Daniel Palacios
12/10/2019

Computer Vision

Introduction:

The goal of our project is to gain a better understanding of computer vision and the important role it plays in the advancement of autonomous vehicles. This essentially means our project is broken up into three distinct phases: image classification, object detection and finally, software integration for the vehicle. This report focuses on the progress that was made and the insights that were achieved during the first phase of image classification.

To better understand image classification, we created a computer vision model using Convolutional Neural Networks to classify a person's Halloween costume based on their perceived creativity and effort. If the costume failed to meet a threshold of expectations, it would be labeled "Basic", otherwise it was considered "Non-Basic."

Dataset Description

Since we are dealing with image classification, using CPU would simply take too long to train, so the NVIDIA Jetson Developer Kit was used for this project for the purpose of its GPU and CUDA. Our dataset was custom-built and included 3,000 images of "Basic" halloween costumes and 3,000 images of "Non-Basic" halloween costumes. We used a script titled "google_image_extractor.py" to download images from Google Images and place them into their respective directories. Next, the images were then split into 70% training data 15% testing, and 15% validation data, where they were then read, resized, labeled, converted to numpy arrays, and then saved under a numpy compression titled "Halloween-Classes.npz" for later use.

One obvious challenge with this dataset is its inherent bias from its authors. The definition of a "basic" costume will surely vary from one sub-group to the next, so the labels of our custom dataset are only as accurate as our own personal opinions and beliefs. For clarity, we decided that any costumes of cats, bunnies, skeletons, ghosts, authority figures, and angels / devils would be labeled "Basic" and any bad costumes, movie costumes, couple costumes, or just well thought out costumes would be considered "Non-Basic".

Baseline Approach Description

For image classification, CNN's are widely considered the most pragmatic approach because their convolutional layers specialize in the detection and understanding of patterns. Thus, the model will be able to differentiate patterns in basic Halloween costumes versus those in a non-basic costume. We chose to use KERAS to build out the network.

Method Description

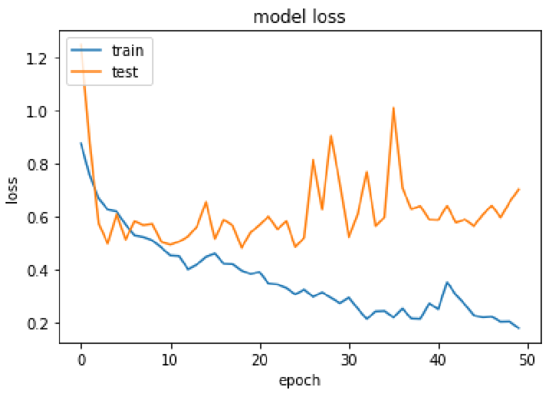
When creating the optimal model, we focused on finding the right Bias-Variance balance. Since Keras was chosen to build this network, the architecture made it easier to tune the hyperparameters and add layers when needed. Our approach was to start off with a small model of three Convolutional layers (2^5 output units) and one dense layer (2^4 output units) and attempt to continue to improve the model from there. This process resulted with 3 different models with separate results.

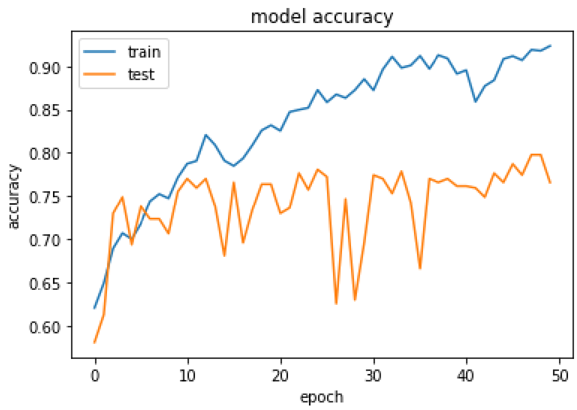
Daniel Palacios
12/10/2019

Evaluation

In Keras, we have the ability to plot the accuracy and loss of the training data over each epoch, by utilizing the weights learned from the previous epoch. This data is represented by the blue line in the following charts while the validation set results are represented by the yellow line, respectively. The first model resulted with a validation accuracy of 78% but after plotting the validation accuracy, it was obvious that this specific model was overfitting; accuracy results on the training set were 90%.

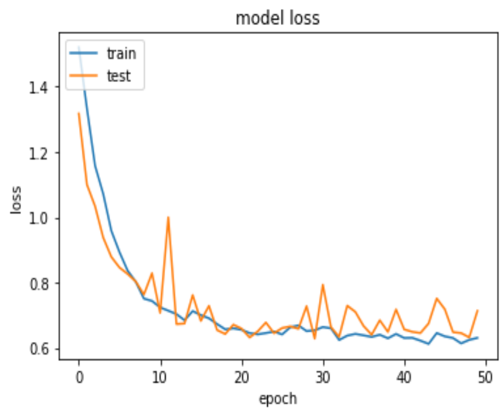
Model #1 Results:

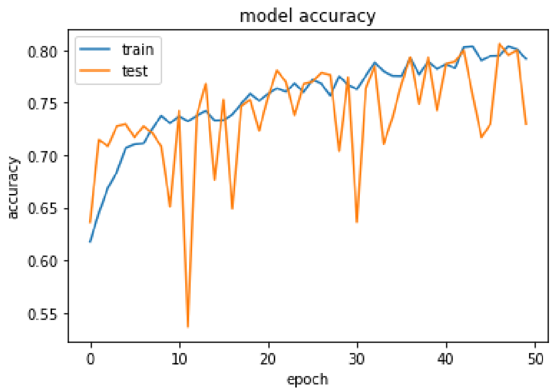




This result may be because there were too many output units for each layer (128 + 64 + 32 +16 output layers) with a very little chance of a dropout. The fix to this issue for Model #2, we decreased the output units and increased the chance for a dropout.

Model #2 Results:

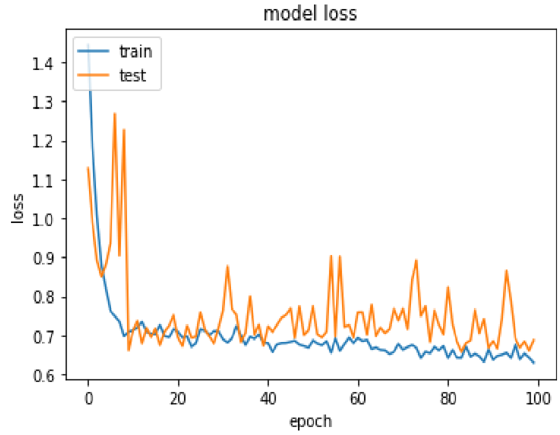


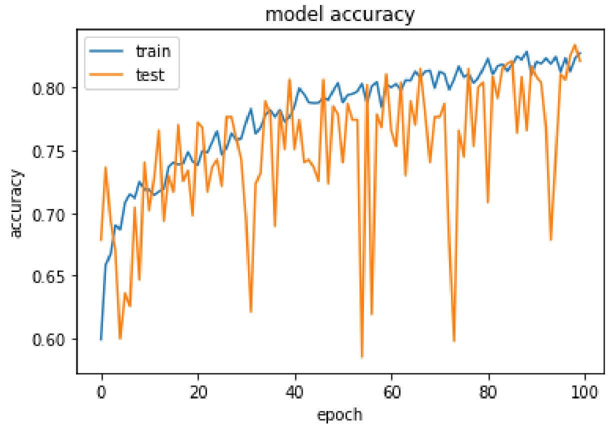


Interestingly enough the resulting validation set accuracy for this architecture ended up being 73%, which was lower than Model #1. That said, we were able to mitigate the overfitting issue, ultimately giving us a better model. We can see that during validation, accuracy seemed to decrease dramatically but then eventually continue to ride very closely with the training accuracy. The drastic variations over epochs may be the result of an excess of connections for the model to take. Therefore, for Model #3 the number of epochs was increased to 100 and we increased the dropout probability in the layers.

Daniel Palacios
12/10/2019

Model #3 Results:

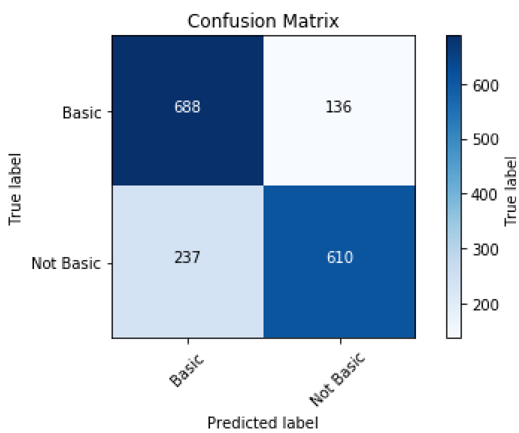




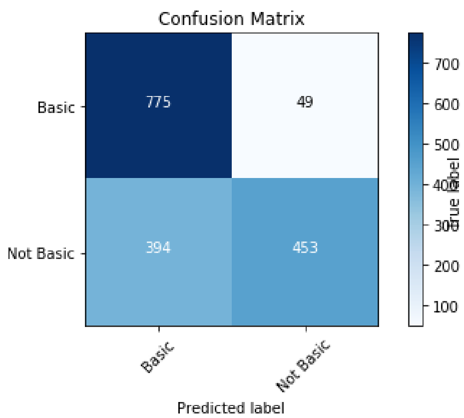
The resulting accuracy of the Model #3 validation set was 80%. As we can see from these two graphs, the validation accuracy continues to find itself dramatically decreasing at the end of certain epochs. With these models in place, we were ready to find results on actual testing data.

Around 1,700 photos in our test set were evaluated and the predictions from each model offered interesting insights. Model #1 produced decent results as it was able to detect the obvious costumes of bunnies, witches, skeletons, etc. and even with personal opinion, it was evaluated correctly. Model #2 resulted with poor testing results when determining the class of a specific costume - even with personal opinion it was not being evaluated well. Lastly, Model #3 gave the best results by once again predicting a correct class with 80% accuracy.

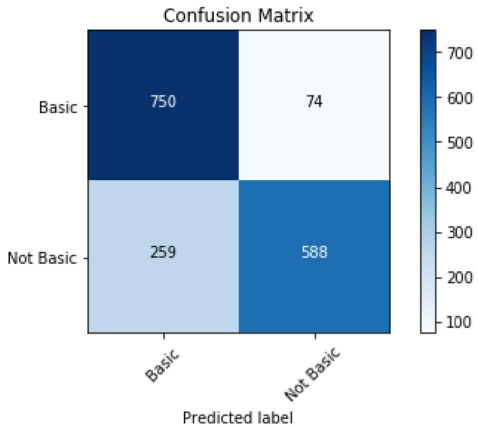
Model #1 (78%):



Model #2 (73%):



Model #3 (80%):



Discussion

When testing this model with live volunteers, there was much more to discover about our task. It is not just about the model, it is also about the quality of the data itself. To clarify, many people who took photos had different poses which dramatically influenced the decision boundary of the image classifier. For example, those who did not have a costume were in the “non-basic” category, however if they were to make a certain pose the model would predict a higher probability that the costume belonged to the “Basic” category. The given features that were discovered during testing goes as follows:

- The pose
- Defined costumes types from dataset
- Amount of clothing material
- Color

Conclusion

As with any machine learning model, there is room for improvement. Our trial and error process stressed the quality of our hyperparameters. Finding the right values for regularization, drop out, layers, and number of units being given are all important attributes for the quality of our model. Moving forward, a helpful way to optimize our hyperparameters would be to develop a grid-search that allows us to experiment with multiple combinations and then select the best performing ones for testing.

Additionally, while a label of “Basic” or “Non-Basic” were satisfactory for our initial model evaluation, we believe that less subjective classifications are necessary for the case of autonomous vehicles. These could be specific nouns like “walls”, “person”, “stoplight”, etc. or possibly descriptive labels so that a car understands the difference between objects it should maneuver around versus objects it can drive through (e.g. plastic bags or debris).

We feel that this Halloween-Classification project provided valuable insights for us to move onto the next phases of the overall project. Because of the importance of clean, quality training images, as well as the additional requirement of object coordinates, we don’t see a custom image dataset as a viable option for the object detection phase; it would simply be too involved to create this on our own. Combined with our updated approach on hyperparameter optimization, a cleaner, more reliable dataset will ensure that we achieve greater results moving into the remaining areas of our project.