Determining Position in a Wrestling Match Using Custom Made Olympic Wrestling Dataset

Tyler Eischens
Stanford University
450 Serra Mall, Stanford, CA 94305
tyeisch@stanford.edu

1. Abstract

There has been almost no published and public research linking Computer Vision applications to combat sports and Wrestling specifically. In this project, I used a custom-made dataset containing various images of Freestyle and Greco wrestling matches along with an image classification algorithm in order to predict the positions of the wrestlers in the match. There were two positions that the algorithm was trained on, 'Neutral' and 'Par Terre'. The algorithm was based off of the representation learning algorithms implemented in the course this year as well as the framework outlined in [1]. I created two different versions of the model to run the dataset through, which I called 'Pre-Optimization' and 'Post-Optimization'. The highest Pre-Optimization test accuracy was XXX and the highest Post-Optimization test accuracy was XXX. While this is by no means a huge stride in Computer Vision application, it is my hope that further advancements in this line of thinking could lead to things such as matches being refereed completely by a computer or videos showing detailed technique breakdowns just by implementing a few cameras

.

2. Introduction

Wrestling is one of the world's oldest sports. Having been in every Olympic Games since their inception, it is a beautiful thing to watch and participate in. I have been wrestling for over 15 years now, and I continue to learn new techniques and be bewildered by the calls of some referees. From bizarre penalty points to not making any call whatsoever, there have been hundreds of matches that have been decided not by the quality of the wrestling going on, but by the discretion of a single referee who has no clue what they are doing. No human is perfect, so it makes sense that referees can make mistakes. But what if there was a way to eliminate human error in refereeing wrestling matches?

This was the question I thought on for my final project, and while the final outcome of my project is not going to be refereeing matches by itself anytime soon, I do believe that it could be applied by someone else to make progress in that direction.

In order to tackle this problem of human error in refereeing matches, I implemented an image classification algorithm that takes as input an RGB image of a Freestyle or Greco Wrestling match and returns either 1 of 2 options, 'Neutral" or 'Par Terre'.

The image classification algorithm I implemented leans heavily on the framework described in [1]. It also is implemented using a dataset that I created by downloading and the labelling Olympic wrestling images from [2].

The main difficulty of this project was the lack of datasets available for use. While the algorithm I created can attain a respectable accuracy on the dataset given, it is hard to know how accurate it would be given a larger sample size, considering the dataset I created only consists of 177 images.

The rest of the paper will be organized like so; in the third section, I will discuss work related to the project at hand. In the fourth section, I will talk about image classification and the techniques that I implemented specifically in this project. In the fifth section I will discuss the dataset that I created, and challenging aspects involving labelling, and then how the dataset was used in tandem with the image classification techniques to acquire our results. In the sixth and final section I will discuss the conclusion and potential future projects involving wrestling and computer vision.

3. Related Work

This project merges two fields that are on quite opposite ends of the spectrum. While image classification has taken huge strides in the computer vision industry recently due to advancements in machine learning, the application of Combat Sports in computer vision is likely the most underrepresented category of sports.

3.1. Image Classification Related Works

Image classification is being put to use in many modern applications, from medicine to facial recognition. In this paper by Yadav and Jadhav [6], they utilized image classification and a convolutional neural network in order to identify x-rays of patients and determine if they had pneumonia or not. The paper expresses the difficulty in obtaining valid datasets for medical practice as they have to be labeled by a medical professional in order for them to be used.

Image classification is used in facial recognition techniques as well. This diagram from the book '*Handbook of Face* Recognition' shows the process by which an image or video is converted to be recognized [7].

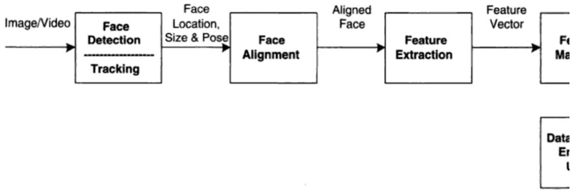


Figure 1: A network of an image or video being processed and prepared for facial recognition

It is extremely similar to the series of processes that I used in implementing our position classification algorithm.

In this paper by Varela et al [8], image classification techniques are used in habitat mapping, and compared to remote sensing, which is generally considered to be a better approach, but image classification is fine tuned and ends up working quite well.

A paper by Rallet [9] takes a set of 13,800 images from 22 different sports and uses a feedfoward neural network in order to properly classify the images. While this is not exactly the same as what my project intended to do, it is the most similar and most recent image classification piece that I could find. After multiple different implementations, including using pretrained models and changing to a convolutional neural network, Rallet managed to achieve 76% test accuracy, which for 22 different classes and 13800 images seems quite good [9].

3.2. Computer Vision Sport Applications

The only prominent article I could find that dealt directly with the combination of computer vision and specifically wrestling was written by Mottaghi *et al.* [4], and was much more complex than the scope of my project. Mottaghi *et al.* [4] researched a branch of Human Action Recognition. Mottaghi *et al.* [4] first used complex mathematics to extract the skeleton silhouette from a various frames of a video of wrestling. They then transformed that silhouette into a graph that reveals skeleton structure [4]. This technique is extremely flexible because it does not rely on a body model or any other information. Mottaghi *et al.* [4] then obtained a final feature vector of the position after extracting the spatial features from the skeleton structure graph. Mottaghi *et al.* [4] notes that although the challenge of studying wrestling interlinked with computer vision techniques is quite high, there is much to gain in the field, especially when it comes to refereeing assistance and teaching wrestling techniques.

While applications of various computer vision techniques in combat sports is not very common, they have been used quite frequently in other sports. This article by Martinez Aratsey [10] discusses the many aspects of computer vision that are integrated within our professional sporting world. Starting with racket and bat-and-ball sports such as tennis, badminton, baseball, etc., these are the events where computer vision has been used for the longest time. Since the mid 2000's, ball tracking technology has been implemented to determine speed, strikes, foul balls, and out-of-bounds calls. At the 2017 Wimbledon, computer vision technology was used to automatically create highlight clips based off of crowd noise, match data, and player movements [10]. In Soccer, computer vision technology has been used for goal tracking, such as a 7-camera system developed by Hawk-Eye and a two-camera player tracking system that was useable from any press box in any stadium [10]. This player tracking system managed to retain quality of information while reducing the necessary number of camera operators. In American Football, computer vision applications have been bountiful. Techniques are used to track offensive and defensive player formations on the field, the frequency of formations, the likelihood of certain plays being ran at certain field positions, and more [10].

A discussion on Quora went over two different ball tracking systems that are used for professional golf. The latter one, Protracer, utilizes a whole host of computer vision applications [11]. These include ball tracking, image differentiation, and ball identification [11].

The use of computer vision in professional sports has significant challenges, however. Player and ball occlusion is a huge one. Other challenges include varying body posture and positions, fast and erratic motion from players and objects, close interactions between competitors, and athletes looking the same as one another [10].

4. Image Classification In Depth

Image classification has developed rapidly with advancements in machine learning and new techniques being brought forward. This article by Sanghvi discusses a few of the major techniques used in current image classification, as well as newer techniques being developed [3]. The main idea behind image classification is to take an input image and pre-process it so that the model can better identify the key features in the image. Some common pre-processing techniques are transformations, gaussian blurring, and noise reduction.

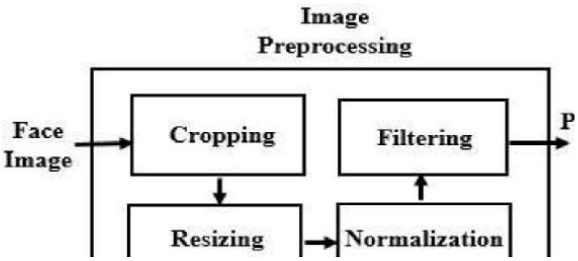


Figure 2: A typical implementation of pre-processing

After pre-processing, images are then subjected to object detection to determine where in the image the object of interest is. This is not always necessary as features are also extracted in the next step when the model we choose is being trained to identify the key features or patterns present in each of the different classes. In the last step of image classification, the model then classifies the objects/images into the different established classes using various classification techniques.

Supervised Classification is an image classification technique in which the user selects specific pixels in the image that are representative of the class in question, at which point the software used to process the image focuses on these selected pixels as 'training sites' to be used in other images [3]. There are multiple algorithms that supervised classification techniques use to develop their models including linear and logistic regression, neural networks, and more [3].

Unsupervised Classification is similar to the technique discussed above, but the software itself makes the decision on which pixels are related in the image without relying on any user input. Unsupervised classification relies on classification algorithms such as neural networks, anomaly detection, and cluster analysis [3].

The technique that is utilized in my project is a Convolutional Neural Network (or CNN), which is a special type of multi-layer neural network that is designed specifically for pattern recognition in images with minimal required pre-processing. There are only two main features of a convolutional neural network, the convolutional layers and the pooling layers. While these are not very complex themselves, the true complexity lies in how to order them, along with other functions that ensure the shapes of the layers remain in sync. The key highlight of using a convolutional neural network is that the user does not have to implement any feature extraction themselves. It is all done internally by the system.

Another type of classifier available is a Support Vector Machine. These are powerful and flexible supervised machine learning algorithms [3]. This algorithms effectiveness is greatly determined by the kernel chosen, just as a linear kernel, guassian kernel, or polynomial kernel [3].

K-Nearest neighbor is another type of image classification algorithm where input is 'k' of the closest training examples in the feature space the algorithm is working in [3]. This algorithm is generally considered one of the simplest classification algorithms, as it does not do much computing and is non parametric [3]. Via this algorithm, objects are classified by how far they are from a certain feature vectors and a plurality vote from its neighbors [3]. This is where the 'k' nearest neighbors come in.

5. The Dataset and the Experiments

The dataset that I created for this project consists of 177 images that 128x128 in size and feature different poses of Olympic wrestling, in both Freestyle and Greco.

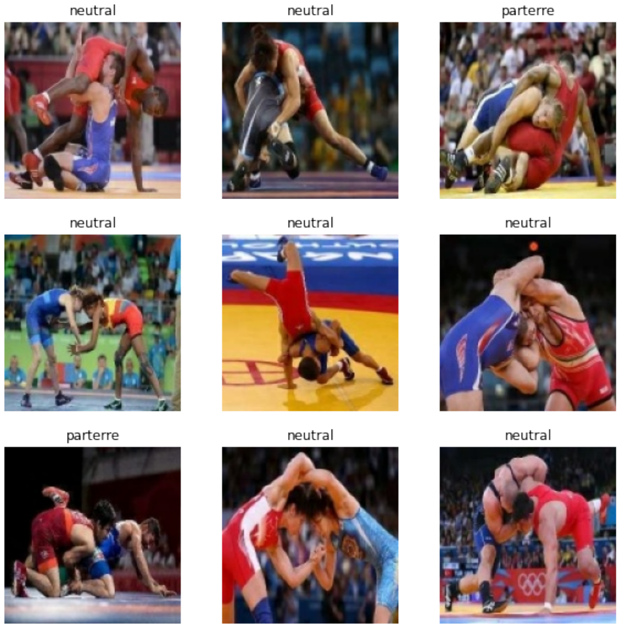
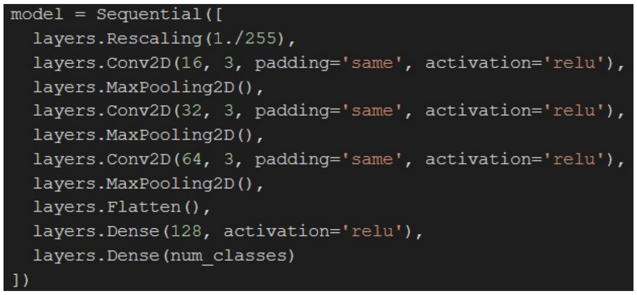


Figure 3: 3x3 showing various labeled images from the dataset

The images from the dataset were gathered from [2], while for the labels I had to create a .csv file to manually label everything initially. After experimenting with exactly how I wanted to format my code, using the tensorflow module in Python, I rearranged the folders that the images were stored in such that everything that was labeled as Neutral in the csv file went in one folder, and everything that was labelled as Par Terre went in to another. Before even discussing results, a major downside to the dataset that exists is the disparity between the number of 'Neutral' classified images and 'Par Terre' classified images. The dataset has 85 images labeled as 'Par Terre' and 92 images labeled as 'Neutral'. While this disparity may not seem so large, it could account for various inaccuracies in the final testing results.

I implemented two different strategies to train the data using a training set of images and a validation set of images. The training and validation sets were split in a 80% to 20% ratio. The first strategy used was a typical convolutional neural network with the following form:



```
model = Sequential([
  layers.Rescaling(1./255),
  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(32, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(64, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Flatten(),
  layers.Dense(128, activation='relu'),
  layers.Dense(num_classes)
])
```

Figure 5: Lines of code showing the layers involved in the first implemented convolutional neural network.

The initial convolutional neural network consists of 3 2D convolution layers and 3 2D max pooling layers, plus the 3 layers at the end to properly format the output. By training on different numbers of epochs, I determined that an optimal number of epochs before no more relevant change happened was 15. After running these 15 epochs for multiple iterations, I returned these graphs:

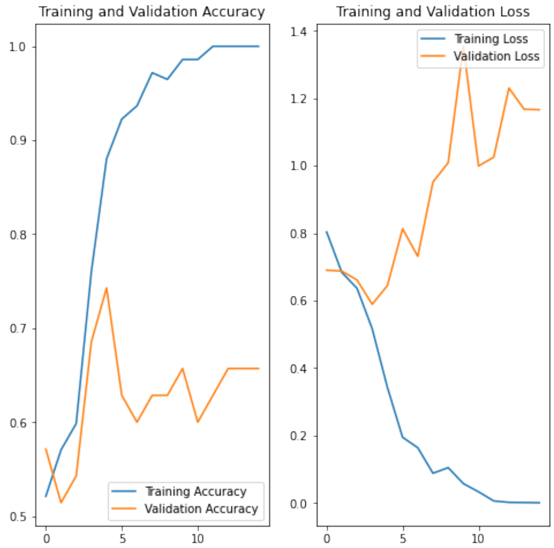
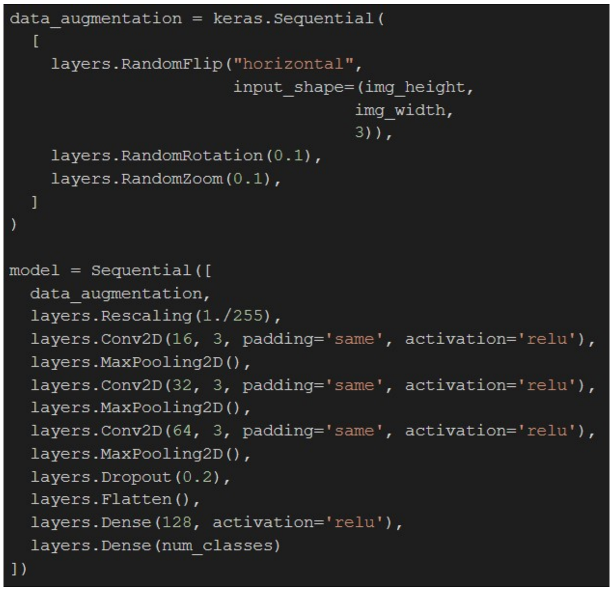


Figure 6: Pre-Optimization Graphs of our Training and Validation Accuracy and Loss

It is clearly visible that while the training accuracy actively reaches 1.0, the validation accuracy is severely off, meaning that something is not quite right. In order to fix this, I modified the model that we were using previously by adding data augmentation and a dropout layer to it. The purpose of the data augmentation layer is to generate additional piece of training data due to the small data set size. As seen in figure 7, the data augmentation has a chance to randomly flip, rotate, or zoom in on the images, creating more diversity in our small data set. The point of the dropout layer is to prevent the smaller number of samples from relying on the other training results too much. With both of these layers implemented our new model looks like so:



```
data_augmentation = keras.Sequential(
  [
    layers.RandomFlip("horizontal",
                      input_shape=(img_height,
                                   img_width,
                                   3)),
    layers.RandomRotation(0.1),
    layers.RandomZoom(0.1),
  ]
)

model = Sequential([
  data_augmentation,
  layers.Rescaling(1./255),
  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(32, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(64, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Dropout(0.2),
  layers.Flatten(),
  layers.Dense(128, activation='relu'),
  layers.Dense(num_classes)
])
```

Figure 7: Our modified convolutional neural network that now implements data augmentation and a dropout layer.

With these new parameters added to the neural network, we generated a new set of training and visualization accuracy and loss graphs, seen here:

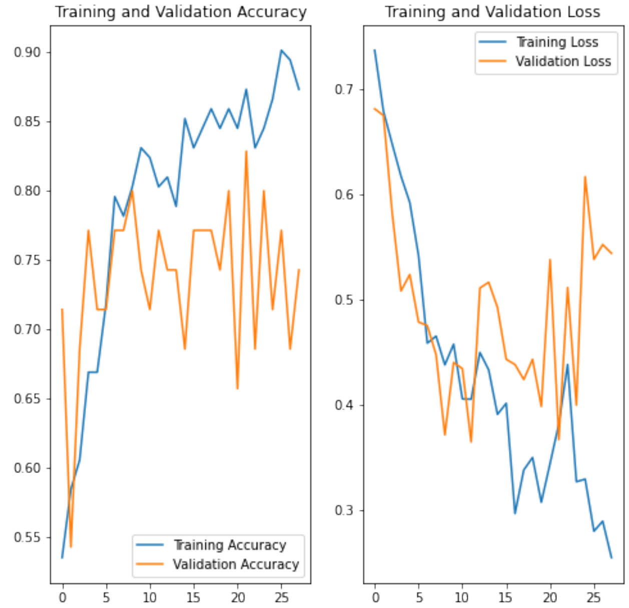
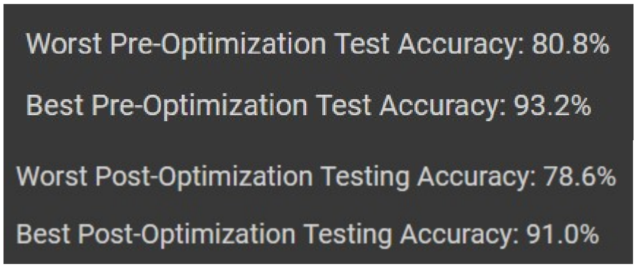


Figure 8: The Training and Validation Accuracy and Loss of our model after adding in data augmentation and a dropout layer for optimizations.

Although the overall training accuracy decreased, the validation accuracy is now much closer to it, with the same going for the loss. I believe that this proves the optimizations were truly helpful.

At the end of each model's training and validation, I then plugged the entire dataset back in to the model to use for testing. I know that typically this is unconventional, but with such a limited amount of data I felt like it was an appropriate route to take. I recorded the highest and lowest accuracies that came out of the tests after around 30 iterations each. The results were as follows:



Worst Pre-Optimization Test Accuracy: 80.8%

Best Pre-Optimization Test Accuracy: 93.2%

Worst Post-Optimization Testing Accuracy: 78.6%

Best Post-Optimization Testing Accuracy: 91.0%

Figure 9: A list of the best and worst testing accuracies from pre- and post-optimization.

From just these results alone, it would seem like the model got less accurate from the 'optimizations' however, we saw above how the two different models compared with their validation accuracies, and therefore the post-optimization model sure is the best fit.

6. Conclusion

In this project, image classification techniques were implemented on a newly constructed Olympic Style wrestling dataset in order to attempt to predict the current position of the wrestlers in the image. Using a Convolutional Neural Network to train the model, we obtained results that were higher than I expected in regard to the low quality of the dataset available.

At the conclusion of my project, I was pleasantly surprised with how accurate the algorithm I created was at determining the correct position. With a high of 93.6% accuracy, the results seem quite promising, especially if the algorithm were to be used on a larger dataset. A major concern however is the accuracy of the labelled images in a larger dataset. The issue with simply looking at and labelling pictures that are not correlated to each other is that it is next to impossible to determine the original position two wrestlers started from in relation to certain positions that are captured in a single image. Take this here picture for example:

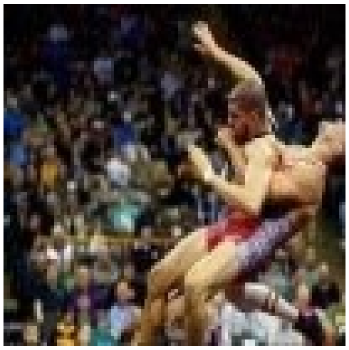


Figure 10: Wrestlers in an Indeterminate Position

In the picture, it could be argued that the wrestler in the blue singlet made it behind the wrestler in the red singlet from the Neutral position and is now attempting to score a takedown. At the same time, it could be argued that the wrestler in the red singlet was lifted from the Par Terre position and now the wrestler in the blue singlet is attempting to throw them to score more points. The best way I can think of to solve this issue is to not only create more classifications of images such as 'Par Terre- Grounded', 'Par Terre- Not Grounded', 'Neutral- Standing', 'Neutral- Scramble', and more, but furthermore, this much more in-depth labeling would only be possible if we had entire sequences of images and not just singular ones. This brings me to my next point, which is that it is unrealistic to make an accurate dataset based solely off of singular images for reference. I propose that any wrestling dataset created must be constructed in a way that accounts for multiple image frames in a row in order to properly determine the original position and how the wrestlers ended up in the current position. This would require using video of matches instead of just singular image captures.

This idea of video being used to create a more in-depth and accurate dataset leads me to my next point, which is the use of multiple cameras to detect positions in matches. This application of 3D vision techniques would be the logical step if this technology was to be implemented such that it was possible for a computer vision system to referee a match on its own with no outside interference. With at least 3 cameras recording a match, things such as technique recognition out-of-bounds detection become much easier to track and implement. The goal of future work in this area is to create a system that can track a wrestling match in real time and determine when points are scored, the clock must be stopped, and the current position that the wrestlers are in.

Link to GitHub Repository: https://github.com/TyEisch/CS231AFinalProject

References

[1] TensorFlow. "Image Classification | TensorFlow Core." Accessed March 19, 2022. https://www.tensorflow.org/tutorials/images/classification.

[2] "Images.Cv | Image Datasets for Computer Vision and Machine Learning." Accessed March 19, 2022. https://images.cv/.

[3] Sanghvi, Kavish. "Image Classification Techniques." Analytics Vidhya (blog), September 25, 2020. https://medium.com/analytics-vidhya/image-classification-techniques-83fd87011cac.

[4] Mottaghi, Ali, Mohsen Soryani, and Hamid Seifi. "Action Recognition in Freestyle Wrestling Using Silhouette-Skeleton Features." *Engineering Science and Technology, an International Journal* 23, no. 4 (August 1, 2020): 921–30. https://doi.org/10.1016/j.jestch.2019.10.008.

[5] Easley, Glenn, Dianne O'leary, and David Schug. "Precise State Tracking Using Three Dimensional Edge Detection." In *Applied and Numerical Harmonic Analysis*, 2017. https://doi.org/10.1007/978-3-319-54711-4_4.

[6]Yadav, S.S., Jadhav, S.M. Deep convolutional neural network based medical image classification for disease diagnosis. *J Big Data* **6,** 113 (2019). https://doi.org/10.1186/s40537-019-0276-2

[7] Stan Z. Li and Anil K. Jain. 2011. Handbook of Face Recognition (2nd. ed.). Springer Publishing Company, Incorporated.

[8] Díaz Varela, R.A., Ramil Rego, P., Calvo Iglesias, S. et al. Automatic habitat classification methods based on satellite images: A practical assessment in the NW Iberia coastal mountains. Environ Monit Assess 144, 229–250 (2008). https://doi.org/10.1007/s10661-007-9981-y

[9] Rallet, Bénédicte. "Sport Image Classification with Neural Networks." Medium, July 22, 2020. https://blog.jovian.ai/sport-image-classification-with-neural-networks-16929b9f7932.

[10] Martinez Arastey, Guillermo. "Computer Vision In Sport." Blog. Sport Performance Analysis, April 17, 2020. https://www.sportperformanceanalysis.com/article/computer-vision-in-sport.

[11] Quora. "What Computer Vision Algorithms Are Used in Protracer for Golf Ball Flight?" Accessed March 19, 2022. https://www.quora.com/What-computer-vision-algorithms-are-used-in-Protracer-for-golf-ball-flight.