**Abstract**

We found that keras, a high-level machine learning TensorFlow API, can be shown to exhibit vulnerabilities when used in conjunction with images of clothing from the Fashion-MNIST dataset that were incrementally morphed in sequence using the *autoimagemorph* python command line application. These image prediction and accuracy ratings between the three types of clothing in each group are plotted across a 2-dimensional line plot. Within these line plots, there occurs strong fluctuations as one article of clothing is morphed into the next, and the keras prediction tool displays an uncertainty over which article of clothing is presented.

**Introduction**

Machine learning models are increasingly becoming better capable of task automation, anomaly detection, and predicting outcomes based on data and training algorithms. However, it should be understood that these tools are prone to errors. For example, while certain articles such as a shoe and a coat may be easily distinguishable to a human being, a machine learning technology’s capability depends on the training data it is supplied, and whether it has encountered factors within this training data that would allow it to categorize the article of clothing. There is a case of a neural network classifier incorrectly predicting an image of a revolver handgun, and an image of a vulture to be an orangutan after an image rotation and translation. However, if there existed ambiguity among these target variables, would the predictions generated by these models change? Would it be possible to demonstrate the inherent vulnerabilities of neural network models when making predictions on test data that is not clearly defined? Finding new means by which the vulnerabilities of these artificial intelligence classifiers can be exposed can potentially lead to better understanding the limitations of these tools, and how they can be improved.

A novel approach to testing the capabilities of these machine learning technologies is to present it with images that have been modified using a morph sequence generator which takes two or more images as input and creates a specified number of new images between them which emulate one image progressively morphing into the next image. While an image classifier may have a high accuracy in predicting a shirt or a dress, a combination of the two of those images mid-morph can have an adversarial effect on ability of the classifier to make that prediction.

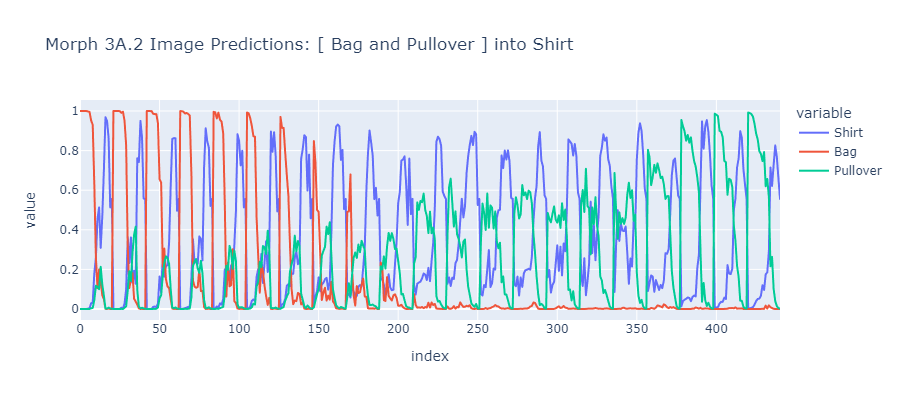
In order to determine whether or not prediction models are susceptible to uncertainty when presented with data that is vague, an appropriate neural network and dataset is necessary. The Fashion MNIST dataset, a collection of grayscale 28x28 pixel images of articles of clothing, and the TensorFlow keras neural network library were both determined to be applicable tools for this purpose. The remaining task of obscuring target variables would be performed using the autoimagemorph python tool, which automatically creates a morphing sequence between two or more select images using delaunay triangulation, delaunay triangulation, projective/affine transformation, application of projections through matrices, masking, and alpha blending[[1]](#footnote-1)[[2]](#footnote-2).

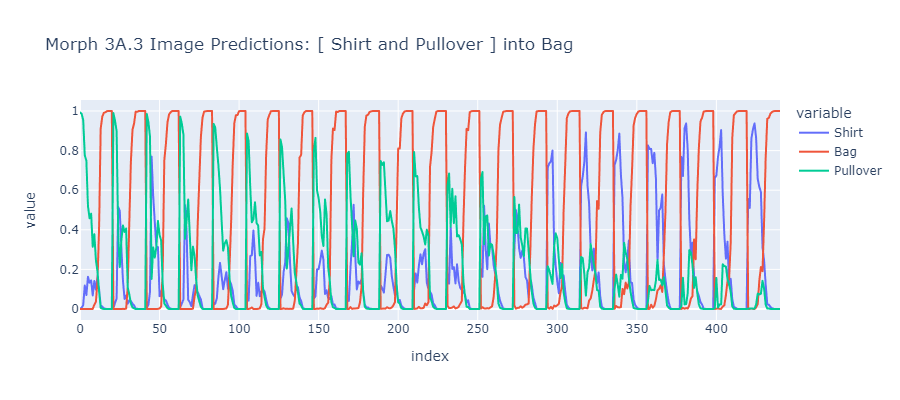
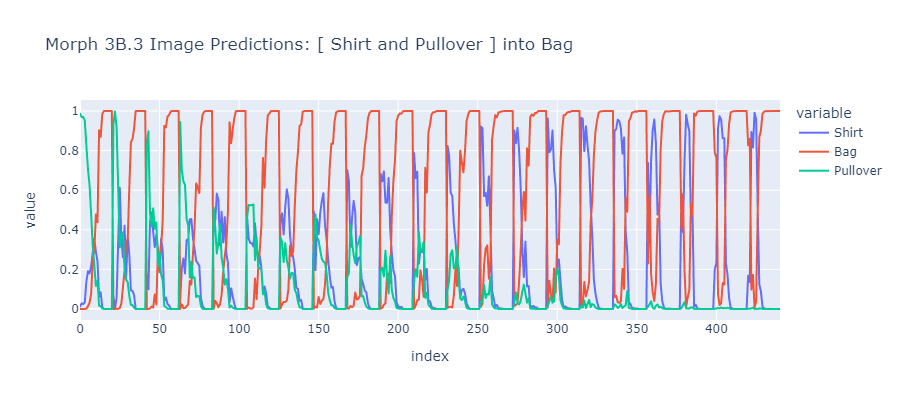
**Image Predictions**

We demonstrate these vulnerabilities by collecting the accuracy results of image predictions using 5 groups of 2 sets of 3 randomly selected articles of clothing from 3 distinct classes from the 10 available within the dataset. This first set of classes is used to produce a second set using different articles of clothing belonging to the same classes as the first set. Each permutation of pairs of clothing from these sets of three are used as input for the autoimagemorph application, and produce a collection of 21 morph sequences each consisting of 21 images of the transition between the first and second image. Starting at 100 percent of the first image and 0 percent of the second image, the first image of each morph sequence changes at 5 percent increments until reaching 0 percent of the first image and 100 percent of the second image.

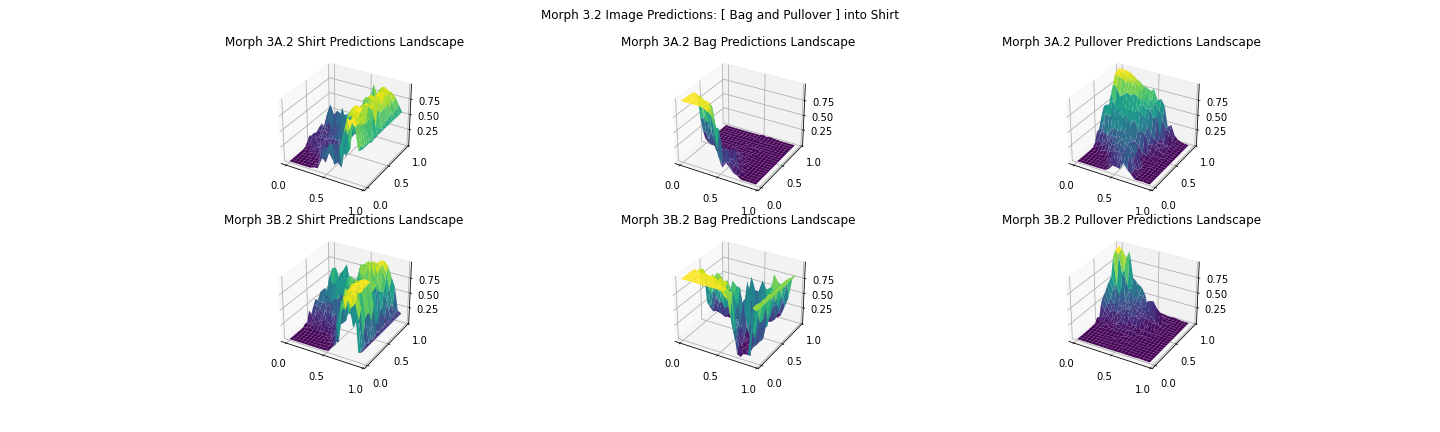
**Results**

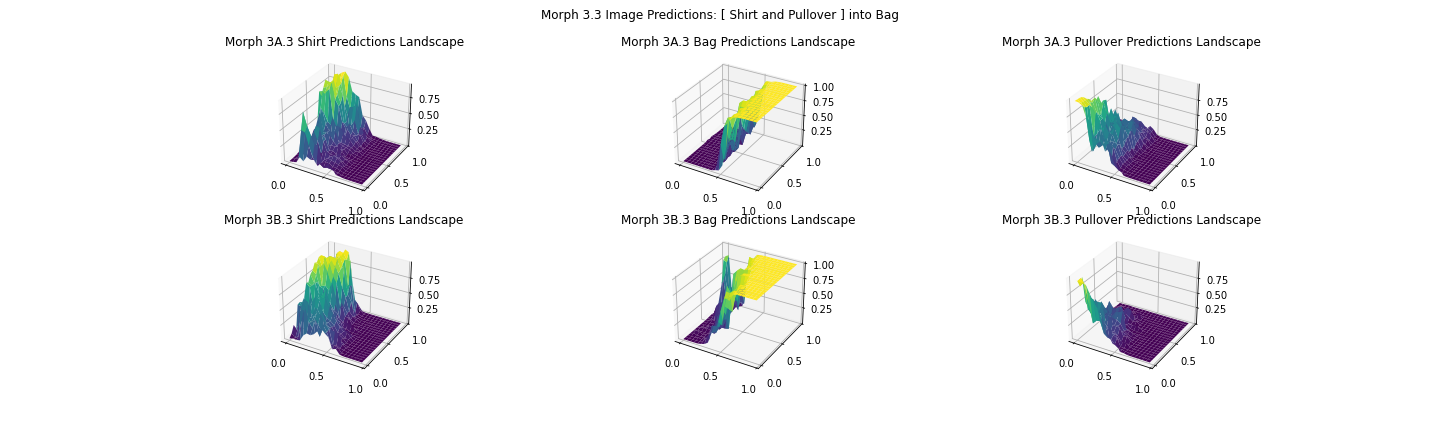
The results of these line graphs reveal the prediction accuracy of the TensorFlow keras library varies according to where in the sequence the prediction is generated and what articles of clothing are used within the morph sequence. For example, consider a comparison between *Figure 3A.2: [Bag and Pullover] into Shirt* and *Figure 3A.3: [Shirt and Pullover] into Bag*. The accuracy rating throughout each of the 21 cycles, while fully morphing into a shirt and bag respectively, share different patterns of fluctuations. While the model is able to discern the image is not a bag around the half way mark of 3A.2, its ability to differentiate between a pullover and a shirt reveals a level of uncertainty. This is apparent when compared with 3A.3, which more quickly and recurringly predicts the transition into a boot during each and every cycle. Using different images belonging to the same types of clothing (3B.2 and 3B.3), the same patterns with similar levels of fluctuations are produced.





The allocation of image predictions across a table were also utilized in creating 3-dimensional landscapes, which also revealed similar patterns albeit with the predictions for each unique article of clothing separated onto an individual graph, with each slice overlapping with the end of each cycle.





The process of generating predictions using morph sequences, while successful for these selected groups of images, were sometimes prone to error. It appeared that while the high granularity and low pixelation of the Fashion MNIST dataset images would be beneficial to the TensorFlow keras model given the smaller field to assess, it served problematic for the autoimagemorph tool using its default settings. From its documentation, the tool is more capable of generating a morph sequence using images with a higher pixel count and greater level of detail. It became necessary to adjust the *featuregridsize* option to 3, which allows compatibility with a greater number of MNIST images. Another issue involved the amount of time required to generate each morph sequence for each permutation within each group and set, totaling 30 morph sequences over several hours, even if TensorFlow is running using the GPU. Memory Limitations along with error occurrences lead to saving data in .pkl format after each morph sequence a safe practice.

The project also at an early stage involved the creation of 3-dimensional landscapes using a random walk function with a vector field in order to visualize the effect that magnitudes of force, fixed weights, and slope can have on a randomly moving object across a landscape. These movements were recorded across landscapes of different shapes, including symmetric/asymmetrical bowls and randomly generated fractal landscapes as the LOG of the average distance vs. the LOG of the number of steps.

1. https://github.com/jankovicsandras/autoimagemorph [↑](#footnote-ref-1)
2. https://github.com/ddowd97/Python-Image-Morpher [↑](#footnote-ref-2)