SPONGEDETECT *

Adam Hearn, Blake Bryan
Computer Science
George Fox University
Newberg
{ahearn19, bbryan19} @georgefox.edu

ABSTRACT

The following paper displays the utilization of a convolutional neural network using TensorFlow and Keras in order to detect and classify distinct Spongebob character images. Through manual data collection of the first few seasons of Spongebob combined with the proper convolutional layers and a neural network, weights and biases are derived in order to process an image and predict the output for that image. The following layers comprise the neural network in which the images are processed - Input layer, convolutional layer, pooling layer, dense layer. The CNN is used to classify images of four different Spongebob characters - Spongebob, Patrick, Squidward, and Mr Krabs.

***K*eywords** Convolutional Neural Networks · Spongebob · TensorFlow

1 Introduction

Nearly 23 years ago Stephen Hillenburg created the world's greatest cartoon - Spongebob Squarepants. As time has passed, Spongebob has engrained himself in pop culture earning himself all the fame he deserves. However, in recent years after the passing of the original creator, the Spongebob franchise has slowly faded into the depths of its final stretch as a children's cartoon. This disappointing turn in its existence has left the show at the beginning of its end where its content will soon become a mere memory in the minds of the aging Generation Z. As the show dissipates into its demise it is up to the research in this paper to keep the legacy alive. An extremely accurate and sought after method is required to aid those who will soon aimlessly wander the earth not knowing which Spongebob character is which. This research paper leans into utilizing a convolutional neural network that detects specific Spongebob characters in images in an attempt to fix these intense world issues.

To implement the convolutional neural network, the Tensorflow library was used alongside Keras. TensorFlow offers multiple levels of abstraction allowing for ease of use in model building. Keras is a deep learning API that runs on top of TensorFlow that is a simple, flexible, and powerful tool providing those essential abstractions for developing machine learning solutions [1]. See the specific implementation and explanation of our model using TensorFlow and Keras in section 2.2. These powerful high level tools allow for this project to come alive and give a true purpose to what it means to know Spongebob characters. It is through the powers of machine learning that Spongebob's legacy shall live on.

2 Methods

The task is to accurately identify and classify images of certain Spongebob characters given the dataset of character images. By properly building and training a machine learning model, this goal can be achieved. The initial design of the model utilized feature extraction involving the edges of objects within Spongebob frames, as well as dominant colors in the image. Both conventional SVM type models and neural networks using these features did not result in an adequate performance. After multiple trials and errors it was determined that the method of the initial design was lacking and needed to be reworked. The method of manual image feature extraction combined with the SVM classifier was trashed.

Through intense research and sheer willpower a convolutional neural network was then written. The revamped design uses Tensorflow's Keras to demonstrate image classification. With just a single layered CNN, this method saw much more success in it's ability to accurately classify Spongebob character's based off of the given image.

* *Citation*: **Adam Hearn, Blake Bryan. SpongeDetect. Pages.... DOI:69420/8008135.**

Adding a few more layers and taking advantage of KerasTuner to tune the model's architecture further improved the model's accuracy. See Section 2.2.1 for more on the utilization of KerasTuner.

2.1 Data Preparation

To prepare the data the entire, Spongebob seasons two, three, and four were downloaded. Using FFmpeg, a FOSS video conversion utility, frames were captured every 2 seconds from each of the episodes. From there the frames were cropped and resized to 256x256 images. See figure 2 In order to ensure a high quality data corpus was created, the images were manually selected for their respective characters and were placed in the proper character folder. Each image that was selected is ensured to include its respective character and that character only and at a minimum contains the character's entire head in the frame. Images were often excluded on the basis of more than one character in frame. In addition, if the characters were visualized in a frame in a way that completely changed their appearance, this frame would be excluded. The images being classified by the model consist of the following four characters - Spongebob, Patrick, Squidward, and Mr Krabs see Figure 1.

Figure 1: The four characters chosen to identify

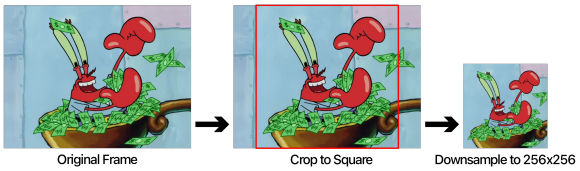

Figure 2: ffmpeg cropping process

The following is an example of the FFmpeg [2] command used to obtain the raw images from the episodes:

```
ffmpeg -i "season/episode.mkv"
-ss 63 -r 1/2 -s 256x256 -f image2
-vf "crop=720:720:114:0"
out/season/season-episode-\%03d.jpeg
```

2.1.1 Command Breakdown

`-i-` specifies the input file
`-ss 63` - skips the first 63 seconds, or to the closet frame.
`-r 1/2` - sets the rate of frames per second - 1/2 one frame every 2 seconds.
`-s 256x256` - resizes the image to the given parameter - 256x256 in this case.
`-f image2` - manually specifies output of the format jpeg.
`-vf "crop=720:720:114:0"`- crops the image to the given parameter, in this case it keeps the center square of the frame.

2.2 Model

After proper data preparation and with enough training data, the model could now be built and trained. The initial build of the initial model consisted of four layers, an input layer, a convolution layer,a pooling layer, and a dense layer. This model yielded results more impressive than the first model's design, but through experimentation it was discovered that adding a few extra layers improved accuracy. The final model consists of seven layers by adding a second convolution layer, a second pooling layer, and a dropout layer to the previous model. The model is compiled using Keras' Adam optimizer as the optimizer parameter and sparse_categorical_crossentropy as the loss function. To further obtain the best results, hyper-parameters in the model (learning rate, boolean for dropout, etc) are tuned using KerasTuner - more on this in the following section 2.2.1. Because of the limited data, a batch size of 16 was used when training the model and the validation split was 20%. The following sample sizes were used for the training of the model:

Characters	Samples
Spongebob	245
Patrick	106
Squidward	114
Mr Krabs	89

2.2.1 KerasTuner

The KerasTuner [3] is a general-purpose hyperparameter tuning library that is extremely useful in providing the best values for hyperparameters to achieve the best results. For the SpongeDetect model, the defined search space consits of a float hyperparameter for the learning rate used within the compilation of the model and a boolean hyperparameter to determine whether or not to include the dropout layer in the model. The tuner class selected for this model that runs the search is Hyperband and its specified objective to optimize is the validation data's accuracy. Tuning the hyperparamters through a series of trials at ten epochs per trial yielded the following results for the highest scoring trial:

```
Score: Validation accuracy = 0.945454537
Float: learning\_rate = 0.00049547586024
Boolean: dropout = False
```

3 Results

After preparing data, building the model, training the model, tuning the model, and testing and tweaking the model, the accuracy was finally satisfactory. To obtain final results for the model, more Spongebob images for each character were obtained from episodes of a season not previously used in the training or validation data. Approximately ten images of each character not yet seen by the model were used to test the models ability to predict.

Predictions:

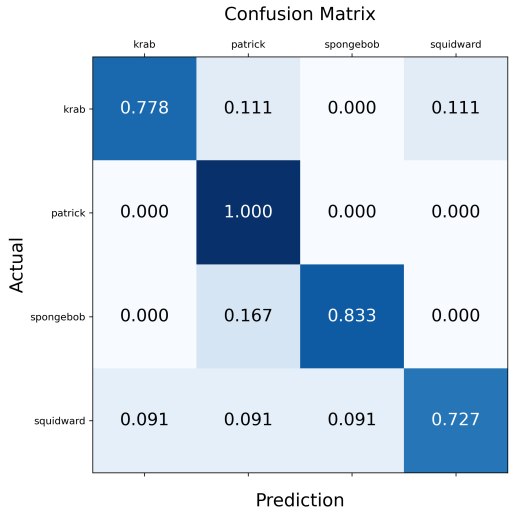

Figure 3: A confusion matrix of predictions made on unseen data.

The confusion matrix shows impressive results in that the model can very accurately predict the characters based on the limited data the model was trained with.

4 Conclusion

In conclusion, TensorFlow and Keras are extremely useful tools for developing machine learning models, and with proper layering and tuning can yield extremely accurate results. Not only can the developed model accurately classify images of Spongebob characters, it can revolutionize the world in that no one can ever forget which character was which in the cartoon. Convolutional neural networks will change the way that society identifies Spongebob characters and can even be applied to subjects beyond the Spongebob realm. Soon enough people will not have a need to store the names of these characetrs in their minds, but can simply rely on the magic behind the learning machine. Spongebob shall forever remain on top thanks to the power of machine learning.

References

[1] K. Team, "Keras documentation: About Keras," https://keras.io/about/.

[2] "ffmpeg Documentation," https://ffmpeg.org/ffmpeg.html.

[3] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi *et al.*, "Kerastuner," https://github.com/keras-team/keras-tuner, 2019.