CMPS 392 Project

Mahdi El Amin

December 19, 2018

1 Abstract

In my project, I will attempt to come up with a machine learning algorithm that differentiates between real and fake, or fabricated, images. I intend on doing so by examining colorized images; in other words, images that were either colored over or been subjected to a filter. In lots of cases of fake/colorized images, say when adding the face of an individual from one picture to the body of another individual in another picture, some additional coloring has to take place; like in my previous example, the two individuals will most likely have different skin tones so either the head or the body would have to be colored over to try and mask that difference. Another case would be the filter option for uploading images in social media.

2 Project Introduction and Dataset

With the introduction of new photo editing software and applications, one can safely assume that the internet is flooded with fake images. Hence, one needs a way to distinguish between what's real and what's not. Although there are plenty of ways to fabricate an image, I will focus on one. My problem is as follows, I need to distinguish images between real and colorized, i.e. they've been colored over. Therefore, I will make a dataset by searching for examples of real and colorized/filtered images on the web, through google, github, and kaggle. The images will be weighted and will hopefully result in a distinguishing model.

3 Methodologies and Related Work

In my project, I will tackle the issue using two learned machine learning techniques. The first is a simple SVM model, and the second is a Convolutional Neural Network. I will run both model using 5-fold cross validation in order to obtain the best possible individual results then evaluate and choose the better of the two. I will also use a Random Forest Classifier and tune its hyperparameters

using Randomized Search, this model will also be evaluated and compared to the previous ones. This is not the first attempt at creating a machine learning algorithm to detect fake images, as one might assume. Many have tried and trying to create such algorithms through several techniques. Dr. Neal Krawetz worked on one and used a method called Error Level Analysis(ELA) that exploits the lossy compression of JPEG images, instead of colorization like in my attempt. When an image is altered, the compression ratio of the specific portion changes with respect to other parts. A well trained neural network can detect the anomaly by and determine whether the image is fake or not. Even Adobe has attempted to develop AI tools to detect fake images, which tries to detect if an image has had objects removed or pasted in or if it is two images spliced together. And finally, several IEEE members have attempted on coming up with such a solution in the same method I am, i.e. fake colorization.
References:
- Guo, Y., Cao, X., Zhang, W. and Wang, R. (2018). Fake Colorized Image Detection.
- GitHub. Fake Image Detection. [online] Available at:
https://github.com/afsalashyana/FakeImageDetection [Accessed 22 Nov. 2018].
- Sutton, M. (2018). Adobe develops AI tool to detect fake images. [online] ITP. Available at: http://www.itp.net/617396-adobe-develops-ai-tool-to-detect-fake-images [Accessed 22 Nov. 2018].

4 Experiments

4.1 Convolutional Neural Network

First, I trained a convolutional neural network for 5 epochs with a mini-batch size of 64 using five-fold cross validation.
The architecture was as follows:

- One hidden convolutional layer with a kernel size of 3 and with 32 filters followed by a MaxPooling layer
- One hidden dense neural netwok
- One output layer
- The activation of the hidden layers is a Relu
- The activation of the output layer is a Softmax
- The loss function is a categorical cross-entropy funtion
- The optimizer of this model is Adam

The best model achieved yielded the following results:
- Accuracy = 0.5006
- Loss = 8.04937689666748

This model suffered from high bias.

4.2 Support Vector Machine (SVM)

Next, I tried using SVM models, once with an rbf kernel and another time with a polynomial kernel, but neither of them converged.

4.3 Random Forest Classifier

Finally, I trained a Random Forest Classifier which had its hyperparameters tuned using Randomized Search.
The hyperparameters that were tuned are:

- Maximum depth
- Maximum features
- Minimum sample splits
- Bootstrap
- Criterion

This model resulted in the following confusion matrix:

Confusion Matrix	Actual Positive	Actual Negative
Classified as Positive	2861	1347
Classified as Negative	2145	3647

and the following accuracy and f1-score:
- Accuracy = 0.6508
- F score = 0.6486292455337377

5 Conclusion

In short, I tested different types of models, including a neural network, to find the best one that can distinguish between real and fake images using the dataset provided. The best of these models, judging by their accuracies, was the Random Forest Classifier, which yielded a 65% accuracy. I believe further testing and experimentation could lead to a model of even higher accuracy and f1 score. Perhaps even trying different types of classifiers.