For this project, a custom dataset consisting of the images extracted from the Tom and Jerry episodes was created since no existing dataset was found. These episodes were collected and downloaded from the Jonni Valentayn channel on Youtube using a downloading Tool named Videoder in MP4 format. Youtube platform was selected for gathering of dataset since these videos could be downloaded with ease and the duration of each episode was clipped to the right amount. Each episode's duration is nearly 3 minutes. The frames were extracted at a ratio of 1:5 in the JPG format using OpenCV from the downloaded videos. For each video, a total of 355 - 359 frames were extracted with the dimensions of 1280 720. These frames were then used to generate the training data for the Mask-RCNN model which was then used to classify and segment Tom & Jerry faces from the input frames.

With reference to the dataset prepared for training the Mask-RCNN model, the frames extracted were classified into two classes, Tom and Jerry. Each frame has an associated class which specifies that the character's face is present in the frame. A frame can have either Tom's face, Jerry's face, or both. The data consists of a JSON file that stores the frame name, character name, and the XY coordinates of the corresponding character face. The XY coordinates of the face were marked using a labeling tool i.e., VIA, that is, VGG Image Annotator. This image annotation tool consists of an HTML file that is used to define regions in an image and the description of that region. VIA-2.0.9 version was used. The regions marked were Tom's face and Jerry's face. Frames with unknown faces were left unmarked.

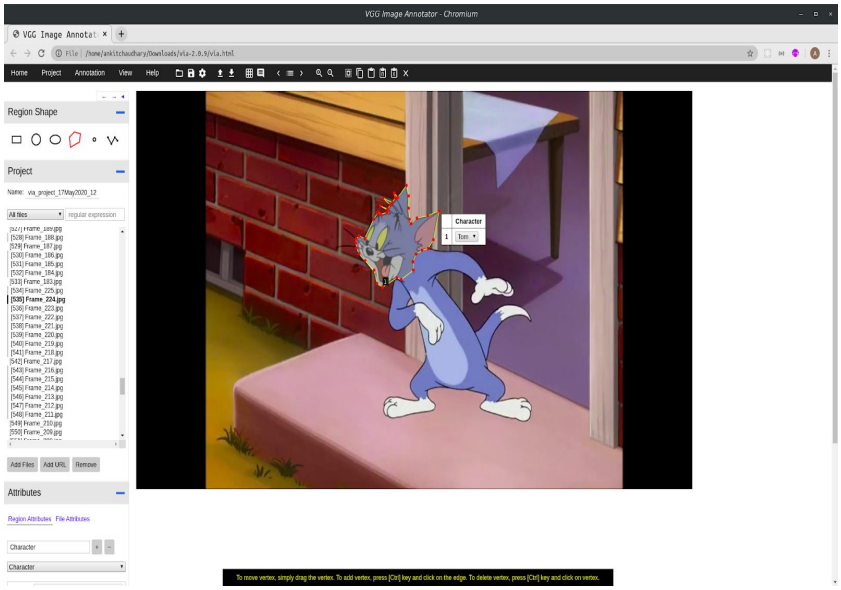
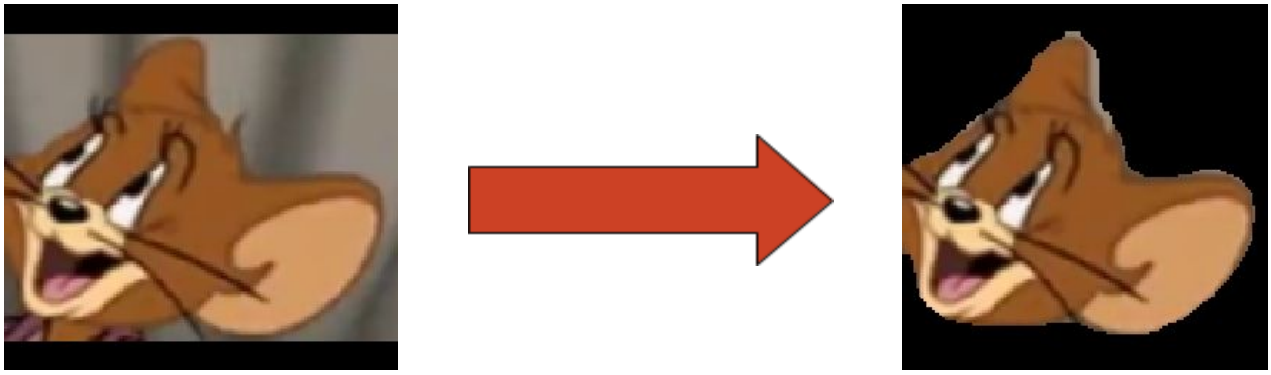


Fig 1. Screenshot of the VGG Image Annotator

For each episode, a JSON file was created. The dataset consisted of 10k images i.e., 28 animations of Tom and Jerry. The frames of each animation along with their JSON data was

used for training the Mask RCNN model. The outcome of this model will be the face detected, confidence score, and boundary box dimensions of the face region. The images were cropped into the size of the boundary box dimensions to extract the faces. These segmented faces [Fig. No. 2] were used to create masks (images with only faces of Tom and Jerry and background as black)[Fig No. 3]. The segmented images and masks were resized into



dimensions of 256*256. The segmented images were stored in two separate directories for both characters named ‘Tom’ and ‘Jerry’. Similarly, mask images were also stored in two different directories named ‘Tom_masks’ and ‘Jerry_masks’. The Segmented images had more extra features (e.g. background features as shown in Fig. No. 2) which are not required while training the Emotion Prediction Model whereas images with only masks and the background as black had more of the right features.

FOLDER STRUCTURE

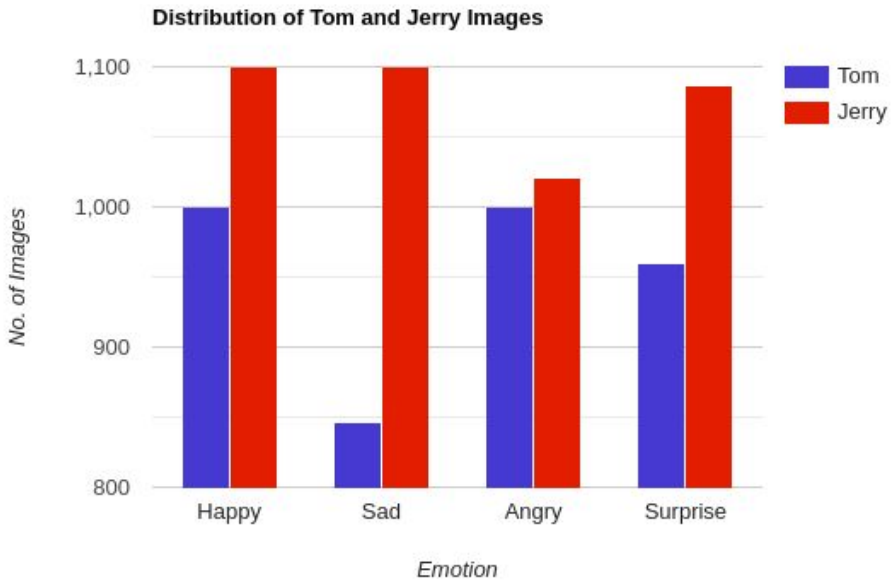
```
extracted_image_folder/
   ├── Tom
   ├── Jerry
   ├── Tom_masks
   └── Jerry_masks
```

Fig4. Folder structure for the extracted Images

Therefore, mask images were selected over segmented images to make the dataset for Emotion Prediction Model.

The dataset generated from Mask RCNN was classified into four emotions. These emotions are Happy, angry, Sad and Surprise. For both characters, around 1000 images depicting each out of the four emotions were manually segregated. While segregating, some of the images

with poor quality were discarded. In total, 8113 images (or masks) of size 256 x 256 were used for training the Emotion Prediction Model.



The segregated images were saved in their respective folders, with the folder structure as given below.

FOLDER STRUCTURE

```
Emotion Dataset/
  ├── Jerry/
      ├── Angry/
          ├── Jerry_Angry_1.png
          ├── Jerry_Angry_2.png
          .
          .
          └── Jerry_Angry_1021.png
      ├── Happy/
          ├── Jerry_Happy_1.png
          ├── Jerry_Happy_2.png
          .
          .
          └── Jerry_Happy_1100.png
      ├── Sad/
          ├── Jerry_Sad_1.png
          ├── Jerry_Sad_2.png
          .
          .
          └── Jerry_Sad_1100.png
      └── Surprised/
          ├── Jerry_Surprised_1.png
          ├── Jerry_Surprised_2.png
          .
          .
          └── Jerry_Surprised_1086.png
  └── Tom/
      ├── Angry/
          ├── Tom_Angry_1.png
          ├── Tom_Angry_2.png
          .
          .
          └── Tom_Angry_1000.png
      ├── Happy/
          ├── Tom_Happy_1.png
          ├── Tom_Happy_2.png
          .
          .
          └── Tom_Happy_1000.png
      ├── Sad/
          ├── Tom_Sad_1.png
          ├── Tom_Sad_2.png
          .
          .
          └── Tom_Sad_846.png
      └── Surprised/
          ├── Tom_Surprised_1.png
          ├── Tom_Surprised_2.png
          .
          .
          └── Tom_Surprised_960.png
```

Fig 5. Folder structure for saving the segregated character masks

The emotions of the both the characters namely Tom and Jerry were labelled in the following order of Sad, Happy, Angry, Surprise respectively, in which the emotions of Jerry are recorded first and then emotions of Tom next in the similar manner. The images with their respective labels were used to train the emotion prediction model.

For training the Emotion prediction model, a csv file was generated consisting of two columns which are **Frame_ID** and **Emotion**.
There are a total of 8113 rows and 2 columns in the CSV file.The Frame_ID column consists of the names of the image files whereas the Emotion column consists of the emotion of our character in the corresponding image.

	Frame_ID	Emotion
0	Jerry_Sad_1	sad_jerry
1	Jerry_Sad_2	sad_jerry
2	Jerry_Sad_3	sad_jerry
3	Jerry_Sad_4	sad_jerry
4	Jerry_Sad_5	sad_jerry
...	...	...
8108	Tom_Surprised_956	surprise_tom
8109	Tom_Surprised_957	surprise_tom
8110	Tom_Surprised_958	surprise_tom
8111	Tom_Surprised_959	surprise_tom
8112	Tom_Surprised_960	surprise_tom

8113 rows × 2 columns

Fig6. The CSV file generated for training the Emotion prediction model

Using the above CSV, the labels were associated with their respective images after which train and test batches were created of batch size = 32 using the batch() method on the tensor slice created for training the emotion prediction model.

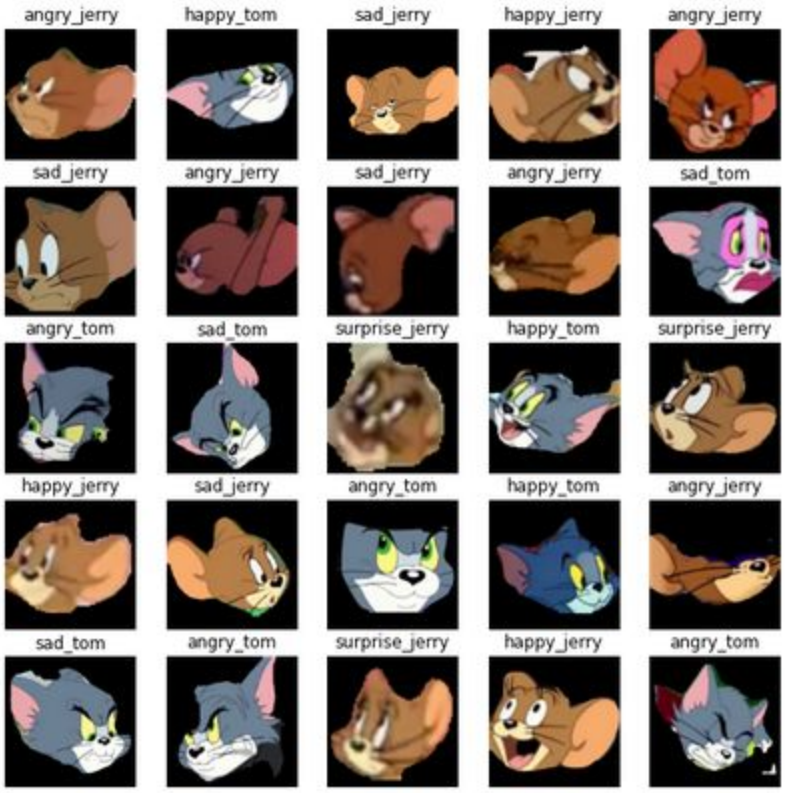


Fig7. Visualizing the images batch with their respective emotion label