LSTM (Long Short-Term Memory): Used to capture temporal dependencies and sequence information from the extracted features.
4. Other Libraries:
    - OpenCV: For processing video frames and extracting frames from GIFs.
    - MoviePy: For handling and manipulating GIFs and video files.
    - Scikit-Learn: For dataset splitting and other preprocessing tasks.
    - Matplotlib: For plotting and visualization training metrics and results.
5. Scraping Mechanism: A custom scraping mechanism will be developed to handle the API rate limit and collect data incrementally, ensuring a substantial dataset is built over time.

•Related work (papers, open source projects etc.)

- https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43793.pdf
- https://arxiv.org/pdf/1609.06782